

КЛАССИЧЕСКОЕ УНИВЕРСИТЕТСКОЕ ИЗДАНИЕ

Серия основана в 2010 году



КЛАССИЧЕСКОЕ УНИВЕРСИТЕТСКОЕ ИЗДАНИЕ

Редакционный совет серии:

Председатель совета
ректор Белорусского
государственного университета
С. В. Абламейко

Члены совета:

А. В. Данильченко (зам. пред.), Н. Н. Герасимович (отв. секретарь),
М. А. Журавков, С. Н. Ходин, И. С. Ровдо, И. И. Пирожник,
В. В. Лысак, О. М. Самусевич, О. А. Ивашкевич (зам. пред.),
В. М. Анишик, П. А. Мандрик

БЕЛАРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КРИПТОЛОГИЯ

*Допущено
Министерством образования Республики Беларусь
в качестве учебника для студентов
учреждений высшего образования
по математическим и техническим специальностям*



МИНСК
БГУ
2013

УДК 004.056.5(075.8)
ББК 32.811.4я73
К82

Авторы:
**Ю. С. Харин, С. В. Агиевич,
Д. В. Васильев, Г. В. Матвеев**

Рецензенты:
кафедра информатики Белорусского государственного
университета информатики и радиоэлектроники
(заведующий кафедрой доктор технических наук, профессор *A. A. Иванюк*);
кафедра высшей математики и физики
УО «Военная академия Республики Беларусь»
(заведующий кафедрой доктор технических наук, профессор *B. A. Липницкий*)

Криптология : учебник / Ю. С. Харин [и др.]. – Минск : БГУ, 2013. –
К82 511 с. – (Классическое университетское издание).
ISBN 978-985-518-962-7.

Изложены математические и компьютерные основы криптографической защиты
информации в сетях и системах связи.

Для студентов учреждений высшего образования, обучающихся по математиче-
ским и техническим специальностям.

УДК 004.056.5(075.8)
ББК 32.811.4я73

ISBN 978-985-518-962-7

© БГУ, 2013

Уважаемые читатели!

Серия «Классическое университетское издание» была основана в 2010 году к 90-летию Белорусского государственного университета. Путь, который прошло наше учебное заведение в своем развитии, свидетельствует о становлении в нем собственной академической и научной традиции. Несомненно, опыт и знания, аккумулированные в стенах БГУ, являются не только предметом нашей гордости, но и достоянием всего белорусского общества. Одна из целей предлагаемой серии — сделать это достояние как можно более открытым и доступным.

Белорусский государственный университет всегда славился академичностью и фундаментальностью в подготовке специалистов. Однако сегодня этого уже недостаточно. От выпускника требуется умение быстро включаться в непосредственную практическую работу, которой свойствен синтез нескольких форм деятельности: собственно производственной, исследовательской, проектно-разработческой. В выигрыше в конечном итоге окажется тот, кто сегодня научится более эффективно создавать и применять знания, оперативно изменять технологии, совершенствовать и радикально трансформировать накопленный опыт. Вот почему совмещение преимуществ фундаментального и прагматического образования стало основой инновационно ориентированной подготовки будущих специалистов в нашем университете.

Серия отражает многолетний опыт научно-педагогической, методической и издательской работы БГУ. Ее цель — представить модель учебного текста, которая в своей структуре содержит набор программ образовательно-научно-производственной деятельности будущих специалистов. Реализация этой модели позволит обеспечить

универсализм выпускника, его способность к эффективному решению важных задач, стоящих перед Республикой Беларусь на национальном и международном уровне.

Классическое университетское издание, являя собой сплав научной и педагогической мысли, призвано формировать особую культуру знания — передового и доступного, теоретического и практического, общекультурного и специализированного. Словом, такого знания, которое будет работать.

Книги этой серии должны стать образцом научно-методического обеспечения современного образовательного процесса в высшей школе, утвердить ведущую роль нашего университета в качестве национального научно-методического центра Республики Беларусь.

Надеемся, что серия «Классическое университетское издание» состоится и как одно из слагаемых особой культурно-образовательной среды БГУ, которая будет способствовать интеллектуальному росту и творческой созидательной деятельности наших студентов.

*Ректор Белорусского
государственного университета
академик НАН Беларуси, профессор*



С. В. Абламейко

ПРЕДИСЛОВИЕ

Глобализация современного мира ведет к тому, что *информация*, являющаяся основой для принятия решений, становится все более высокоценным товаром, который необходимо не только передавать и хранить, но и *защищать*. Проблемы защиты информации, информационной безопасности приобретают статус ключевых в XXI в.

Среди способов защиты информации наиболее надежным считается криптографический, предусматривающий такое преобразование информации, при котором она становится доступной для прочтения лишь обладателю некоторого секретного параметра (ключа).

В последние годы область применения криптографии значительно расширилась. Криптографические системы защиты информации стали использовать в повседневной практике многие организации, коммерческие фирмы, частные лица. При этом законного пользователя того или иного криптографического средства прежде всего беспокоит его надежность. Одним из способов оценки надежности является попытка «взлома», т. е. получение доступа к информации без знания ключа. Задачи защиты информации призвано решать смежное научное направление, называемое криptoанализом. Криptoанализ и криптография в совокупности формируют новую науку – криптологию. *Криптология* – это раздел прикладной математики и информатики, в котором изучаются модели, методы, алгоритмы, программные и аппаратные средства криптографической защиты информации, а также оценки ее эффективности.

За рубежом курсы по различным разделам криптологии уже более двадцати лет читают студентам, специализирующимся в области математики, прикладной математики, информатики и электроники [2, 9, 12]. В Республике Беларусь в 1997 г. открыта новая специализация «Математическое и программное обеспечение криптографии и анализа данных» (квалификация «Математик-программист»), а в 2002 г. – новая специальность «Компьютерная безопасность» (квалификация «Математик. Специалист по защите информации»).

Настоящее издание – *первый отечественный учебник по криптологии*. В нем изложены математические и компьютерные основы криптографической защиты информации в компьютерных сетях и системах связи. Приведены математические модели и методы построения крипtosистем, а также методы оценки и надежности (стойкости). Рассмотрены методы генерации и статистического тестирования случайных и псевдослучайных последовательностей. Представлены математические подходы к решению проблем аутентификации, к построению и анализу криптографических протоколов, систем разделения секрета.

При написании данной книги учтен 10-летний опыт преподавания предмета «Компьютерная безопасность» в Белорусском государственном университете, а

также опыт ведущих зарубежных учебно-научных центров. Учебник включает восемнадцать глав, составляющих две части, список обозначений, приложения, перечень библиографических ссылок и предметный указатель. Часть 1 состоит из девяти глав, представляющих математические и компьютерные основы криптологии. Часть 2, также состоящая из девяти глав, посвящена математическим и компьютерным методам синтеза и анализа криптографических систем защиты информации. Наряду с теоретическим материалом в учебнике приводятся описания многих известных компьютерных криптографических алгоритмов и анализируются их свойства. Представлено более 300 упражнений и заданий для выполнения лабораторного компьютерного практикума. Обращение к источникам, на которые ссылаются авторы, необходимо не только для учебной, но и для научно-исследовательской работы студентов и аспирантов.

Настоящий учебник составляет основу учебно-методического комплекса по дисциплине «Криптология» («Криптографические методы защиты информации») и другим родственным дисциплинам. Он будет полезен при выполнении учебной программы первой и второй ступени, а также специалистам в области прикладной математики, информатики и электроники, желающим познакомиться с математическими и компьютерными методами защиты информации.

Авторство в учебнике распределено следующим образом: Ю. С. Харин – главы 1, 5, 6, 7, 10, 11, 18; С. В. Агиевич – главы 8, 12, 13, 16; Д. В. Васильев – главы 4, 9, 15; Г. В. Матвеев – главы 1, 2, 3, 14, 17.

Авторы благодарны главному научному сотруднику Института математики НАН Беларусь профессору В. И. Бернику за помощь в написании глав 4, 15 и младшему научному сотруднику НИИ прикладных проблем математики, информатики Белорусского государственного университета И. С. Никитиной за подготовку материала для п. 10.2, а также О. А. Куцепаловой – за оформление рукописи в издательской системе \TeX .

Предложения и замечания по содержанию учебника просьба направлять по адресу: кафедра математического моделирования и анализа данных, факультет прикладной математики и информатики, Белорусский государственный университет, пр. Независимости, 4, Минск 220030, Республика Беларусь; тел./факс: +375 17 2095104; e-mail: kharin@bsu.by.

ОСНОВНЫЕ ОБОЗНАЧЕНИЯ

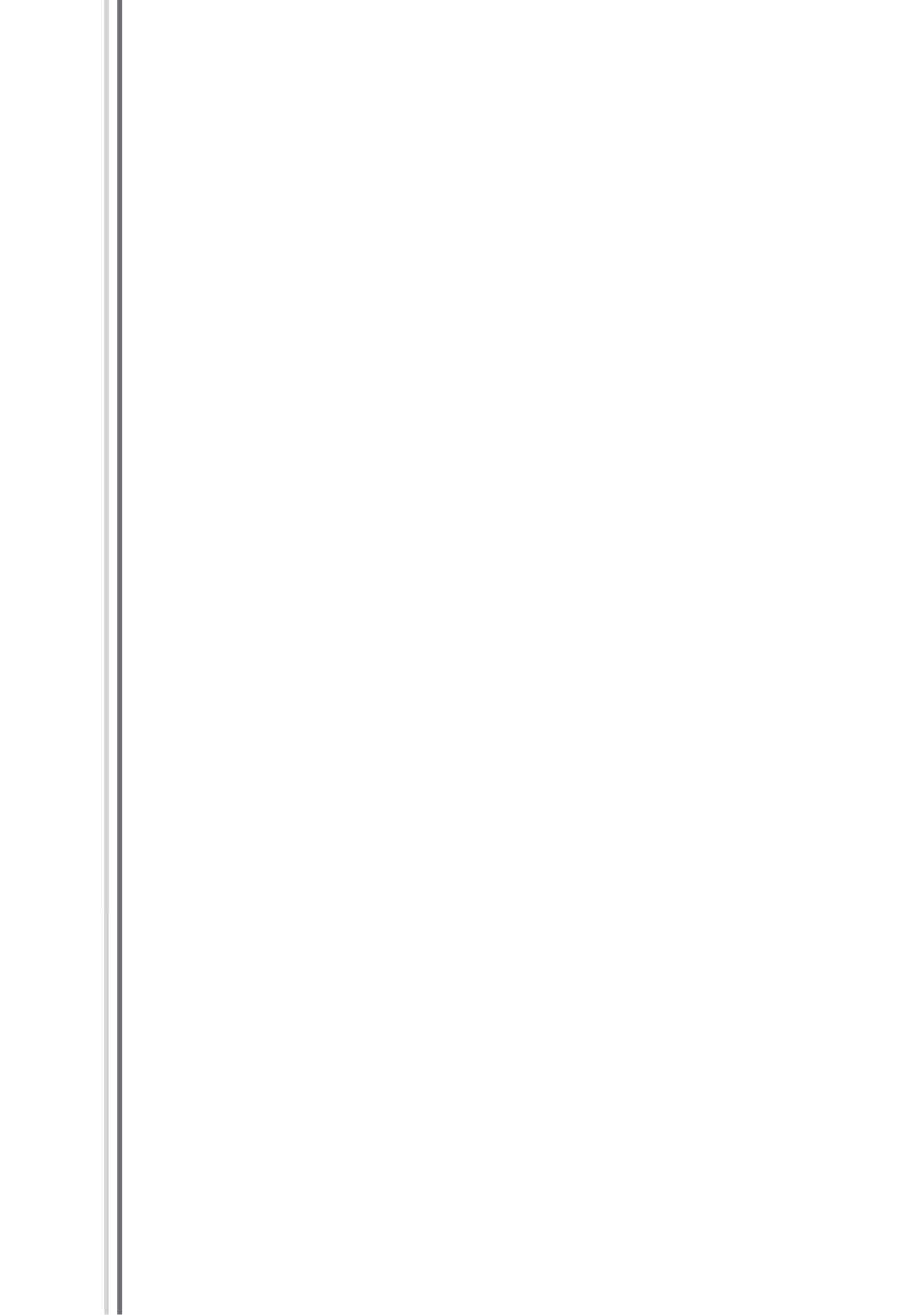
- $|A|$ – модуль числа A , или мощность множества A , или определитель матрицы A
- $S(A)$ – симметрическая группа, действующая на множестве A
- S_n – симметрическая группа степени n
- id – тождественная подстановка
- \mathbb{N} – множество натуральных чисел
- \mathbb{Z} – кольцо целых чисел
- \mathbb{R} – поле вещественных чисел
- A^* – множество слов конечной длины в алфавите A
- A^n – множество слов длины n в алфавите A
- $\mathbb{Z}_n = \mathbb{Z}/(n)$ – кольцо классов вычетов целых чисел по модулю n
- \mathbb{F}_q – конечное поле из q элементов
- V_n – n -мерное векторное пространство над полем \mathbb{F}_2
- $GL_n(\mathbb{F}_q)$ – группа обратимых матриц порядка n над полем \mathbb{F}_q
- \mathcal{F}_n – множество всех булевых функций от n переменных, т. е. отображений $V_n \rightarrow \mathbb{F}_2$
- \oplus – операция сложения в поле \mathbb{F}_2
- $v \lll d$ – циклический сдвиг компонентов вектора $v \in V_n$ на d позиций влево
- \parallel – конкатенация
- $[x], \lfloor x \rfloor$ – целая часть числа $x \in \mathbb{R}$, наибольшее целое $n \leq x$
- $\lceil x \rceil$ – наименьшее целое $n \geq x$
- $\varphi(n)$ – функция Эйлера
- $\mu(n)$ – функция Мёбиуса
- $(a, b) = \text{НОД}(a, b)$ – наибольший общий делитель чисел a и b
- $[a, b] = \text{НОК}(a, b)$ – наименьшее общее кратное чисел a и b

- $\text{ind}_a b$ – индекс (дискретный логарифм) b по основанию a
- $\mathbf{1}(x)$ – единичная функция Хэвисайда
- $\mathbf{I}\{A\}$ – индикатор наступления события A
- $\mathbf{1}_A\{x\} = \mathbf{I}\{x \in A\}$ – индикаторная функция множества A
- $\delta_{ij} = \mathbf{I}\{i = j\}$ – символ Кронекера
- $\mathbf{P}\{A\}$ – вероятность наступления случайного события A
- $\mathbf{E}\{\xi\}$ – математическое ожидание случайной величины ξ
- $\mathbf{D}\{\xi\}$ – дисперсия случайной величины ξ
- $\mathbf{Cov}\{\xi, \eta\}$ – ковариация случайных величин ξ, η
- $\mathcal{L}\{\xi\}$ – закон распределения вероятностей случайной величины ξ
- $\xrightarrow{\text{п.н.}}$ – сходимость почти наверное
- $\xrightarrow{\mathbf{P}}$ – сходимость по вероятности
- $O(\varepsilon), o(\varepsilon)$ – символы Ландау
- $\binom{n}{m}$ – $\frac{n!}{m!(n-m)!}$, если $0 \leq m \leq n$, и 0 – в противном случае
- $u \leftarrow a$ – присвоить значение a переменной u
- $u \xleftarrow{R} U$ – выбрать u случайно равновероятно из множества U
- $u \leftrightarrow v$ – поменять местами значения переменных u и v
- \square – конец доказательства

Часть I

МАТЕМАТИЧЕСКИЕ И КОМПЬЮТЕРНЫЕ ОСНОВЫ КРИПТОЛОГИИ

17xcre28tdmuz8nafn6pxzv3t1pvc34a1tcc109d0
xtqzo5x9vidgsagokn5c403rs29ckd46c29wetpg5
rv00pyo8en9gzw8tj9d9oydwcnukyddags0jp5sou
8j2flfn65zuqjhvxqfoqoabbs0svtsbgcnqdpsbx
lai0deax10dzgqdn2afwhhp9vg9algtwaotlqtupt
zcxml90shmvw527n5anbnzz3xbmzddwhh5j9d90tg
fdbipz7a5z78nw0c0x12s5nkcuuo12g0dp8hy4gqb
zdbe5m3xcqw5hg5qiy538bzgwvuus3cysaq0m8ti1
t8gfqr5h8520hbnrfyxdaf913yuatui370uqcpriv8
g61bgnpu2q0h1qkj13b56feuxew0c6bz6rtss5io7
r4bjwyzlyywblulf16qoax8jos487qbe4136ccq0j
0jzwb2pzsl09g4wqwm0dg2zhm3f1gc9k11uswz4w
ireiaialwgvvluq4v3o3d9kctxgd8hv17hb5z62fr
v5k1czyz83hceqtryywmlm7d033apepref4sm2bfm
ndrchkgxbusv7coauis5z5d7exxy2c7j6p1gousi30
n891cx5kkdixnq7zdo86064xjyn9yhjfbg4vqb1cx
v996n0a28oydslytptt4nm8npvhaeyexdfwsjxdvi
myp1kd049w9gdn6d4x42qy5mh10jwa5mfcmikffqy
ow2k0qttc2k5pwinwijlzeysihs464juxibsvh4m6
qfb5co6ykq360mpz755hnb9fj7iqm49z61nfx8fyw
fh919b4jolqw1ccej01vcc2p7rt07uhzuqoiabnf
qmtmtr0zwo4ajatr27k34gy9cooidapsa1sivdhv
m8n7d2eavmx5i1v05i1f8h10nf19vv2e1vt29wn96



Г л а в а 1

ВВЕДЕНИЕ В КРИПТОЛОГИЮ

1.1. ПРЕДМЕТ КРИПТОЛОГИИ

В современном обществе, имеющем тенденцию к рационализации, информация все отчетливее становится высокоценным товаром, который, как и любой товар, необходимо производить (порождать), хранить (накапливать), обрабатывать (анализировать), транспортировать (передавать) и защищать. В связи с мощным развитием компьютерной индустрии, интенсивным проникновением информационных и компьютерных технологий во все сферы человеческой деятельности и созданием информационного общества (цифрового мира) особенно актуальными становятся проблемы защиты информации.

Условимся вначале о терминологии.

Защита информации – это всевозможные средства и функции, обеспечивающие доступность, конфиденциальность или целостность информации или связи, исключая средства и функции, предохраняющие от неисправностей. Защита информации включает в себя криптографию, криptoанализ, защиту от побочных электромагнитных излучений и наводок используемой аппаратуры и защиту (компьютера) от несанкционированного доступа.

Криптография – раздел прикладной математики и информатики, изучающий модели, методы, алгоритмы, программные и аппаратные средства преобразования информации (шифрования) в целях сокрытия ее содержания, предотвращения видоизменения или несанкционированного использования.

Криптосистема – это система, реализованная программно, аппаратно или программно-аппаратно и осуществляющая криптографическое преобразование информации.

Криptoанализ – раздел прикладной математики и информатики, изучающий модели, методы, алгоритмы, программные и аппаратные средства анализа криптосистемы или ее входных и выходных сигналов с целью извлечения конфиденциальных параметров, включая открытый текст.

Из данных определений видно, что криptoанализ занимается задачами, которые в математическом смысле обратны задачам криптографии. Система криптографии и криptoанализа образует новую науку – *криптологию*.

1.2. ИСТОРИЯ РАЗВИТИЯ КРИПТОЛОГИИ

История возникновения и развития криптологии сложна и увлекательна [17, 32, 36, 60, 68, 94, 115, 138, 162]. В ней выделяются три этапа. *Первый этап* (с древних времен до 1949 г.) характеризуется частными, узкоспециальными и вычислительно простыми алгоритмами криптографии и криптоанализа без использования компьютеров. Его часто называют этапом докомпьютерной криптографии. *Второй этап* (1949–1976 гг.) принято отсчитывать с момента публикации работы американского математика-прикладника К. Шеннона «Теория связи в секретных системах». В этот период активно проводились систематические исследования по криптологии с использованием ЭВМ. Криптология становится математической наукой. Однако потребителями результатов криптологии являлись лишь службы связи и информации в дипломатических и военных организациях, поэтому криптология являлась «закрытой» наукой. *Третий этап* (1976 г. – настоящее время), который можно назвать этапом открытой криптологии, принято отсчитывать с момента публикации работы американских математиков У. Диффи и М. Хеллмана «Новые направления в криптографии». В этой работе показано, что «секретная» передача информации возможна (в отличие от результатов К. Шеннона) без предварительной передачи «секретного ключа». Главной особенностью этого этапа становится массовое применение криптографии в банковском деле, электронной торговле, компьютерных сетях (например, в сети Интернет) и других областях.

Во многих странах в развитие криптологии вкладываются значительные государственные средства. Например, в США ежегодные расходы на криптологию составляют более 20 млрд долл. Характерно также, что криптология стимулирует развитие информатики.

Современная криптология широко использует теорию вероятностей, математическую статистику, алгебру, теорию чисел, теорию алгоритмов и сложности вычислений, а сам процесс шифрования осуществляется на мощных специализированных компьютерах, называемых *устройствами шифрования*. На Западе известны устройства *B-CRYPT*, *IBM-4755*, *Datacryptor* и др.

Приведем ряд интересных фактов из истории развития криптологии. При раскопках в Месопотамии был найден относящийся к XX в. до н. э. один из самых древних шифртекстов. Он был написан клинописью на глиняной дощечке и содержал коммерческую тайну: рецепт глазури для покрытия гончарных изделий. В XVII в. кардинал Ришелье создал первую шифрослужбу. Отметим, что задачами, имеющими непосредственное отношение к криптологии, занимались известные математики И. Ньютон, Г. Лейбниц, Л. Эйлер, К. Ф. Гаусс, А. Н. Колмогоров и др.

1.3. ЗАДАЧИ КРИПТОГРАФИИ И КРИПТОАНАЛИЗА

В криптологии общеприняты следующие понятия.

Пространство сообщений PT – множество всевозможных сообщений pt (plaintext). Для сообщения используется также обозначение m (message).

Пространство ключей K . Каждый ключ $k \in K$ определяет некоторую подстановку E_k (encryption) на пространстве PT и обратное преобразование D_k (decryption).

Пространство зашифрованных сообщений (криптоограмм) CT , состоящее из зашифрованных сообщений ct (ciphertext), $ct = E_k(pt)$. Используется также обозначение c .

Основной задачей специалиста по криптографии (*криптографа*) является построение криптосистемы, к которой предъявляются требования:

- 1) $E_k(pt)$, $D_k(ct)$ – легко вычислимые;
- 2) не зная ключа k , невозможно вычислить pt при известном ct .

В классических криптосистемах «секретный ключ» k определяет преобразования E_k и D_k , причем справедливо тождество $D_k(E_k(pt)) = D_k(ct) = pt$.

Основной задачей специалиста по криptoанализу (*криptoаналитика*) является поиск ключа k . При этом ему могут предоставиться, в частности, следующие возможности для *атаки*:

- 1) он знает лишь зашифрованный текст (Ciphertext only attack);
- 2) известен незашифрованный и зашифрованный текст (Known plaintext attack);
- 3) имеется возможность получить пару $(pt, E_k(pt))$, где pt выбрано криptoаналитиком (Chosen plaintext attack).

Криптография и криptoанализ развиваются параллельно. Криптографы всегда пытаются создать такую криптосистему, которая была бы стойкой ко всем известным методам криptoанализа.

Остается уточнить понятие текста. При этом обычно фиксируют некоторую систему Σ символов, называемую *алфавитом*. Это может быть английский, русский или какой-либо другой алфавит. Часто в качестве алфавита используются натуральные числа или символы 0 и 1. *Словом* называется упорядоченный набор букв данного алфавита. Множество слов обозначают через Σ^* . *Текст* – набор слов.

Одним из самых старых шифров является шифр Юлия Цезаря. При шифровании с его помощью каждая буква латинского алфавита сдвигается циклически вправо на $k = 3$ позиций. Таким образом, имеем подстановку (замену) символов алфавита:

$$\begin{array}{ccccccccccccc} A & B & C & D & E & F & G & H & \dots & T & U & V & W & X & Y & Z \\ D & E & F & G & H & I & J & K & \dots & W & X & Y & Z & A & B & C \end{array}$$

Шифрование осуществляется в соответствии с этой подстановкой. Например, $E_3(CUT) = FXW$. Понятно, что выбор $k = 3$ не является единственным воз-

можным. При других ключах k имеем $E_{25}(IBM) = HAL$, $E_6(IDMUP) = OJSBV$. Нетрудно показать, что $D_k = E_{26-k}$, $D_k E_k = E_0 = D_0$, а ключ k определен по модулю 26. Исключая слабый ключ $k = 0$, пространство ключей имеет мощность $|k| = 25$.

Шифр Цезаря – пример *шифра подстановки или замены*. Его еще называют *шифром простой подстановки*. Можно сказать, что при шифровании простой заменой буквы текста заменяются буквами этого или другого алфавита в соответствии с некоторой подстановкой. Криптоанализ такого шифра очень прост. Дело в том, что для любого современного языка вычислены частотные характеристики букв, т. е. относительные частоты их появления в «нормальных» текстах. Приведем эти частоты в процентах (с упорядочением) для английского языка:

	Высокий уровень		Средний уровень		Низкий уровень
E	12,31	L	4,03	B	1,62
T	9,59	D	3,65	G	1,61
A	8,05	C	3,20	V	0,93
O	7,94	U	3,10	K	0,52
N	7,19	P	2,29	Q	0,20
I	7,18	F	2,28	X	0,20
S	6,59	H	2,25	J	0,10
R	6,03	W	2,03	Z	0,09
H	5,14	Y	1,88		

Еще один пример шифра простой замены – *модулярный шифр*. Выберем число a , взаимно простое с модулем $m=26$. Пусть p – буква английского алфавита, отождествленная со своим порядковым номером $(0, 1, \dots, 25)$. Тогда $E_a(p) = ap + k \pmod{m}$, где k – фиксировано. В этом случае ключом является пара чисел (a, k) . Условие взаимной простоты необходимо для обратимости шифра. Конечно, буквы можно заменять и какими-то другими символами.

К семейству шифров замены относятся гомофонические, полиграммные и многоалфавитные шифры.

Гомофоническое шифрование – один из способов защиты от частотной криптоатаки. Каждая буква текста шифруется несколькими символами этого или другого алфавита. Число этих символов пропорционально частотной характеристике шифруемой буквы. Например:

Буква	Гомофония							
	17	19	34	41	56	60	67	83
A								
I	08	22	53	65	88	90		
L	03	44	76					
N	02	09	15	27	32	40	59	
O	01	11	23	42	54	70	80	
P	33	91						
T	05	10	20	29	45	58	64	78

Одна из возможностей зашифровать текст *PLAIN PILOT*: 91 44 56 65 59 33 08 76 28 78.

При *полиграммном шифровании* заменяются не буквы текста, а их комбинации. Если заменяются пары букв, то имеем *биграммное шифрование*. Примером биграммного шифрования является шифр Плейфера. Образуем из английского алфавита какой-нибудь квадрат 5×5 и будем хранить его, как всякий ключ, в секрете. Например:

H	A	R	P	S
I	C	O	D	B
E	F	G	K	L
M	N	Q	T	U
V	W	X	Y	Z

Здесь буква *J* не употребляется или отождествляется с буквой *I*.

Замена биграмм проводится по следующим правилам:

- 1) если m_1 и m_2 находятся в одной строке, то биграмма $m_1 m_2$ шифруется диграммой $c_1 c_2$, где буквы c_1 и c_2 являются правыми соседями букв m_1 и m_2 соответственно; если правого соседа нет, то берется первая буква строки;
- 2) если m_1 и m_2 – в одном столбце, то берутся нижние соседи с аналогичной оговоркой;
- 3) если $m_1 = m_2$, то в незашифрованном тексте между ними вставляется незначащая буква (например, *X*);
- 4) при нечетном количестве букв в незашифрованном тексте к нему дописывается незначащая буква;
- 5) в наиболее вероятном случае, когда m_1 и m_2 расположены в разных столбцах и строках, c_1 и c_2 выбираются как показано на схеме:

$$\begin{array}{ccccccccc} m_1 & \cdots & c_1 & c_2 & \cdots & m_2 & c_1 & \cdots & m_1 & m_2 & \cdots & c_2 \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ c_2 & \cdots & m_2 & m_1 & \cdots & c_1 & m_2 & \cdots & c_2 & c_1 & \cdots & m_1 \end{array}$$

Покажем это на примере:

$$\begin{aligned} m &= RE \ NA \ IS \ SA \ NC \ EX \\ E(m) &= HG \ WC \ BH \ HR \ WF \ GV \end{aligned}$$

Еще одна биграммная криптосхема, принадлежащая Л. Хиллу, основана на линейной алгебре. Осуществим цифровую кодировку букв английского алфавита: $A = 0, B = 1, C = 2, \dots, Z = 25$. Выберем какую-нибудь обратимую по модулю 26 квадратную матрицу M порядка 2. Это – ключ. Пусть, например,

$$M = \begin{pmatrix} 2 & 5 \\ 3 & 3 \end{pmatrix}, \quad M^{-1} = \begin{pmatrix} 17 & 15 \\ 9 & 20 \end{pmatrix}.$$

Биграммы будем записывать в виде матриц-столбцов. Например,

$$P_1 = \begin{pmatrix} H \\ E \end{pmatrix} = \begin{pmatrix} 7 \\ 4 \end{pmatrix}, \quad P_2 = \begin{pmatrix} L \\ P \end{pmatrix} = \begin{pmatrix} 11 \\ 15 \end{pmatrix}.$$

Шифрование биграмм определим формулой $C = MP$. Зашифруем, для примера, слово $m = P_1 P_2$: $m = HELP$, $c = IHUX$.

К биграммному шифрованию, так же как и к шифру простой подстановки, применима частотная криптоатака. Укажем в связи с этим наиболее распространенные английские биграммы с указанием процентов их встречаемости.

<i>TH</i>	6,3	<i>AK</i>	2,0	<i>HA</i>	1,7
<i>IN</i>	3,1	<i>EN</i>	2,0	<i>OU</i>	1,4
<i>ER</i>	2,7	<i>TI</i>	2,0	<i>IT</i>	1,4
<i>RE</i>	2,5	<i>TE</i>	1,9	<i>ES</i>	1,4
<i>AN</i>	2,2	<i>AT</i>	1,8	<i>ST</i>	1,4
<i>HE</i>	2,2	<i>ON</i>	1,7	<i>OR</i>	1,4

При многоалфавитном подстановочном шифровании задается d шифров простой подстановки, определяемых функциями f_1, f_2, \dots, f_d , а сообщение $m = m_1, m_2, \dots, m_d, m_{d+1}, \dots, m_{2d}, \dots$ шифруется по правилу:

$$E_k(m) = f_1(m_1), f_2(m_2), \dots, f_d(m_d), f_1(m_{d+1}), \dots, f_d(m_{2d}), \dots$$

К таким шифрам относится шифр Виженера. Ключ образуется последовательностью букв k_1, k_2, \dots, k_d , при этом для буквы a i -го алфавита

$$f_i(a) = (a + k_i) \pmod{m}.$$

Например,

$$\begin{aligned} m &= RENA \quad ISSA \quad NCE, \\ k &= BAND \quad BAND \quad BAN, \\ E_k(m) &= SEAD \quad JSFD \quad OCR. \end{aligned}$$

Наряду с подстановочными шифрами известны так называемые *перестановочные (транспозиционные)* шифры. При этом буквы сообщения остаются прежними, но меняют свое расположение в тексте. Приведем два примера.

Сообщение можно разбить на группы букв, скажем, по три буквы, а затем в каждой группе сделать одну и ту же перестановку. Например,

$$LETUSGOTOLONDON \rightarrow ETLSGUTOOONLOND.$$

То же сообщение можно записать в прямоугольнике 3×5 :

<i>L</i>	<i>E</i>	<i>T</i>	<i>U</i>	<i>S</i>
<i>G</i>	<i>O</i>	<i>T</i>	<i>O</i>	<i>L</i>
<i>O</i>	<i>N</i>	<i>D</i>	<i>O</i>	<i>N</i>

а затем переписать его по столбцам: *LGOEONTTDUOOSLN*, что и будет криптограммой (шифртекстом).

В заключение отметим еще один шифр, предложенный Г. Вернамом. Сообщение m обычно записывают в виде последовательности нулей и единиц. Ключ k является реализацией «чисто случайной» двоичной последовательности; длина ключа k равна длине сообщения. Шифрование состоит в применении к m и k операции XOR : $E_k(m) = m \oplus k$.

Очевидно, $D_k = E_k$, так как $(m \oplus k) \oplus k = m \oplus (k \oplus k) = m$. Шифр Вернама считается практически нераскрываемым (теорема Шеннона), так как данное сообщение с помощью подбора ключа, к сожалению, слишком большого, можно преобразовать в любое другое. Основная проблема состоит в хранении и передаче такого ключа.

В современной компьютерной криптографии многие перечисленные выше шифры используются как составные части сложных криптосистем, рассматриваемых в данной книге.

1.4. ЗАДАНИЯ

1. Сколько существует модулярных шифров $p \rightarrow ap + k \pmod{26}$ в английском алфавите?
2. Показать, что для обратимости модулярного шифра необходимо и достаточно условие взаимной простоты $(a, m) = 1$. Как найти обратное преобразование? Будет ли оно модулярным шифром?
3. Описать алгоритм D для шифра Плейфера.
4. Описать алгоритм D для шифра Виженера и найти соответствующий ключ.
5. Какие докомпьютерные шифры являются групповыми (замкнуты относительно операции повторного шифрования), т. е. для любых двух ключей k_1, k_2 существует ключ k_3 такой, что $E_{k_2}E_{k_1} = E_{k_3}$?
6. Пусть словами являются булевы n -ки, т. е. $m \in V_n$. Сформулировать условие перестановочности шифров перестановки и Вернама.
7. Показать, что шифр перестановки является линейным преобразованием в пространстве V_n .
8. Пусть P – криптопреобразование простой перестановки. При каком условии верно равенство $P^2 = id$?
9. Что для криптографии означает условие $E^2 = id$?
10. Пусть P – перестановка, а V – преобразование Вернама. Какая система шифрования предпочтительнее: PV или VP ?
11. При каком условии перестановочное шифрование с помощью прямоугольника будет тождественным?
12. Сколько всего нелинейных криптопреобразований $E: V_3 \rightarrow V_3$?

Г л а в а 2

АРИФМЕТИЧЕСКИЕ ОСНОВЫ

2.1. АЛГОРИТМ ДЕЛЕНИЯ С ОСТАТКОМ

Множество всех натуральных чисел $1, 2, 3, \dots$ будем обозначать через \mathbb{N} , а через \mathbb{Z} – множество целых чисел $0, \pm 1, \pm 2, \dots$.

Пусть a, b – элементы из \mathbb{Z} , $a \neq 0$. Говорят, что a делит b (записывают $a | b$), если существует такое целое c , что $b = ac$. Это отношение рефлексивно, транзитивно, но не симметрично. Легко доказываются следующие свойства.

Свойство 2.1. $a | b, a | c \Rightarrow a | b \pm c$.

Свойство 2.2. $a | b \Rightarrow a | bc, c \in \mathbb{Z}$.

Свойство 2.3. $a | b, b | a \Leftrightarrow a = \pm b$.

Теорема 2.1. Пусть a – целое, b – натуральное. Тогда существуют такие однозначно определенные $q, r \in \mathbb{Z}, 0 \leq r < b$, что $a = bq + r$.

Доказательство. Возьмем наибольшее bq , не превосходящее a . Положим $r = a - bq$. Условие $b(q+1) > a$ влечет $r < b$. Допустив также, что $a = bq_1 + r_1, 0 \leq r_1 < b$, получим $0 = b(q - q_1) + r - r_1$. Отсюда следует, что $r - r_1$ кратно b . Но так как $|r - r_1| < b$, то последнее возможно лишь при $r - r_1 = 0$, т. е. при $r = r_1$, откуда вытекает также $q = q_1$. \square

Теорема справедлива для любого целого $b \neq 0$ при условии, что ограничение $r < b$ заменяется на $r < |b|$.

2.2. НАИБОЛЬШИЙ ОБЩИЙ ДЕЛИТЕЛЬ

Всякое целое, делящее числа a и b , называется их *общим делителем*. Наибольший из общих делителей для чисел a и b называется *наибольшим общим делителем (НОД)* и обозначается (a, b) . Ввиду конечности числа делителей данного числа, существование и единственность наибольшего общего делителя очевидны. Если $(a, b) = 1$, то числа a и b называются *взаимно простыми*. Найдем НОД двух чисел.

Лемма 2.1. $b | a \Rightarrow (a, b) = b$.

Доказательство. Легко видеть, что совокупность общих делителей a и b совпадает с совокупностью делителей b . \square

Лемма 2.2. $a = bq + c \Rightarrow (a, b) = (b, c)$.

Доказательство. По свойствам делимости из п. 2.1 у пары a, b те же общие делители, что и у пары b, c . \square

Эти две леммы лежат в основе алгоритма нахождения наибольшего общего делителя (алгоритм Евклида). Выполним следующие деления с остатком:

$$\begin{aligned} a &= bq_1 + r_1, & 0 \leq r_1 < b, \\ b &= r_1 q_2 + r_2, & 0 \leq r_2 < r_1, \\ r_1 &= r_2 q_3 + r_3, & 0 \leq r_3 < r_2, \\ \dots &\dots \dots & \dots \dots \dots \\ r_{n-2} &= r_{n-1} q_n + r_n, & 0 \leq r_n < r_{n-1}, \\ r_{n-1} &= r_n q_{n+1}. \end{aligned} \tag{2.1}$$

Тогда $(a, b) = (b, r_1) = (r_1, r_2) = \dots = (r_{n-1}, r_n) = r_n$. Последнее равенство выполняется в силу леммы 2.1, а все предыдущие – в силу леммы 2.2.

Таким образом, НОД двух чисел равен последнему отличному от нуля остатку в алгоритме Евклида.

Пример 2.1. Применим алгоритм Евклида к нахождению $(175, 77)$:

$$175 = 77 \cdot 2 + 21, \quad 77 = 21 \cdot 3 + 14, \quad 21 = 14 \cdot 1 + 7, \quad 14 = 7 \cdot 2.$$

Последний положительный остаток есть $r_3 = 7$. Значит, $(175, 77) = 7$.

Умножая все равенства (2.1) на натуральное m , убеждаемся в справедливости свойства $(am, bm) = (a, b)m$. Пусть δ – делитель a и b . Тогда

$$(a, b) = \left(\frac{a}{\delta}, \frac{b}{\delta} \right) \delta = \left(\frac{a}{\delta}, \frac{b}{\delta} \right) \delta.$$

Отсюда получаем

$$\left(\frac{a}{\delta}, \frac{b}{\delta} \right) = \frac{(a, b)}{\delta}, \quad \left(\frac{a}{(a, b)}, \frac{b}{(a, b)} \right) = 1. \tag{2.2}$$

Понятие наибольшего общего делителя можно ввести и для нескольких чисел a_1, a_2, \dots, a_n . Его обозначают (a_1, a_2, \dots, a_n) . Наибольший общий делитель нескольких чисел можно вычислять последовательно. Например, $(a_1, a_2, a_3) = ((a_1, a_2), a_3)$, $(a_1, a_2, a_3, a_4) = ((a_1, a_2, a_3), a_4)$ и т. д.

2.3. ВЗАИМНО ПРОСТЫЕ ЧИСЛА

Теорема 2.2 (критерий взаимной простоты чисел). Два целых a и b будут взаимно простыми тогда и только тогда, когда найдутся целые u и v такие, что $au + bv = 1$.

Доказательство. Необходимость вытекает из алгоритма Евклида. Из равенства $r_{n-2} = r_{n-1}q_n + r_n$ находим линейное выражение наибольшего общего делителя $r_n = r_{n-1}q_n + r_{n-2}$ через предыдущие остатки. Затем вставляем сюда аналогичное выражение для r_{n-1} и т. д. В итоге получается так называемое *линейное разложение НОД*: $d = au + bv$. Достаточность вытекает из свойств делимости:

$$(a, b) \mid a, (a, b) \mid b \Rightarrow (a, b) \mid au + bv = 1 \Rightarrow (a, b) = 1.$$

□

Из доказанного критерия взаимной простоты вытекают следующие свойства.

Свойство 2.4. $(a, b) = 1, (a, c) = 1 \Rightarrow (a, bc) = 1$.

Свойство 2.5. $a \mid bc, (a, b) = 1 \Rightarrow a \mid c$.

Свойство 2.6. $a \mid c, b \mid c, (a, b) = 1 \Rightarrow ab \mid c$.

Например, первое из них выводится так. Умножим почленно равенство $au + bv = 1$ на c . Получим $acu + bcv = c$. Если теперь натуральное d делит a и bc , то оно должно делить и c . Поэтому $d = 1$.

2.4. РАСШИРЕННЫЙ АЛГОРИТМ ЕВКЛИДА

Расширенный алгоритм Евклида – это еще один способ нахождения линейного разложения НОД: $d = au + bv$. От обычного алгоритма он отличается тем, что наряду с остатками r_i и неполными частными q_i вычисляются еще две вспомогательные последовательности x_i и y_i .

Сначала полагают

$$r_{-1} = a, \quad r_0 = b;$$

$$x_{-1} = 1, \quad y_{-1} = 0;$$

$$x_0 = 0, \quad y_0 = 1.$$

Затем последовательно находят

$$r_i = r_{i-2} - q_i r_{i-1},$$

$$x_i = x_{i-2} - q_i x_{i-1},$$

$$y_i = y_{i-2} - q_i y_{i-1}.$$

Значения x_k и y_k , при которых $d = r_k$, и будут искомыми u и v .

Это видно из следующей простой теоремы.

Теорема 2.3. Для любого i , $-1 \leq i \leq k$, выполняется равенство

$$ax_i + by_i = r_i.$$

Доказательство. Применим индукцию по i . При $i = -1, 0$ доказывать нечего. Пусть теперь все равенства доказаны для всех индексов, меньших i . Тогда, пользуясь индуктивным предположением, получим

$$\begin{aligned} ax_i + by_i &= a(x_{i-2} - q_i x_{i-1}) + b(y_{i-2} - q_i y_{i-1}) = \\ &= (ax_{i-2} + by_{i-2}) - q_i(ax_{i-1} + by_{i-1}) = r_{i-2} - q_i r_{i-1} = r_i. \end{aligned}$$

2.5. НАИМЕНЬШЕЕ ОБЩЕЕ КРАТНОЕ

Если $a | M, b | M$, то число $M \in \mathbb{N}$ называют *общим кратным* целых чисел $a, b \in \mathbb{Z}$. *Наименьшее общее кратное (НОК)* чисел a и b принято обозначать $[a, b]$.

Теорема 2.4. *Если M – общее кратное целых a и b , то $[a, b] | M$.*

Доказательство. Разделим с остатком M на $[a, b]$: $M = [a, b]q + r$. Ввиду того, что $r = M - [a, b]q$ и $a | M, a | [a, b]$, имеем $a | r$. Аналогично, $b | r$. Поскольку $0 \leq r < [a, b]$, то это возможно лишь при $r = 0$.

Теорема 2.5. *Справедливо соотношение $[a, b] = ab/(a, b)$.*

Доказательство. Пусть сначала $(a, b) = 1$. Тогда по свойству 2.6 $a | [a, b], b | [a, b] \Rightarrow ab | [a, b] \Rightarrow [a, b] = ab$. Пусть $(a, b) = d \neq 1$. В этом случае $(a/d, b/d) = 1$. Следовательно, $[a/d, b/d] = ab/d^2$. Если два числа умножить на d , то их НОК и НОД также приобретут этот множитель. Поэтому $[a, b] = dab/d^2 = ab/(a, b)$.

Наименьшее общее кратное нескольких чисел можно вычислять последовательно.

Так, $[a_1, a_2, a_3] = [[a_1, a_2], a_3], [a_1, a_2, a_3, a_4] = [[a_1, a_2, a_3], a_4]$ и т. д.

2.6. ПРОСТЫЕ ЧИСЛА

Натуральное число $p > 1$ называется *простым*, если оно не имеет других натуральных делителей, кроме 1 и p . Простым числом будет наименьший отличный от единицы делитель целого $a, a > 1$.

Теорема 2.6 (теорема Евклида). *Существует бесконечно много простых чисел.*

Доказательство. Пусть p_1, p_2, \dots, p_n – различные простые числа. Простой делитель числа $p_1 p_2 \cdots p_n + 1$ не может совпадать ни с одним из чисел p_1, p_2, \dots, p_n . \square

Отметим еще несколько свойств простых чисел (p – простое).

Свойство 2.7. $(p, a) \neq 1 \Rightarrow p | a$.

Действительно, (p, a) , будучи делителем p , может быть равен или 1, или p . Следовательно, $(p, a) = p \Rightarrow p | a$.

Свойство 2.8. $p \mid ab \Rightarrow p \mid a$ либо $p \mid b$.

Если бы имело место $(p, a) = (p, b) = 1$, то по свойству 2.4 $(p, ab) = 1$. Поэтому хотя бы один из множителей делится на p . Свойство легко обобщается на случай нескольких чисел a, b, c, \dots .

Теорема 2.7. Всякое целое, большее единицы, разложимо в произведение простых множителей. Это разложение единствено с точностью до порядка следования множителей.

Доказательство. Если целое a не простое, то $a = p_1 a_1$, где p_1 – наименьший простой делитель. В случае, если a_1 не простое, то $a_1 = p_2 a_2$. В итоге придем к случаю $a_n = 1$. Следовательно,

$$a = p_1 p_2 \cdots p_n. \quad (2.3)$$

Пусть есть еще одно разложение $a = q_1 q_2 \cdots q_s$. Тогда $p_1 p_2 \cdots p_n = q_1 q_2 \cdots q_s$. По свойству 2.8 p_1 делит хотя бы одно из чисел в правой части. Пусть, например, $p_1 \mid q_1$. Это значит, $p_1 = q_1$. Сокращая обе части на p_1 , имеем $p_2 p_3 \cdots p_n = q_2 q_3 \cdots q_s$. Аналогичным образом доказывают совпадение p_2 с каким-либо множителем в правой части. В итоге получим тождественность двух разложений. \square

В разложении (2.3) некоторые множители могут повторяться. Если объединить повторения, то получается так называемое *каноническое разложение* числа a на простые множители:

$$a = p_1^{k_1} p_2^{k_2} \cdots p_s^{k_s}. \quad (2.4)$$

2.7. СРАВНЕНИЯ

Будем рассматривать целые числа в связи с остатками от их деления на натуральное m , называемое *модулем*. Если два целых a и b имеют одинаковые остатки от деления на m , то они называются *сравнимыми по модулю m* . Сравнимость чисел a и b записывают в виде

$$a \equiv b \pmod{m}.$$

Отметим следующие легко доказываемые либо очевидные свойства.

Свойство 2.9. $a \equiv b \pmod{m} \Leftrightarrow m \mid a - b$.

Свойство 2.10. $a \equiv b \pmod{m}, b \equiv c \pmod{m} \Rightarrow a \equiv c \pmod{m}$.

Действительно, $m \mid a - b, m \mid b - c \Rightarrow m \mid ((a - b) + (b - c)) \Rightarrow m \mid a - c$.

Свойство 2.11. Сравнения можно почленно складывать.

Пусть $a_1 \equiv b_1 \pmod{m}, a_2 \equiv b_2 \pmod{m}$. Тогда $m \mid a_1 - b_1, m \mid a_2 - b_2$.

В силу свойства 2.1 $m \mid a_1 - b_1 + a_2 - b_2$, т. е. $m \mid (a_1 + a_2) - (b_1 + b_2)$.

Свойство 2.12. Сравнения можно почленно перемножать.

Пусть $a_1 = mq_1 + r_1$, $b_1 = mq_2 + r_1$, $a_2 = mq_3 + r_2$, $b_2 = mq_4 + r_2$. Тогда $a_1a_2 = m(mq_2 + q_1r_1 + q_2r_1) + r_1r_2$, т. е. $a_1a_2 \equiv r_1r_2 \pmod{m}$. Аналогично, $b_1b_2 \equiv r_1r_2 \pmod{m}$.

Свойство 2.13. К обеим частям сравнения можно прибавить одно и то же число.

Свойство 2.14. Обе части сравнения можно умножить на одно и то же число.

Свойство 2.15. Обе части сравнения можно разделить на их общий делитель, если он взаимно прост с модулем.

Пусть $a_1d \equiv b_1d \pmod{m}$. Тогда $m \mid d(a_1 - b_1)$. В силу свойства 2.5 $m \mid a_1 - b_1$, поскольку $(m, d) = 1$.

Свойство 2.16. Обе части сравнения и модуль можно сокращать на их общий делитель.

Свойство 2.17. $a \equiv b \pmod{m_1}$, $a \equiv b \pmod{m_2} \Rightarrow a \equiv b \pmod{[m_1, m_2]}$.

Вытекает из теоремы 2.4.

Свойство 2.18. $a \equiv b \pmod{m}$, $d \mid b$, $d \mid m \Rightarrow d \mid a$.

Действительно, $m \mid a - b \Rightarrow d \mid a - b \Rightarrow d \mid a - b + b \Rightarrow d \mid a$.

Свойство 2.19. $a \equiv b \pmod{m} \Rightarrow (a, m) = (b, m)$.

Ввиду $a = b + m$, непосредственно следует из леммы 2.2.

2.8. КЛАССЫ ВЫЧЕТОВ

Числа, сравнимые по модулю m , образуют *класс вычетов по модулю m* . Все числа из одного класса имеют один и тот же остаток r от деления на m . Любое число a из класса вычетов называется *вычетом по модулю m* . Соответствующий класс обозначается через \bar{a} . Поскольку отношение $a \equiv b \pmod{m}$ является бинарным отношением эквивалентности, то имеем разбиение целых чисел на *классы эквивалентности (классы вычетов)*. Всего имеется m классов вычетов по модулю m : $\overline{0}, \overline{1}, \overline{2}, \dots, \overline{m-1}$.

Свойство 2.20. $a \equiv b \pmod{m} \Leftrightarrow \bar{a} = \bar{b}$.

Свойство 2.21. $a \not\equiv b \pmod{m} \Leftrightarrow \bar{a} \cap \bar{b} = \emptyset$.

Взяв из каждого класса по одному вычету, получим *полную систему вычетов*. Например, наряду с $0, 1, 2, \dots, m-1$ полной системой вычетов будет $1, 2, \dots, m$.

Свойство 2.22. Любые m чисел, попарно несравнимые по модулю m , образуют полную систему вычетов.

Свойство 2.23. Если $(a, m) = 1$ и x пробегает полную систему вычетов по модулю m , то $ax + b$, где b – любое целое, также пробегает полную систему вычетов по модулю m .

Доказательство. В силу свойства 2.22 достаточно проверить, что $ax_1 + b \not\equiv ax_2 + b \pmod{m}$ при $x_1 \not\equiv x_2 \pmod{m}$. Допустив $ax_1 + b \equiv ax_2 + b \pmod{m}$, получим $ax_1 \equiv ax_2 \pmod{m} \Rightarrow x_1 \equiv x_2 \pmod{m}$. \square

Согласно свойству 2.19 числа одного класса вычетов имеют с модулем m один и тот же общий делитель. Рассмотрим те классы, для которых этот делитель равен единице. Взяв от каждого такого класса по одному вычету, получим приведенную систему вычетов. Например, приведенная система по модулю 42 будет следующей: 1, 5, 11, 13, 17, 19, 23, 25, 29, 31, 37, 41.

Свойство 2.24. Если $(a, m) = 1$ и x пробегает приведенную систему вычетов по модулю m , то ax также будет пробегать приведенную систему вычетов по модулю m .

Достаточно показать, что числа ax попарно несравнимы (2.15) и взаимно просты с модулем m . Второе следует из $(m, a) = 1$, $(m, x) = 1$.

2.9. ФУНКЦИЯ ЭЙЛЕРА

Количество классов вычетов в приведенной системе вычетов обозначают через $\varphi(m)$ и называют функцией Эйлера. Она определена для всех натуральных чисел и представляет собой количество чисел ряда $0, 1, \dots, a - 1$ $(1, 2, \dots, a)$, взаимно простых с a .

Примеры. $\varphi(1) = 1$, $\varphi(2) = 1$, $\varphi(3) = 2$, $\varphi(4) = 2$, $\varphi(5) = 4$, $\varphi(6) = 2$. Очевидны следующие свойства (p – простое).

Свойство 2.25. $\varphi(p) = p - 1$.

Свойство 2.26. $\varphi(p^k) = p^k - p^{k-1}$, $k \in \mathbb{N}$.

При $k = 1$ все ясно. Пусть $k > 1$. Тогда в ряду $i = 1, 2, 3, \dots, p^k$ условие $(i, p^k) = 1$ нарушается лишь для каждого p -го члена. Всего их $p^k/p = p^{k-1}$.

Лемма 2.3 (мультипликативность функции Эйлера).

$$(a, b) = 1 \Rightarrow \varphi(ab) = \varphi(a)\varphi(b).$$

Доказательство. Поместим числа $1, 2, \dots, ab$ в таблицу:

1	2	3	...	b
$b + 1$	$b + 2$	$b + 3$...	$2b$
$2b + 1$	$2b + 2$	$2b + 3$...	$3b$
...
$(a - 1)b + 1$	$(a - 1)b + 2$	$(a - 1)b + 3$...	ab

Числа, взаимно простые с b , могут быть лишь в столбцах, номера которых взаимно простые с b . Все числа такого столбца взаимно простые с b . Таких столбцов всего $\varphi(b)$. По свойству 2.23 любой такой столбец представляет собой полную систему вычетов по модулю a . Поэтому он содержит $\varphi(a)$ чисел,

взаимно простых с a . Воспользуемся тем, что $(i, ab) = 1 \Leftrightarrow (i, a) = (i, b) = 1$ (свойство 2.4). Следовательно, $\varphi(ab) = \varphi(a)\varphi(b)$. \square

Используя лемму и каноническое разложение числа на простые множители $a = p_1^{k_1} p_2^{k_2} \cdots p_s^{k_s}$, имеем

$$\varphi(a) = \left(p_1^{k_1} - p_1^{k_1-1} \right) \left(p_2^{k_2} - p_2^{k_2-1} \right) \cdots \left(p_s^{k_s} - p_s^{k_s-1} \right)$$

или

$$\begin{aligned} \varphi(a) &= p^{k_1} p^{k_2} \cdots p^{k_s} \left(1 - \frac{1}{p_1} \right) \left(1 - \frac{1}{p_2} \right) \cdots \left(1 - \frac{1}{p_s} \right), \\ \varphi(a) &= a \prod_{p|a} \left(1 - \frac{1}{p} \right). \end{aligned}$$

Функция Эйлера используется в теории сравнений.

Теорема 2.8 (теорема Ферма). $a^{p-1} \equiv 1 \pmod{p}$, если $(a, p) = 1$.

Теорема 2.9 (теорема Эйлера). $a^{\varphi(m)} \equiv 1 \pmod{m}$, если $(a, m) = 1$.

Первая теорема вытекает из второй, которая будет доказана в следующей главе.

2.10. СРАВНЕНИЯ ПЕРВОЙ СТЕПЕНИ

Рассмотрим сравнение

$$ax \equiv b \pmod{m} \tag{2.5}$$

при условии $(a, m) = 1$. Под *решением* любого сравнения понимают класс вычетов по модулю m , один элемент которого (а значит, и все) удовлетворяет сравнению. В нашем случае найдутся целые u, v такие, что $au + mv = 1$. Следовательно, $au \equiv 1 \pmod{m}$. Будем называть u *обратным к* a *по модулю* m . Умножим обе части сравнения (2.5) на u . Получим

$$x \equiv bu \pmod{m}. \tag{2.6}$$

Следовательно, сравнение имеет единственное решение по модулю m .

Пусть $(a, m) = d > 1$. По свойству 2.18 условие $d | b$ является необходимым условием разрешимости сравнения (2.5). Будем считать его выполненным. Пусть $a = a_1d$, $b = b_1d$, $m = m_1d$. Тогда наше сравнение равносильно $a_1x \equiv b_1 \pmod{m_1}$. Имеем одно решение $x \equiv x_1 \pmod{m_1}$. По модулю m имеем d решений: $x_1, x_1 + m_1, x_1 + 2m_1, \dots, x_1 + (d-1)m_1$.

Теорема 2.10. Пусть $(a, m) = d$. Сравнение $ax \equiv b \pmod{m}$ разрешимо тогда и только тогда, когда $d | b$. В этом случае оно имеет d решений.

При небольшом m сравнение $ax \equiv b \pmod{m}$ решается подбором. Для этого достаточно найти число u такое, что $au \equiv 1 \pmod{m}$; это можно сделать с помощью алгоритма Евклида. В качестве u можно также взять $u = a^{\varphi(m)-1}$ (способ Эйлера).

2.11. СИСТЕМА СРАВНЕНИЙ ПЕРВОЙ СТЕПЕНИ

Система сравнений

$$\left\{ \begin{array}{l} a_1x \equiv b_1 \pmod{m_1}, \\ a_2x \equiv b_2 \pmod{m_2}, \\ \dots \dots \dots \\ a_nx \equiv b_n \pmod{m_n} \end{array} \right. \quad (2.7)$$

сводится к системе вида

$$\left\{ \begin{array}{l} x \equiv b_1 \pmod{m_1}, \\ x \equiv b_2 \pmod{m_2}, \\ \dots \dots \dots \\ x \equiv b_n \pmod{m_n}. \end{array} \right. \quad (2.8)$$

Для решения последней достаточно уметь решать систему

$$\left\{ \begin{array}{l} x \equiv b_1 \pmod{m_1}, \\ x \equiv b_2 \pmod{m_2}. \end{array} \right. \quad (2.9)$$

Из первого сравнения $x = b_1 + m_1t$. Подставляя x во второе сравнение, получаем $m_1t \equiv b_2 - b_1 \pmod{m}$. Поэтому критерием разрешимости (2.9) является условие $(m_1, m_2) | b_2 - b_1$. В этом случае имеем одно решение по модулю $\frac{m_2}{(m_1, m_2)}$: $t \equiv t_0 \left(\pmod{\frac{m_2}{(m_1, m_2)}} \right)$. Поэтому

$$x = b_1 + m_1 \left(t_0 + \frac{m_2}{(m_1, m_2)} l \right) = b_0 + \frac{m_1 m_2}{(m_1, m_2)} t = b_0 + [m_1, m_2]t.$$

Таким образом, система (2.9) в случае ее разрешимости имеет единственное решение по модулю $[m_1, m_2]$. Исходная система (2.7) в случае ее разрешимости имеет единственное решение по модулю $[m_1, m_2, \dots, m_n]$.

В случае, когда все модули m_1, m_2, \dots, m_n попарно взаимно простые, к системе (2.8) применим так называемый *китайский способ*. Определим числа x_1, x_2, \dots, x_n из условий $m_2 m_3 \cdots m_n x_1 \equiv 1 \pmod{m_1}$, $m_1 m_3 \cdots m_n x_2 \equiv 1 \pmod{m_2}$, \dots , $m_1 m_2 \cdots m_{n-1} x_n \equiv 1 \pmod{m_n}$. Тогда решением системы (2.8) будет число

$$x = m_2 m_3 \cdots m_n x_1 b_1 + m_1 m_3 \cdots m_n x_2 b_2 + \dots + m_1 m_2 \cdots m_{n-1} x_n b_n.$$

Теорема 2.11 (китайская теорема об остатках). *Система сравнений (2.8) при попарно взаимно простых модулях имеет единственное решение по модулю произведения.*

2.12. ПЕРВООБРАЗНЫЕ КОРНИ

Говорят, что число a , взаимно простое с модулем m , *принадлежит показателю* δ , если δ – такое наименьшее натуральное число, что выполняется сравнение $a^\delta \equiv 1 \pmod{m}$. Справедливы следующие свойства.

Свойство 2.27. Числа $a^0, a^1, \dots, a^{\delta-1}$ попарно несравнимы по модулю m .

Действительно, $a^l \equiv a^k \pmod{m}$, $l > k \Rightarrow a^{l-k} \equiv 1 \pmod{m}$, где $l - k \in \mathbb{N}$, $l - k < \delta$.

Свойство 2.28. $a^\gamma \equiv a^{\gamma'} \pmod{m} \Leftrightarrow \gamma \equiv \gamma' \pmod{\delta}$.

Разделим γ, γ' на δ с остатками $\gamma = \delta q + r, \gamma' = \delta q' + r'$. Тогда $a^\gamma \equiv a^{\gamma'} \Leftrightarrow a^{\delta q+r} \equiv a^{\delta q'+r'} \Leftrightarrow a^r \equiv a^{r'} \Leftrightarrow r' = r$. Отсюда вытекает следующее свойство:

Свойство 2.29. $\delta \mid \varphi(m)$.

Число, принадлежащее показателю $\varphi(m)$, называется *первообразным корнем* по модулю m .

Свойство 2.30. По любому простому модулю p существует первообразный корень.

Это свойство будет доказано в следующей главе. К. Ф. Гауссом доказано существование первообразных корней по модулям p^k и $2p^k$ при любом нечетном простом p . Легко убедиться, что при $m = 4$ первообразный корень также существует. Таким образом, первообразные корни существуют по модулям $2, 4, p^\alpha, 2p^\alpha$, где p – нечетное простое, $\alpha \in \mathbb{N}$.

Докажем отсутствие первообразных корней по всем остальным модулям. Если m не является степенью 2, $m = m_1 m_2$, $m_1 > 2, m_2 > 2, (m_1, m_2) = 1$, то $\varphi(m_1) \equiv \varphi(m_2) \equiv 0 \pmod{2}$. Поэтому

$$a^{\varphi(m)/2} = (a^{\varphi(m_1)})^{\varphi(m_2)/2} \equiv 1 \pmod{m},$$

аналогично, $a^{\varphi(m)/2} \equiv 1 \pmod{m_2}$. Следовательно, $a^{\varphi(m)/2} \equiv 1 \pmod{m}$ ввиду свойства 2.17. Если $m = 2^i$, то при нечетном a $a^2 \equiv 1 \pmod{8}$. Следовательно, $a^{2^{i-2}} \equiv 1 \pmod{2^i}$.

Нахождение первообразных корней упрощает следующее свойство.

Свойство 2.31. Пусть $c = \varphi(m)$ и q_1, q_2, \dots, q_k – различные простые делители числа c . Число a , взаимно простое с модулем m , будет первообразным корнем тогда и только тогда, когда не выполнено ни одно из следующих сравнений:

$$a^{c/q_1} \equiv 1 \pmod{m}, a^{c/q_2} \equiv 1 \pmod{m}, \dots, a^{c/q_k} \equiv 1 \pmod{m}. \quad (2.10)$$

Необходимость следует из того, что $a^{\varphi(m)} \equiv 1 \pmod{m}$ и сравнение не имеет места при меньших показателях степени. Обратно: допустим, что a не

удовлетворяет ни одному из сравнений и a принадлежит показателю $\delta < c$. Тогда $\delta | c \Rightarrow c = \delta p$. Обозначим через q простой делитель u . Тогда легко получить противоречие

$$a^{c/q} = a^{\delta u/q} = (a^\delta)^{u/q} \equiv 1 \pmod{m}.$$

Пример 2.2. Пусть $m = 41$. Имеем $c = \varphi(41) = 40 = 2^3 \cdot 5$. Итак, первообразный корень не должен удовлетворять двум сравнениям

$$a^8 \equiv 1 \pmod{41}, \quad a^{20} \equiv 1 \pmod{41}.$$

Испытываем числа $2, 3, 4, \dots : 2^8 \equiv 10, 2^{20} \equiv 1, 3^8 \equiv 1, 4^8 \equiv 18, 4^{20} \equiv 1, 5^8 \equiv 18, 5^{20} \equiv 1, 6^8 \equiv 10, 6^{20} \equiv 40$. Отсюда видим, что 6 является наименьшим первообразным корнем по модулю 41.

2.13. СУЩЕСТВОВАНИЕ ПЕРВООБРАЗНЫХ КОРНЕЙ

Теорема 2.12. Пусть p – нечетное простое. Тогда по модулям вида p^k и $2p^k$, $k \geq 1$, существуют первообразные корни.

Доказательство. Пусть a – первообразный корень по модулю p . Покажем, что существует такое целое x , что $a' = a + px$ будет первообразным корнем сразу по всем модулям вида p^k , $k > 1$. Поскольку a – первообразный корень по модулю p , то $a^{p-1} = 1 + py$ при некотором целом y . Применим формулу бинома Ньютона:

$$\begin{aligned} (a')^{p-1} &= (a + px)^{p-1} = a^{p-1} + (p-1)a^{p-2}px + \dots = \\ &= 1 + py + (p-1)a^{p-2}px + \dots = 1 + p(y + (p-1)a^{p-2}x + \dots). \end{aligned}$$

В скобках коэффициент при x не делится на p . Поэтому x можно выбрать так, что содержимое скобок не будет делиться на p .

В этом случае $(a')^{p-1} = 1 + pz$, где $(z, p) = 1$. Пусть a' принадлежит показателю d по модулю p^k . Тогда d делит $p^{k-1}(p-1)$.

С другой стороны, $p-1$ делит d . Поэтому $d = p^l(p-1)$ при некотором $l < k$. Далее,

$$(a')^d = (1 + pz)^{p^l} = 1 + p^{l+1}u, \quad (u, p) = 1.$$

Это означает, что $l = k-1$, т. е. $d = p^{k-1}(p-1) = \varphi(p^k)$.

Рассмотрим модуль $2p^k$. Возьмем первообразный корень a по модулю p^k . Одно из чисел, a или $a + p^k$, будет нечетным. Оно также является и первообразным корнем по модулю $2p^k$, так как $\varphi(2p^k) = \varphi(p^k)$. \square

Таким образом, первообразные корни существуют лишь по модулям 2, 4, p^k , $2p^k$, где p – простое нечетное.

2.14. ИНДЕКСЫ ПО МОДУЛЯМ p^k И $2p^k$

Обозначим через m модуль вида p^k или $2p^k$, а через g – первообразный корень по этому модулю. Пусть $c = \varphi(m)$. Из свойства 2.27 вытекают следующие свойства:

Свойство 2.32. Если число γ принимает последовательно значения $0, 1, \dots, c - 1$, то g^γ пробегает приведенную систему вычетов по модулю m .

Для чисел a , взаимно простых с m , введем понятие индекса, называемого иногда *дискретным логарифмом*.

Пусть $a \equiv g^\gamma \pmod{m}$. Число γ ($\gamma \geq 0$) называется *индексом числа a по модулю m при основании g* . Используются обозначения $\gamma = \text{ind}_g a$ или $\gamma = \text{ind } a$. В силу теоремы Эйлера индекс определен по модулю c . Тем самым было бы правильнее говорить о классе вычетов по модулю c .

Свойство 2.33. $\text{ind } ab \equiv \text{ind } a + \text{ind } b \pmod{c}$.

Перемножая сравнения $a \equiv g^{\text{ind } a} \pmod{m}$ и $b \equiv g^{\text{ind } b} \pmod{m}$, получаем требуемое.

Свойство 2.34. $\text{ind } a^n \equiv n \text{ind } a \pmod{c}$.

Если воспользоваться таблицами индексов, то можно решать показательные и степенные сравнения путем их индексирования (дискретного логарифмирования). В самом деле, степенное сравнение $x^n \equiv a \pmod{m}$ равносильно сравнению $n \text{ind } x \equiv \text{ind } a \pmod{c}$, решение которого при наличии таблиц не составляет труда. Положим $d = (n, c)$.

Свойство 2.35. Сравнение $x^n \equiv a \pmod{m}$ разрешимо тогда и только тогда, когда d делит $\text{ind } a$. В случае разрешимости имеется d решений.

2.15. СИМВОЛ ЛЕЖАНДРА

В п. 2.9 мы изучали сравнение $ax \equiv b \pmod{m}$. Рассмотрим теперь сравнение $ax^2 + bx + c \equiv 0 \pmod{m}$. Оно легко сводится к сравнению $(2ax + b)^2 \equiv b^2 - 4ac \pmod{4am}$. Полагая $y = 2ax + b$, $d = b^2 - 4ac$, имеем сравнение $y^2 \equiv d \pmod{4am}$. Фактически, исходное сравнение сведено к сравнению вида $x^2 \equiv a \pmod{m}$. Рассмотрим случай, когда m – простое.

Пусть p – нечетное простое число, $(a, p) = 1$. Символ Лежандра $\left(\frac{a}{p}\right)$ определяется равенством $\left(\frac{a}{p}\right) = 1$, если сравнение $x^2 \equiv a \pmod{p}$ разрешимо, и $\left(\frac{a}{p}\right) = -1$ – в противном случае. Говорят также, что в первом случае a является *квадратичным вычетом* по модулю p и *квадратичным невычетом* – во втором. Таким образом, $\left(\frac{a}{p}\right) = \pm 1$.

Пример 2.3. Квадратичные вычеты по $\mod 7$ – это 1, 2, 4; невычеты – 3, 5, 6.

Если g – первообразный корень по $\mod p$, то каждое целое g^{2k} – квадратичный вычет, а каждое g^{2k+1} – квадратичный невычет. В самом деле, сравнение $(g^t)^2 \equiv g^{2k+1} \pmod{p}$ влечет $g^{|2k+1-2t|} \equiv 1 \pmod{p}$, а значит, $2k+1-2t$ должно делиться на $p-1$. Противоречие, так как первое число нечетно, а второе четно. Следовательно, существует $(p-1)/2$ квадратичных вычетов и столько же невычетов по $\mod p$. Представление вычетов в форме g^{2k} , а невычетов в форме g^{2k+1} влечет за собой важное свойство.

Свойство 2.36. $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right) \left(\frac{b}{p}\right)$, где $(a, p) = (b, p) = 1$.

Теорема 2.13 (критерий Эйлера). Если $(a, p) = 1$, то

$$\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} \pmod{p}.$$

Доказательство.

$$a^{p-1} - 1 = \left(a^{\frac{p-1}{2}} - 1\right) \left(a^{\frac{p-1}{2}} + 1\right) \equiv 0 \pmod{p}.$$

Множители $a^{\frac{p-1}{2}} - 1$ и $a^{\frac{p-1}{2}} + 1$ отличаются на 2. Поэтому только один из них делится на p . Для $a = g^{2k}$ это будет первый множитель. \square

Свойство 2.37. $\left(\frac{-1}{p}\right) = \begin{cases} 1, & p \equiv 1 \pmod{4}, \\ -1, & p \equiv 3 \pmod{4}. \end{cases}$

2.16. КВАДРАТИЧНЫЙ ЗАКОН ВЗАИМНОСТИ

Гаусс доказал следующую лемму, упрощающую вычисление символа Лежандра.

Лемма 2.4. $\left(\frac{a}{p}\right) = (-1)^{\sum_{i=1}^{p_1} \left[\frac{2ai}{p}\right]}$, где $p_1 = \frac{p-1}{2}$.

Доказательство. Рассмотрим числа $a, 2a, \dots, p_1a$ и их абсолютно наименьшие вычеты $\varepsilon_1 r_1, \varepsilon_2 r_2, \dots, \varepsilon_{p_1} r_{p_1}$, где $\varepsilon_j = \pm 1$, $|\varepsilon_j r_j| = r_j$. Числа $\pm a, \pm 2a, \dots, \pm p_1a$ образуют приведенную систему вычетов по $\mod p$, среди которых содержатся r_1, r_2, \dots, r_{p_1} . Последние отличаются от чисел $1, 2, \dots, p_1$ лишь порядком. Перемножая почленно сравнения $ia \equiv \varepsilon_i r_i \pmod{p}$, $i = 1, 2, \dots, p_1$, и сокращая произведение на $1 \cdot 2 \dots p_1 = r_1 \cdot r_2 \dots r_{p_1}$, получаем $a^{\frac{p-1}{2}} = \varepsilon_1, \varepsilon_2, \dots, \varepsilon_{p_1} \pmod{p}$. В силу

$$\left[\frac{2ia}{p}\right] = \left[2 \left[\frac{ia}{p}\right] + 2 \frac{ia}{p}\right] = 2 \left[\frac{ia}{p}\right] + \left[2 \frac{ia}{p}\right]$$

видно, что $\varepsilon_i = 1$ при четном $\left[\frac{2ia}{p} \right]$ и $\varepsilon_i = -1$ в противном случае. Отсюда легко следует утверждение леммы. \square

При нечетном a лемму можно сформулировать в другом виде:

$$\left(\frac{2a}{p} \right) = \left(\frac{4 \frac{a+p}{2}}{p} \right) = \left(\frac{a+p}{p} \right) = (-1)^{\sum_{i=1}^{p_1} \left[\frac{(a+p)i}{p} \right]} = (-1)^{\sum_{i=1}^{p_1} \left[\frac{ai}{p} \right] + \sum_{i=1}^{p_1} i}.$$

$$\text{Поэтому } \left(\frac{2}{p} \right) \left(\frac{a}{p} \right) = (-1)^{\sum_{i=1}^{p_1} \left[\frac{ai}{p} \right] + \frac{p^2 - 1}{q}}.$$

$$\text{При } a = 1 \text{ имеем } \left(\frac{2}{p} \right) = (-1)^{\frac{p^2 - 1}{q}}, \text{ а при нечетном } a \left(\frac{a}{p} \right) = (-1)^{\sum_{i=1}^{p_1} \left[\frac{ai}{p} \right]}.$$

Эта формула необходима при доказательстве следующей весьма важной теоремы.

Теорема 2.14 (квадратичный закон взаимности). Для любых нечетных простых p, q выполнено равенство

$$\left(\frac{q}{p} \right) = (-1)^{\frac{p-1}{2} \cdot \frac{q-1}{2}} \left(\frac{p}{q} \right).$$

Доказательство. Для вычисления $\left(\frac{q}{p} \right)$ используем лемму Гаусса. Пусть $q_1 = \frac{q-1}{2}$. Рассмотрим теперь p_1, q_1 пар чисел pj и qj , где $i = 1, 2, \dots, p_1$; $j = 1, 2, \dots, q_1$. Все они попарно различны. Образуем из них S_1 пар с условием $qi < pj$ и S_2 пар с условием $qi > pj$. Первое условие означает также, что $i \leqslant pj/q$. Поэтому

$$S_1 = \sum_{j=1}^{q_1} \left[\frac{p}{q} j \right].$$

Аналогично:

$$S_2 = \sum_{i=1}^{p_1} \left[\frac{q}{p} i \right].$$

В силу леммы Гаусса

$$\left(\frac{p}{q} \right) = (-1)^{S_1}, \quad \left(\frac{q}{p} \right) = (-1)^{S_2}.$$

Перемножая эти равенства, получаем

$$\left(\frac{p}{q} \right) \left(\frac{q}{p} \right) = (-1)^{p_1 q_1}.$$

\square

2.17. СИМВОЛ ЯКОБИ

Символ Якоби является обобщением символа Лежандра и служит для упрощения вычисления последнего. Пусть P – нечетное натуральное число, $P = p_1 p_2 \cdots p_s$ – его разложение на простые множители. Для всякого целого a , $(a, P) = 1$, символ Якоби определяется по формуле

$$\left(\frac{a}{P} \right) = \left(\frac{a}{p_1} \right) \left(\frac{a}{p_2} \right) \cdots \left(\frac{a}{p_s} \right).$$

Отметим следующие свойства.

Свойство 2.38. $a \equiv b \pmod{P} \Rightarrow \left(\frac{a}{P} \right) = \left(\frac{b}{P} \right)$.

Доказательство. Поскольку $a \equiv b \pmod{P} \Rightarrow a \equiv b \pmod{p_i}$, то $\left(\frac{a}{p_i} \right) = \left(\frac{b}{p_i} \right)$, $i = 1, 2, \dots, s$. \square

Свойство 2.39. $\left(\frac{a_1 a_2 \cdots a_k}{P} \right) = \left(\frac{a_1}{P} \right) \left(\frac{a_2}{P} \right) \cdots \left(\frac{a_k}{P} \right)$.

Доказательство. По определению левая часть равна

$$\left(\frac{a_1 a_2 \cdots a_k}{p_1} \right) \left(\frac{a_1 a_2 \cdots a_k}{p_2} \right) \cdots \left(\frac{a_1 a_2 \cdots a_k}{p_s} \right).$$

Первый множитель по следствию 2.36 равен $\left(\frac{a_1}{p_1} \right) \left(\frac{a_2}{p_1} \right) \cdots \left(\frac{a_k}{p_1} \right)$. Применяя это свойство ко всем множителям и группируя их, получаем правую часть. \square

Свойство 2.40. $\left(\frac{-1}{P} \right) = (-1)^{\frac{p-1}{2}}$.

Доказательство. Заметим, что

$$\left(\frac{-1}{P} \right) = \left(\frac{-1}{p_1} \right) \left(\frac{-1}{p_2} \right) \cdots \left(\frac{-1}{p_s} \right) = (-1)^{\frac{p_1-1}{2} + \frac{p_2-1}{2} + \cdots + \frac{p_s-1}{2}}.$$

С другой стороны, можно многократно воспользоваться легко проверяемым утверждением. Пусть a и b нечетные. Тогда

$$\frac{1}{2}(ab - 1) \equiv \frac{1}{2}(a - 1) + \frac{1}{2}(b - 1) \pmod{2}.$$

\square

Свойство 2.41. $\left(\frac{1}{P} \right) = 1$.

Оно вытекает из предыдущих свойств.

Заметим, что из равенства $\left(\frac{a}{P}\right) = 1$ не следует, что a является квадратичным вычетом по модулю P . В действительности a является квадратичным вычетом по модулю P тогда и только тогда, когда a – квадратичный вычет по модулю каждого простого p_i , $i = 1, 2, \dots, s$. В то же время из равенства $\left(\frac{a}{P}\right) = -1$ следует, что a – квадратичный невычет по $\mod P$.

Теорема 2.15. *Пусть P и Q – положительные взаимно простые нечетные модули. Тогда*

$$\left(\frac{Q}{P}\right) = (-1)^{\frac{P-1}{2} \frac{Q-1}{2}} \left(\frac{P}{Q}\right).$$

Доказательство. Во-первых,

$$\left(\frac{Q}{P}\right) = \prod_{i=1}^r \prod_{j=1}^s \left(\frac{q_i}{p_j}\right) = (-1)^{\sum_i \sum_j \frac{p_j-1}{2} \frac{q_i-1}{2}} \prod_i \prod_j \left(\frac{p_j}{q_i}\right).$$

Во-вторых, правую часть легко преобразовать к виду

$$(-1)^{\left(\sum_{j=1}^s \frac{p_j-1}{2}\right) \left(\sum_{i=1}^r \frac{q_i-1}{2}\right)} \left(\frac{P}{Q}\right).$$

Осталось заметить, что

$$\sum_{j=1}^s \frac{p_j-1}{2} \equiv \frac{P-1}{2} \pmod{2}, \quad \sum_{i=1}^r \frac{q_i-1}{2} \equiv \frac{Q-1}{2} \pmod{2}.$$

□

2.18. ЦЕПНЫЕ ДРОБИ

Пусть α – положительное вещественное число. Положим $q_1 = [\alpha]$. Тогда если α нецелое, то

$$\alpha = q_1 + \frac{1}{\alpha_2},$$

где $\alpha_2 > 1$. Продолжая этот процесс, получаем

$$\begin{aligned} \alpha_2 &= q_2 + \frac{1}{\alpha_3}, \\ \alpha_3 &= q_3 + \frac{1}{\alpha_4}, \\ &\dots \\ \alpha_s &= q_s + \frac{1}{\alpha_{s+1}}. \end{aligned}$$

Следовательно,

$$\alpha = q_1 + \cfrac{1}{q_2 + \cfrac{1}{q_3 + \cdots \cfrac{1}{q_s + \cfrac{1}{\alpha_{s+1}}}}}.$$

Представление α в указанном виде называется разложением α в цепную (непрерывную) дробь. При иррациональном α цепная дробь, очевидно, оказывается бесконечной. При рациональном α дробь будет конечной. Покажем это.

Пусть $\alpha = \frac{a}{b}$. Применим алгоритм Евклида:

$$\begin{aligned} a &= bq_1 + r_2, & \frac{a}{b} &= q_1 + \frac{1}{\overline{r_2}}, \\ b &= r_2 q_2 + r_3, & \frac{b}{r_2} &= q_2 + \frac{1}{\overline{r_3}}, \\ &\dots & &\dots \\ r_{n-2} &= r_{n-1} q_{n-1} + r_n, & \frac{r_{n-2}}{r_{n-1}} &= q_{n-1} + \frac{1}{\overline{r_r}}, \\ r_{n-1} &= r_n q_n, & \frac{r_n}{r_n} &= q_n. \end{aligned}$$

Следовательно,

$$\frac{a}{b} = q_1 + \cfrac{1}{q_2 + \cfrac{1}{q_3 + \cdots \cfrac{1}{q_n}}}.$$

Теорема 2.16. *Всякое иррациональное α разлагается в бесконечную цепную дробь. Всякое рациональное α разлагается в конечную цепную дробь.*

2.19. ПОДХОДЯЩИЕ ДРОБИ

Далее будем использовать запись

$$\alpha = [q_1, q_2, \dots, q_s, \alpha_{s+1}].$$

Числа q_1, q_2, \dots, q_s называются неполными частными, а рациональное число $\delta_s = [q_1, q_2, \dots, q_s]$ – s -й подходящей дробью числа α . Через P_s обозначается числитель, а через Q_s – знаменатель подходящей дроби δ_s .

Например, $P_1 = q_1$, $Q_1 = 1$. Удобно считать, что $P_0 = 1$, $Q_0 = 0$. Имеет место следующий закон образования подходящих дробей.

Теорема 2.17. $P_s = q_s P_{s-1} + P_{s-2}$, $Q_s = q_s Q_{s-1} + Q_{s-2}$.

Доказательство. Заметим, что δ_s получается из δ_{s-1} заменой в формуле $\delta_{s-1} = [q_1, q_2, \dots, q_{s-1}] q_{s-1}$ на $q_{s-1} + \frac{1}{q_s}$. Поэтому

$$\delta_1 = \frac{P_1}{Q_1} = \frac{q_1}{1},$$

$$\delta_2 = \frac{P_2}{Q_2} = q_1 + \frac{1}{q_2} = \frac{q_2 q_1 + 1}{q_2 1 + 0} = \frac{q_2 P_1 + P_0}{q_2 Q_1 + 0}.$$

Применим индукцию:

$$\delta_s = \frac{P_s}{Q_s} = \frac{\left(q_{s-1} + \frac{1}{q_s}\right) P_{s-2} + P_{s-3}}{\left(q_{s-1} + \frac{1}{q_s}\right) Q_{s-2} + Q_{s-3}} = \frac{\frac{1}{q_s} P_{s-2} + P_{s-1}}{\frac{1}{q_s} Q_{s-2} + Q_{s-1}} = \frac{q_s P_{s-1} + P_{s-2}}{q_s Q_{s-1} + Q_{s-2}}.$$

□

Теорема 2.18. $\delta_s - \delta_s = \frac{(-1)^s}{Q_s Q_{s-1}}$, $P_s Q_{s-1} - Q_s P_{s-1} = (-1)^s$.

Доказательство.

$$\delta_s - \delta_{s-1} = \frac{P_s}{Q_s} - \frac{P_{s-1}}{Q_{s-1}} = \frac{P_s Q_{s-1} - Q_s P_{s-1}}{Q_s Q_{s-1}},$$

$$\begin{aligned} P_s Q_{s-1} - Q_s P_{s-1} &= (q_s P_{s-1} + P_{s-2}) Q_{s-1} - (q_s Q_{s-1} + Q_{s-2}) Q_{s-1} = \\ &= -(\delta_{s-1} - \delta_{s-2}) = (-1)^{s-1} (P_1 Q_0 - Q_1 P_0) = (-1)^s. \end{aligned}$$

□

Следствие 2.1. Все подходящие дроби несократимы.

Теорема 2.19. При $\delta_s \neq \alpha$ подходящие дроби $\delta_1, \delta_2, \dots, \delta_s$ с четными номерами являются последовательными приближениями α с одной стороны, а с нечетными номерами – с другой, т. е.

$$\alpha_1 < \alpha_3 < \dots < \alpha < \dots < \alpha_4 < \alpha_2.$$

Доказательство. Рассмотрим последовательность

$$\alpha = q_1 + \frac{1}{\alpha_2}, \alpha_2 = q_2 + \frac{1}{\alpha_3}, \dots, \alpha_{s-2} = q_{s-2} + \frac{1}{\alpha_{s-1}}, \alpha_{s-1} = q_{s-1} + \frac{1}{\alpha_s}.$$

Если $\frac{1}{\alpha_s}$ заменить на ноль, то последнее число α_{s-1} уменьшится, α_{s-2} увеличится и т. д. Если α_s заменить на q_s , то характер увеличений и уменьшений будет обратным. Это означает, что все подходящие дроби с четными номерами больше α , а с нечетными – меньше.

Докажем убывание четных номеров:

$$\frac{P_{2m}}{Q_{2m}} - \frac{P_{2m-2}}{Q_{2m-2}} = \frac{P_{2m}Q_{2m-2} - P_{2m-2}Q_{2m}}{Q_{2m}Q_{2m-2}}.$$

В первой части сделаем замену:

$$P_{2m} = q_{2m}P_{2m-1} + P_{2m-2}, \quad Q_{2m} = q_{2m}Q_{2m-1} + Q_{2m-2}.$$

Тогда

$$\begin{aligned} \delta_{2m} - \delta_{2m-2} &\equiv \frac{q_{2m}P_{2m-1} + P_{2m-2}Q_{2m-2} - P_{2m-2}(q_{2m}Q_{2m-1} + Q_{2m-2})}{Q_{2m}Q_{2m-2}} = \\ &= \frac{q_{2m}(P_{2m-1}Q_{2m-2} - P_{2m-2}Q_{2m-1})}{Q_{2m}Q_{2m-2}} = q_{2m} \frac{(-1)^{2m-1}}{Q_{2m}Q_{2m-2}} < 0. \end{aligned}$$

Аналогично доказывается возрастание нечетных номеров. \square

2.20. ПОДХОДЯЩИЕ ДРОБИ В КАЧЕСТВЕ НАИЛУЧШИХ ПРИБЛИЖЕНИЙ

Несократимую дробь $\frac{a}{b}$ ($b > 0$) назовем *наилучшим приближением первого рода* числа $\alpha \in R$, если

$$\frac{a}{b} \neq \frac{c}{d}, \quad 0 < d \leq b \implies \left| \alpha - \frac{c}{d} \right| > \left| \alpha - \frac{a}{b} \right|.$$

В случае, когда выполняется условие

$$\frac{a}{b} \neq \frac{c}{d}, \quad 0 < d \leq b \implies |\alpha d - c| > |\alpha b - a|,$$

говорят о *наилучшем приближении второго рода*, которое является и наилучшим приближением первого рода. В самом деле, перемножая неравенства

$$\left| \alpha - \frac{c}{d} \right| \leq \left| \alpha - \frac{a}{b} \right|, \quad d \leq b,$$

получаем $|\alpha d - c| \leq |\alpha b - a|$. Обратное утверждение неверно. Легко проверить, что $\frac{1}{3}$ является наилучшим приближением лишь первого рода числа $\frac{1}{5}$.

Теорема 2.20. Всякая подходящая дробь δ_s , $s > 1$ есть наилучшее приближение второго рода.

Доказательство. При $\alpha = \delta_s$ утверждение очевидно. Пусть $\alpha \neq \delta_s$. Требуется доказать, что

$$0 < Q \leq Q_s, \quad \frac{P}{Q} \neq \frac{P_s}{Q_s} \implies |\alpha Q - P| > |\alpha Q_s - P_s|.$$

Представим P и Q в виде

$$P = uP_s + vP_{s-1}, \quad Q = uQ_s + vQ_{s-1},$$

где u, v – целые. Это возможно, так как по теореме 2.18

$$\begin{vmatrix} P_s & P_{s-1} \\ Q_s & Q_{s-1} \end{vmatrix} = P_s Q_{s-1} - Q_s P_{s-1} = (-1)^s,$$

т. е. применимо правило Крамера. Поэтому

$$|\alpha Q - P| = |u(\alpha Q_s - P_s) + v(\alpha Q_{s-1} - P_{s-1})| > |\alpha - Q_s P_s|.$$

Последнее неравенство имеет место, так как $\alpha Q_s - P_s \neq 0$, поскольку $\delta_s \neq \alpha$. По той же причине $\alpha Q_{s-1} - P_{s-1} \neq 0$. Более того, эти величины – разных знаков. Числа u, v также ненулевые. Например, $v = 0$ влечет $\frac{P}{Q} = \frac{P_s}{Q_s}$.

Более того, числа u, v – разных знаков, так как $Q \leq Q_s$. \square

Теорема 2.21. $|\alpha Q_{s-1} - P_{s-1}| > |\alpha Q_s - P_s|$, а значит, $u \left| \alpha - \frac{P_{s-1}}{Q_{s-1}} \right| > \left| \alpha - \frac{P_s}{Q_s} \right|$.

Доказательство. Представим число α в виде $\alpha = [q_1, q_2, \dots, q_s, \alpha_{s+1}]$. Либо α_{s+1} нет, либо $\alpha_{s+1} > 1$. В первом случае $|\alpha Q_s - P_s| = 0$, а $|\alpha Q_{s-1} - P_{s-1}| > 0$. Во втором случае

$$\alpha = \frac{P_{s+1}}{Q_{s+1}} = \frac{\alpha_{s+1} P_s + P_{s-1}}{\alpha_{s+1} Q_s + Q_{s-1}} \implies \alpha \alpha_{s+1} Q_s + \alpha Q_{s-1} = \alpha_{s+1} P_s + P_{s-1}.$$

Поэтому $\alpha_{s+1} (\alpha Q_s - P_s) = -\alpha Q_{s-1} + P_{s-1}$. Поскольку $\alpha_{s+1} > 0$, то $|\alpha Q_s - P_s| < |\alpha Q_{s-1} - P_{s-1}|$. \square

Теорема 2.22. Если $\frac{P}{Q}$ – несократимая дробь, $Q > 0$, такая, что

$$\left| \alpha - \frac{P}{Q} \right| < \frac{1}{2Q^2},$$

то $\frac{P}{Q}$ – подходящая дробь числа α .

Доказательство. Выберем s такое, что $Q_s \leq Q < Q_{s+1}$. Как и в доказательстве теоремы 2.20, находим, что $|\alpha Q - P| \geq |\alpha Q_s - P_s|$. Тогда

$$\begin{aligned} \left| \frac{P}{Q} - \frac{P_s}{Q_s} \right| &\leq \left| \alpha - \frac{P_s}{Q_s} \right| = \frac{1}{Q} |\alpha Q - P| + \frac{1}{Q_s} |\alpha Q_s - P_s| \leq \\ &\leq \left(\frac{1}{Q} + \frac{1}{Q_s} \right) |\alpha Q - P| < \left(\frac{1}{Q} + \frac{1}{Q_s} \right) \frac{1}{2Q} \leq \frac{1}{Q_s Q}. \end{aligned}$$

Значит, число в левой части равно нулю. \square

В заключение отметим факт, который потребуется в дальнейшем: $Q_k \geq 2^{\frac{k-1}{2}}$.

В самом деле, $Q_k = q_k Q_{k-1} + Q_{k-2} \geq Q_{k-1} + Q_{k-2} \geq 2Q_{k-2}$.

Это означает, в частности, что при разложении рационального числа $\frac{P}{Q}$ в цепную дробь длина последней ограничена сверху величиной $\log_2 Q + 1$.

2.21. ЗАДАНИЯ

1. Показать, что $30 \mid m^5 - m$, $6 \mid m(m^2 + 5)$, $42 \mid m^7 - m$, $30 \mid mn(m^4 - n^4)$ при любых натуральных m , n .

2. Показать, что при натуральном m произведение

$$(m+1)(m+2) \cdots (m+m)$$

делится на 2^m .

3. Будут ли целыми числа $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$, $1 + \frac{1}{3} + \frac{1}{5} + \dots + \frac{1}{2n+1}$, $n > 1$?

4. Пусть $(a, b) = 1$. Показать, что $(a+b, a-b) \leq 2$.

5. Вычислить НОД: (549, 387), (589, 343), (12606, 6494), (6188, 4709) и найти их линейные разложения.

6. Доказать, что НОД можно определить как такой общий делитель, который делится на любой другой общий делитель.

7. Пусть $(a, b) = 1$, $ab = c^2$. Показать, что числа a и b будут квадратами.

8. Показать, что $p^2 - q^2$ кратно 24, где p, q – простые числа и $p, q > 3$.

9. Доказать бесконечность множества простых чисел вида $4m + 3$.

10. Доказать бесконечность множества простых чисел вида $6m + 5$.

11. Пусть k – натуральное. Доказать, что в натуральном ряду имеется бесконечно много отрезков $m, m+1, \dots, m+k$, не содержащих простых чисел.

12. Найти канонические разложения чисел 82798848, 81057226635.

13. Разложить на простые множители числа $10!$, $15!$, $20!$, $30!$.
14. Определить, сколькими нулями оканчиваются числа $50!$, $100!$.
15. Найти функцию Эйлера для чисел 375 , 720 , 957 , 988 , 1200 , 4320 .
16. Сколько чисел в интервале от 1 до 120 , не взаимно простых с 30 ?
17. Дано $\varphi(a) = 120$, $a = p q$, где p , q – простые. Найти a , если $p - q = 2$.
18. Доказать, что уравнение $15x^2 - 7y^2 = 9$ не имеет решений в целых числах.
19. Решить в целых числах уравнение $x^2 + y^2 = z^2$.
20. Доказать, что сумма квадратов пяти последовательных целых чисел не может быть точным квадратом.
21. Решить в целых числах уравнения $53x + 47y = 1$, $22x + 32y = 18$.
22. Путем перебора решить сравнения:
 - 1) $5x^2 - 15x + 22 \equiv 0 \pmod{3}$; 4) $7x \equiv 31 \pmod{6}$;
 - 2) $x^3 - 12 \equiv 0 \pmod{5}$; 5) $12x \equiv 1 \pmod{7}$;
 - 3) $3x \equiv 1 \pmod{5}$; 6) $6x + 5 \equiv 6 \pmod{7}$.
23. Способом Эйлера решить сравнения:
 - 1) $5x \equiv 0 \pmod{7}$; 3) $5x \equiv 7 \pmod{7}$;
 - 2) $25x \equiv 15 \pmod{7}$; 4) $5x \equiv 26 \pmod{12}$.
24. Решить системы сравнений:

$$\begin{cases} x \equiv 3 \pmod{8}, \\ x \equiv 11 \pmod{20}, \\ x \equiv 1 \pmod{15}, \end{cases} \quad \begin{cases} x \equiv 2 \pmod{17}, \\ 5x \equiv 3 \pmod{9}, \\ 8x \equiv 4 \pmod{14}. \end{cases}$$

25. Решить системы сравнений китайским способом:

$$\begin{cases} x \equiv 2 \pmod{7}, \\ x \equiv 9 \pmod{11}, \\ x \equiv 3 \pmod{13}, \end{cases} \quad \begin{cases} x \equiv 3 \pmod{8}, \\ x \equiv 2 \pmod{9}, \\ x \equiv 5 \pmod{7}. \end{cases}$$

26. Составить таблицы индексов:

- 1) по $\text{mod } 29$ с основанием 2 ;
- 2) по $\text{mod } 23$ с основанием 5 .

Путем индексирования решить сравнения:

- 1) $2^x \equiv \pmod{67}$; 5) $37x^{16} \equiv 62 \pmod{73}$;
- 2) $52^x \equiv 38 \pmod{29}$; 6) $2x^3 \equiv 17 \pmod{41}$;
- 3) $13^x \equiv \pmod{47}$; 7) $5x^4 \equiv 3 \pmod{11}$;
- 4) $12^x \equiv 17 \pmod{31}$; 8) $27x^5 \equiv 2 \pmod{31}$.

27. Пусть a принадлежит показателю δ , b – показателю γ , $(\delta, \gamma) = 1$. Показать, что ab принадлежит показателю $\delta\gamma$.

28. Пусть a принадлежит показателю δ , b – показателю γ . Как построить элемент, принадлежащий показателю $[\delta, \gamma]$?

29. Пусть a принадлежит показателю δ . Какому показателю принадлежит a^γ ?

30. Пусть g – первообразный корень по модулю m . Сколько всего первообразных корней по этому модулю?

31. Вычислить символы Лежандра и Якоби:

$$\left(\frac{47}{125} \right), \left(\frac{131}{283} \right), \left(\frac{123}{917} \right).$$

32. Найти все квадратичные вычеты по модулю числа p :

$$p = 11, p = 13, p = 17.$$

33. Доказать, что при $p = 4k + 1$ числа a и $p - a$ – одновременно квадратичные вычеты или невычеты, а при $p = 4k + 3$ – наоборот.

34. Решить сравнения:

$$x^2 \equiv 19 \pmod{31};$$

$$x^2 \equiv 15 \pmod{53};$$

$$x^2 \equiv 11 \pmod{59}.$$

35. Решить сравнения:

$$x^2 + 8x - 20 \equiv 0 \pmod{45};$$

$$5x^2 + x + 4 \equiv 0 \pmod{10}.$$

36. Пользуясь леммой Гаусса и представлением $\left(\frac{2}{p} \right) = \left(\frac{\frac{p+1}{2}}{p} \right) = \left(\frac{1+p}{p} \right)$, доказать еще раз, что $\left(\frac{2}{p} \right) = (-1)^{\frac{p^2-1}{8}}$.

37. Показать, что для любого целого d и любого нечетного простого p количество решений сравнения $x^2 \equiv d \pmod{p}$ равно $1 + \left(\frac{d}{p} \right)$.

38. Найти способ решения сравнений вида $x^2 \equiv 4 \pmod{m}$.

39. Доказать, что количество решений уравнения $x^2 + y^2 = p$, $(x, y) = 1$, $x > 0$, $y > 0$, равно количеству решений сравнения $z^2 + 1 \equiv 0 \pmod{p}$.

40. Разложить в цепные дроби числа $\frac{170}{109}, \frac{99}{170}, \frac{125}{92}, \sqrt{2}$.

Г л а в а 3

АЛГЕБРАИЧЕСКИЕ ОСНОВЫ

3.1. ПОНЯТИЕ ГРУППЫ

Группой называется непустое множество G с алгебраической операцией $*$ на нем, для которой выполняются три следующие аксиомы.

1. Операция $*$ ассоциативна, т. е. для любых $a, b, c \in G$

$$a * (b * c) = (a * b) * c.$$

2. В G имеется *единичный элемент* (или *единица*) e такой, что для любого $a \in G$

$$a * e = e * a = a.$$

3. Для каждого $a \in G$ существует *обратный элемент* $a^{-1} \in G$ такой, что

$$a * a^{-1} = a^{-1} * a = e.$$

Если дополнительно группа удовлетворяет четвертой аксиоме:

4. Для любых $a, b \in G$

$$a * b = b * a,$$

то группа называется *абелевой* (или *коммутативной*).

Для групповой операции будем использовать мультипликативное обозначение и вместо $a * b$ писать ab , называя этот элемент *произведением* элементов a и b . Иногда для групповой операции используют аддитивную запись: $a + b$. В этом случае вместо единицы пишут ноль, а вместо a^{-1} — $-a$. Такие обозначения обычно резервируют для абелевых групп.

В группе имеется лишь один единичный элемент. Действительно, если e' — еще одна единица, то $e' = e'e = e$. Для любого элемента имеется лишь один обратный. Пусть x и y — обратные элементы для $a \in G$. Тогда, по ассоциативности,

$$x = xe = x(ay) = (xa)y = ey = y.$$

Примеры. Множество \mathbb{Z} целых чисел образует группу относительно операции сложения. То же можно сказать относительно рациональных чисел \mathbb{Q} , вещественных чисел \mathbb{R} и комплексных чисел \mathbb{C} .

Обозначим через $\bar{0}, \bar{1}, \bar{2}, \bar{3}, \bar{4}$ классы вычетов по модулю 5. Определим их сложение по модулю 5. Например, $\bar{1} + \bar{2} = \bar{3}, \bar{2} + \bar{3} = \bar{0}, \bar{2} + \bar{4} = \bar{1}$, т. е. осуществляется обычное сложение и при необходимости берется остаток от деления на 5. Эта группа обозначается через \mathbb{Z}_5 и называется (*аддитивной*) группой классов вычетов по модулю 5. Аналогично строится группа классов вычетов \mathbb{Z}_m по любому модулю m . Если взять все классы вычетов, взаимно простые с модулем m , и определить их умножение по модулю m , то получается группа, обозначаемая через \mathbb{Z}_m^* . Отметим, что существование обратного элемента для $\bar{a} \in \mathbb{Z}_m^*$ вытекает из разрешимости сравнения $ax \equiv 1 \pmod{m}$ при $(a, m) = 1$.

Число элементов конечной группы G называется *порядком группы* и обозначается через $|G|$. Например, $|\mathbb{Z}_m| = m, |\mathbb{Z}_m^*| = \varphi(m)$.

Мультиликативная группа G называется *циклической*, если она порождена одним элементом, т. е. в ней имеется такой элемент a (образующий), что любой другой элемент b представим в виде $b = a^n, n \in \mathbb{Z}$. Если n – отрицательное, то под a^n понимается произведение $(a^{-1})^{|n|}$. Циклическими являются группы \mathbb{Z} и \mathbb{Z}_m . Группа \mathbb{Z}_m^* циклическая лишь в случае, когда по модулю m существует первообразный корень. В циклической группе, конечной или нет, может быть несколько образующих элементов. В аддитивной группе \mathbb{Z} образующими будут элементы 1 и -1 . Циклическая группа всегда коммутативна.

Существует удобный способ задания конечной группы: в виде таблицы. Обычно она называется *таблицей Кэли*. Ее строки и столбцы помечаются элементами группы, и на пересечении строки, помеченной элементом a , и столбца, помеченного элементом b , ставится элемент ab . Например, таблица Кэли группы \mathbb{Z}_5 имеет вид

$+$	$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{4}$
$\bar{0}$	$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{4}$
$\bar{1}$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{4}$	$\bar{0}$
$\bar{2}$	$\bar{2}$	$\bar{3}$	$\bar{4}$	$\bar{0}$	$\bar{1}$
$\bar{3}$	$\bar{3}$	$\bar{4}$	$\bar{0}$	$\bar{1}$	$\bar{2}$
$\bar{4}$	$\bar{4}$	$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$

3.2. ПОДГРУППЫ ГРУПП

Подмножество H группы G называется *подгруппой* этой группы, если H образует группу относительно операции группы G .

Подгруппы группы G , отличные от тривиальных подгрупп $\{e\}$ и G , называются *собственными* подгруппами.

Теорема 3.1. Подмножество H группы G будет ее подгруппой тогда и только тогда, когда выполняются условия:

- 1) $a, b \in H \Rightarrow ab \in H$;
- 2) $a \in H \Rightarrow a^{-1} \in H$.

Доказательство. Необходимость следует из определения группы. Докажем достаточность. Аксиома 1 в определении группы (ассоциативность) автоматически выполняется и на любом подмножестве. Осталось показать, что $e \in H$. Возьмем произвольный элемент $a \in H$. Тогда $a^{-1} \in H$, а значит, $e = aa^{-1} \in H$. \square

Условия 1 и 2 в теореме 3.1 эквивалентны условию

$$a, b \in H \Rightarrow ab^{-1} \in H,$$

а если группа G конечна, то условие 1 влечет за собой условие 2.

Теорема 3.2. *Если H – подгруппа группы G , то отношение R_H на G , определяемое условием*

$$(a, b) \in R_H \Leftrightarrow a = bh,$$

для некоторого $h \in H$ является бинарным отношением эквивалентности.

Классы эквивалентности по отношению R_H называются *левыми смежными классами* группы G по подгруппе H и обозначаются

$$aH = \{ah \mid h \in H\}.$$

Смежные классы группы по подгруппе либо совпадают, либо не пересекаются. Аналогично определяются *правые смежные классы* по подгруппе H , которые имеют вид $Ha = \{ha \mid h \in H\}$. Для абелевой группы эти два понятия идентичны.

Пусть, например, $G = \mathbb{Z}$, $H = 2\mathbb{Z}$ – подгруппа четных чисел. Тогда имеем два смежных класса: $2\mathbb{Z}$ – четные числа, $1 + 2\mathbb{Z}$ – нечетные.

Теорема 3.3. *Если H – конечная группа, то каждый (левый или правый) смежный класс по ней содержит $|H|$ элементов.*

Доказательство. $a_1h = a_2h \Rightarrow a_1 = a_2$, так как обе части первого равенства можно умножить на h^{-1} . \square

Если количество смежных классов группы G по подгруппе H конечно, то оно называется *индексом* подгруппы H в группе G и обозначается через $[G : H]$. В этом случае имеет место так называемое *разложение группы по подгруппе*:

$$G = H \cup g_1H \cup g_2H \cup \dots \cup g_\rho H,$$

где все смежные классы попарно не пересекаются, $\rho = [G : H] - 1$. Аналогично можно ввести разложение на правые смежные классы по подгруппе.

Теорема 3.4. *Пусть G – конечная группа. Тогда*

$$|G| = [G : H]|H|.$$

Следствие 3.1 (теорема Лагранжа). *Порядок конечной группы делится на порядок любой ее подгруппы.*

Пусть $a \in G$. Положим $a^n = a a \cdots a$, если n – натуральное, $a^n = (a^{-1})^{|n|}$, если n – целое отрицательное, и, наконец, $a^0 = e$. Таким образом, можно рассмотреть подмножество

$$\langle a \rangle = \{a^n \mid n \in \mathbb{Z}\}.$$

Оно, как легко показать, является подгруппой группы G . Эта подгруппа называется *циклической подгруппой*, порожденной элементом a . Ее порядок называется *порядком элемента a* . Иными словами, элемент $a \in G$ называется элементом порядка $m \in \mathbb{N}$, если $a^m = e$, где m – наименьшее натуральное с этим условием. Легко показать, что $m \mid l$, если $a^l = e$. Если такого m нет, то элемент a называется *элементом бесконечного порядка*. Из теоремы Лагранжа вытекает, что порядок конечной группы G делится на порядок любого ее элемента a . Поэтому $a^{|G|} = e$. Поскольку $|\mathbb{Z}_m^*| = \varphi(m)$, то в качестве еще одного следствия получается теорема Эйлера о том, что если $(a, m) = 1$, то $a^{\varphi(m)} \equiv 1 \pmod{m}$.

В дальнейшем нам потребуется утверждение о порядках элементов в абелевой группе G . Пусть $a, b \in G$ и их порядки – m и n соответственно, причем $(m, n) = 1$. Покажем, что порядок произведения ab равен mn . Обозначим порядок ab через δ . Тогда $(ab)^\delta = 1$, а значит, $(ab)^{\delta m} = 1 \Rightarrow a^{\delta m}b^{\delta m} = 1 \Rightarrow b^{\delta m} = 1 \Rightarrow n \mid \delta m \Rightarrow n \mid \delta$. Аналогично показывается, что $m \mid \delta$. Поэтому $mn \mid \delta$. Последнее в силу минимальности δ означает $\delta = mn$. Доказанное является частным случаем следующего утверждения.

Теорема 3.5. *Пусть даны два элемента $a, b \in G$ абелевой группы. Тогда в группе найдется элемент порядка $[m, n]$.*

Доказательство. Случай $(m, n) = 1$ уже рассмотрен. Пусть $(m, n) = d$. Легко построить разбиение $d = d_1d_2$ такое, что $(m/d_1, n/d_2) = 1$. Элементы a^{d_1} и a^{d_2} имеют порядки m/d_1 и n/d_2 соответственно. Их произведение имеет порядок $(m/d_1)(n/d_2) = mn/d = [m, n]$. \square

3.3. ЦИКЛИЧЕСКИЕ ГРУППЫ

Пусть $\langle a \rangle$ – циклическая группа, порожденная элементом a .

Теорема 3.6. *Каждая подгруппа циклической группы также является циклической.*

Доказательство. Пусть H – подгруппа циклической группы $\langle a \rangle$ такая, что $H \neq \{e\}$. Если $a^n \in H$, то $a^{-n} \in H$. Поэтому H содержит степень a с натуральным показателем. Пусть d – наименьший натуральный показатель, для которого $a^d \in H$. Пусть $a^s \in H$. Положим $s = dq + r$, $0 \leq r < d$; $q, r \in \mathbb{Z}$. Тогда $a^r = a^s(a^{-d})^q \in H$, что противоречит минимальности d , если $r \neq 0$. Поэтому $H = \langle a^d \rangle$. \square

Теорема 3.7. В конечной циклической группе $\langle a \rangle$ порядка m элемент a^k порождает подгруппу порядка $m(k, m)^{-1}$.

Доказательство. Пусть $d = (k, m)$. Порядок группы $\langle a^d \rangle$ – наименьшее натуральное n такое, что $a^{dn} = e$. Условие $a^{kn} = e$ справедливо лишь когда $m \mid kn$, т. е. тогда и только тогда, когда $(m/d) \mid n$. Наименьшее натуральное n с таким свойством есть m/d . \square

Теорема 3.8. Если d – положительный делитель m , то G содержит единственную подгруппу индекса d .

Доказательство. Если d задано, то $\langle a^d \rangle$ является подгруппой порядка m/d , и потому она имеет индекс d . Если $\langle a^k \rangle$ – другая подгруппа индекса d , то ее порядок равен m/d . Поэтому в силу теоремы 3.7 $\frac{m}{d} = \frac{m}{(k, m)} \Rightarrow d = (k, m) \Rightarrow d \mid k$. Следовательно, $a^k \in \langle a^d \rangle$. Но поскольку обе группы одного порядка, то они совпадают. \square

Теорема 3.9. Для любого положительного делителя l числа m группа $\langle a \rangle$ содержит в точности одну подгруппу порядка l .

Доказательство. В силу предыдущей теоремы подгруппами порядка l являются только подгруппы, индексы которых равны m/l . \square

Теорема 3.10. Пусть l – положительный делитель порядка конечной циклической группы $\langle a \rangle$. Тогда $\langle a \rangle$ содержит $\varphi(l)$ элементов порядка l .

Доказательство. Пусть $|\langle a \rangle| = m = dl$. В силу теоремы 3.7 элемент a^k имеет порядок l лишь в случае, если $(k, m) = d$. Поэтому количество элементов порядка l равно количеству целых k , $1 \leq k \leq m$, для которых $(k, m) = d$. Поэтому $k = dn$, где $1 \leq n \leq l$. Теперь условие $(k, m) = d$ равносильно условию $(k, l) = 1$. Количество таких чисел равно $\varphi(l)$. \square

Попутно нами доказана следующая теорема.

Теорема 3.11. Конечная циклическая группа $\langle a \rangle$ порядка m содержит $\varphi(m)$ образующих. Элемент a^l является образующим лишь при условии $(l, m) = 1$.

3.4. ГОМОМОРФИЗМЫ ГРУПП

Отображение $f : G \rightarrow G'$ группы G в группу G' называется *гомоморфизмом*, если оно согласовано с операциями на группах G и G' , т. е. $f(ab) = f(a)f(b)$ для любых двух элементов $a, b \in G$. Если это отображение сюръективное, то оно называется *эпиморфизмом*. В этом случае группа G' называется *гомоморфным образом* группы G . Приставка «моно» употребляется в случае, когда гомоморфизм инъективен. Биективный гомоморфизм называется *изоморфизмом*. Для изоморфных групп употребляется обозначение $G \cong H$. Изоморфизм группы G на себя называется *автоморфизмом*.

Примеры. Обозначим через $GL(n, R)$ группу по умножению всех невырожденных матриц n -го порядка с вещественными элементами. Тогда отображение $A \rightarrow \det A$, $A \in GL(n, R)$ будет эпиморфизмом на мультиликативную группу поля вещественных чисел \mathbb{R}^* .

Еще один пример эпиморфизма дает отображение $\varphi : \mathbb{Z} \rightarrow \mathbb{Z}_m$, при котором $\varphi(a) = \bar{a}$, т. е. элемент $a \in \mathbb{Z}$ отображается в соответствующий класс вычетов по модулю m .

Ядром гомоморфизма $f : G \rightarrow H$ называется множество

$$\ker \varphi = \{a \in G \mid f(a) = e'\},$$

где e' – единичный элемент группы H .

В случае гомоморфизма $GL(n, \mathbb{R}) \rightarrow \mathbb{R}^*$ ядром будет подгруппа матриц с единичным определителем. Ядром во втором примере является группа чисел $m\mathbb{Z}$, кратных модулю m .

Сохранив для аддитивной группы поля вещественных чисел обозначение \mathbb{R} и обозначив через \mathbb{R}^+ мультиликативную группу положительных вещественных чисел, имеем изоморфизм $\mathbb{R}^+ \cong R$, заданный функцией $y = \ln x$.

Легко показать, что ядро любого гомоморфизма является подгруппой H группы G с важным дополнительным условием: $g^{-1}Hg = H$ для любого элемента $g \in G$. Такие подгруппы называются *нормальными подгруппами* (*нормальными делителями*). Используется обозначение $H \triangleleft G$. Условие нормальности, как нетрудно видеть, можно переписать в виде $gH = Hg$ или $gHg^{-1} = H$. В абелевой группе все подгруппы являются нормальными.

Если H – нормальная подгруппа группы G , то множество смежных классов группы G по подгруппе H можно наделить групповой структурой. Соответствующая группа называется *факторгруппой* группы G по подгруппе H и обозначается G/H . Определим композицию смежных классов по формуле

$$(g_1H)(g_2H) = g_1g_2H.$$

Докажем корректность. Пусть g_1h_1 и g_2h_2 – другие представители смежных классов. Тогда $g_1h_1g_2h_2$ можно представить в виде $g_1g_2h'_1h_2$, так как в силу $gH = Hg$ произведение h_1g_2 можно представить в виде $g_2h'_1$. Поэтому

$$(g_1h_1g_2h_2)H = (g_1g_2)((h'_1h_2)H) = g_1g_2H.$$

Теорема 3.12 (теорема о гомоморфизме). Пусть $f : G \rightarrow G_1$ – эпиморфизм. Тогда $\ker f \triangleleft G$, причем группа G_1 изоморфна факторгруппе $G / \ker f$. Если H – нормальная подгруппа группы G , то $f : G \rightarrow G/H$, определяемое условием $f(a) = aH$, является эпиморфизмом, причем $\ker f = H$.

Доказательство. Первое утверждение уже доказано. Покажем, что $G_1 \cong G / \ker f$. Легко убедиться, что гомоморфизм $f : G \rightarrow G_1$ постоянен на смежных классах. В самом деле, если $f(g) = g'$, то $f(gh) = f(g)f(h) = f(g) = g'$, где $h \in \ker f$. Поэтому есть сюръекция \bar{f} группы $G / \ker f$ на группу G_1 . Докажем ее инъективность. Если $f(g_1) = f(g_2)$, то $f(g_1)f(g_2)^{-1} = f(g_1g_2^{-1}) = e$, а значит, $g_1 \in g_2 \ker f$. Таким образом, нами построена биекция с условием $\bar{f}(g \ker f) = f(g)$. Условие $\bar{f}(g_1 \ker f)g_2 \ker f) = \bar{f}(g_1 \ker f)\bar{f}(g_2 \ker f)$ следует из условия $f(g_1g_2) = f(g_1)f(g_2)$. Аналогично проверяется третье утверждение. \square

3.5. ГРУППЫ ПОДСТАНОВОК

Обозначим через X конечное множество, а его элементы обозначим через $1, 2, \dots, n$. Рассмотрим все биекции (подстановки) $\sigma : X \rightarrow X$. Легко видеть, что они образуют группу относительно операции композиции отображений. Эта группа называется *симметрической группой n -й степени* и обозначается через S_n или $S(X)$. Нетрудно показать, что $|S_n| = n!$. Так, например, группа S_3 состоит из шести подстановок:

$$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}, \\ \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}.$$

В нижней строке указаны образы элементов 1, 2, 3, расположенных в верхней строке. Условимся при вычислении произведения подстановок $\sigma_1\sigma_2$ выполнять отображения справа налево, т. е. сначала отображение σ_2 , а затем σ_1 . Например:

$$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}.$$

Подгруппы симметрической группы называются *группами подстановок*.

Подстановку вида $1 \rightarrow 2 \rightarrow 3 \rightarrow \dots \rightarrow k \rightarrow 1$ назовем *циклом длины k* и обозначим $(1, 2, \dots, k)$. Два цикла называются *независимыми*, если перемешаемые ими элементы попарно различны. Независимые циклы коммутируют, т. е. для них выполнено условие $\sigma_1\sigma_2 = \sigma_2\sigma_1$. Цикл длины 2 называется *транспозицией*.

Теорема 3.13. *Каждая подстановка единственным образом разложима в произведение независимых циклов.*

Доказательство. Пусть π – произвольная подстановка, i – любой элемент из X . Рассмотрим последовательность $i, \pi(i), \pi^2(i), \dots$. При некотором k в силу конечности множества X $\pi^k(i) = i$. Таким образом, имеем цикл длины k : $i \rightarrow \pi(i) \rightarrow \pi^2(i) \rightarrow \pi^3(i) \rightarrow \dots \rightarrow \pi^k(i) \rightarrow i$. Если $k < n$, то берем

один из оставшихся элементов и поступаем точно так же. В итоге получаем разложение в произведение независимых циклов $\pi = \sigma_1 \sigma_2 \cdots \sigma_s$.

Пусть $\pi = \alpha_1 \alpha_2 \cdots \alpha_r$ – еще одно разложение, i – элемент, не остающийся на месте под действием π . Тогда найдется по одному циклу, например, σ_1 и α_1 , обладающему тем же свойством, что и π . Для них имеем $\pi^k(i) = \sigma_1^k(i) = \alpha_1^k(i)$. Но цикл определяется действием своих степеней на любой элемент, который не остается на месте. Поэтому $\sigma_1 = \alpha_1$. Далее применима индукция по r или по s . \square

Теорема 3.14. *Каждая подстановка $\tau \in S_n$ является произведением транспозиций.*

Доказательство. В силу предыдущей теоремы достаточно рассмотреть случай цикла. Но это можно сделать, например, так:

$$(1, 2, 3, \dots, k) = (1, k)(1, k-1) \dots (1, 3)(1, 2).$$

\square

Ни о какой единственности не может быть и речи хотя бы потому, что для любой транспозиции τ и подстановки σ имеем $\sigma\tau^2 = \sigma$. Тем не менее характер четности числа k в разложении подстановки в произведении транспозиций $\pi = \tau_1 \tau_2 \dots \tau_k$ определяется подстановкой π однозначно. В самом деле, рассмотрим очевидное равенство

$$\begin{pmatrix} 1 & 2 & \cdots & n \\ a_1 & a_2 & \cdots & a_n \end{pmatrix} (1, 2) = \begin{pmatrix} 1 & 2 & \cdots & n \\ a_2 & a_1 & \cdots & a_n \end{pmatrix},$$

которое показывает, что умножение подстановки на транспозицию меняет характер четности перестановки a_1, \dots, a_n на противоположный. Следовательно, если транспозиции τ_1, \dots, τ_k приводят перестановку a_1, \dots, a_n к виду $1, \dots, n$, то $\pi = \tau_k \cdots \tau_1$, и наоборот, поэтому характер четности подстановки π совпадает с характером четности перестановки a_1, \dots, a_n . Подстановка называется *четной* или *нечетной* в зависимости от того, четно или нечетно число k .

Теорема 3.15. *При $n > 1$ количество четных подстановок равно количеству нечетных подстановок и равно $n!/2$.*

Доказательство. Утверждение вытекает из аналогичного свойства перестановок. \square

Нетрудно показать, что все четные перестановки образуют подгруппу группы S_n . Эта подгруппа называется *знакопеременной группой* и обозначается через A_n . При $n > 1$ имеем разложение $S_n = A_n \cup (1, 2)A_n$. Поэтому $[S_n : A_n] = 2$.

Для любой подстановки $\pi \in S_n$ смежные классы πA_n и $A_n \pi$ состоят из всех четных или всех нечетных подстановок в зависимости от четности подстановки π . Поэтому $A_n \triangleleft S_n$.

Теорема 3.16 (теорема Кэли). *Всякая конечная группа G изоморфна подгруппе симметрической группы S_n , где $n = |G|$.*

Доказательство. Обозначим через e, g_1, \dots, g_{n-1} элементы группы G . Рассмотрим отображение $\tau_a(g) = ag$, где a – фиксированный элемент, $a \in G$. Имеем подстановку

$$\tau_a = \begin{pmatrix} e & g_1 & \cdots & g_{n-1} \\ a & ag_1 & \cdots & ag_{n-1} \end{pmatrix}.$$

Вместе с тем получаем отображение $G \rightarrow S_n$, при котором $a \rightarrow \tau_a$. Убедимся, что это – искомый мономорфизм. Данное отображение инъективно, поскольку различным элементам a и b отвечают различные подстановки τ_a и τ_b . В первом случае $e \rightarrow a$, а во втором $e \rightarrow b$. Осталось проверить условие $\tau_{ab} = \tau_a \tau_b$. Убедиться в равенстве двух подстановок можно, проверив их действие на произвольном элементе группы:

$$\tau_{ab}(x) = abx, \quad (\tau_a \tau_b)x = \tau_a(\tau_b(x)) = abx.$$

□

3.6. ДЕЙСТВИЕ ГРУППЫ НА МНОЖЕСТВЕ

Пусть G – произвольная конечная группа, X – конечное множество из n элементов. Будем говорить, что G *действует на* X , если задан любой гомоморфизм $G \rightarrow S(X)$. Тем самым задано отображение декартова произведения $G \times X$ в множество X . Если $g \rightarrow \pi \in S(X)$, то $(g, x) \rightarrow \pi(x)$. Вместо (g, x) будем писать $gx = \pi(x)$. При этом выполняются очевидные свойства:

$$e(x) = x, \quad x \in X; \quad (gh)x = g(h(x)), \quad g, h \in G.$$

Два элемента $x, x' \in X$ называются *эквивалентными относительно группы* G , действующей на X , если $x' = gx$. Легко проверяются свойства рефлексивности, транзитивности и симметричности. Соответствующие классы эквивалентности называются *орбитами*. Орбиту, содержащую элемент x_0 , удобно обозначать символом $G(x_0)$, т. е. $G(x_0) = \{gx_0 \mid g \in G\}$. Например, $S_n(1) = \{1, 2, \dots, n\}$.

Пусть x_0 – элемент из X . Рассмотрим множество $St(x_0) = \{g \in G \mid gx_0 = x_0\}$. Легко убедиться, что $St(x_0)$ – подгруппа в G . Она называется *стабилизатором* элемента x_0 . Для рассмотренного примера $St(1)$ – множество всех подстановок, оставляющих элемент 1 на месте. Очевидно, $St(1) \cong S_{n-1}$, т. е. это фактически симметрическая группа на множестве $2, 3, \dots, n$.

Теорема 3.17. $\text{Card } G(x_0) = [G : St(x_0)]$.

Доказательство. $gx_0 = g_1x_0 \Leftrightarrow x_0 = g^{-1}g_1x_0 \Leftrightarrow g^{-1}g_1 \in St(x_0) \Leftrightarrow g_1 \in gSt(x_0)$. Поэтому левые смежные классы находятся во взаимно однозначном соответствии с элементами орбиты $G(x_0)$. □

Из этой теоремы и теоремы Лагранжа следует, что длина любой орбиты конечной группы является делителем порядка группы.

Группа перестановок $G \subset S_n$, действующая на множестве $X = \{1, 2, \dots, n\}$, называется *транзитивной*, если орбита некоторой (а значит, и любой) точки совпадает со всем множеством X . Транзитивными будут вся группа S_n и, как нетрудно убедиться, знакопеременная группа A_n . Определим действие группы G на левых смежных классах по подгруппе H по правилу $g(g_1H) = (g_1g_2)H$. В этом случае также имеем дело с транзитивностью. В самом деле, если g_1H и g_2H – два смежных класса, то $g_2g_1^{-1}g_1H = g_2H$.

3.7. КОЛЬЦА И ПОЛЯ

Кольцом называется множество R с двумя бинарными операциями, обозначаемыми символами $+$ и \cdot , такими, что:

- 1) R – абелева группа относительно операции $+$;
- 2) операция умножения ассоциативна, т. е. для всех $a, b, c \in R$ $(ab)c = a(bc)$;
- 3) выполняются законы дистрибутивности, т. е. для всех $a, b, c \in R$

$$a(b + c) = ab + ac \text{ и } (b + c)a = ba + ca.$$

Условимся называть нейтральный элемент аддитивной группы кольца *нулем* и обозначать его символом 0. Противоположный к a элемент обозначают через $-a$. Вместо $a + (-b)$ обычно пишут $a - b$. Легко доказываются свойства $a0 = 0a = 0$ для всех $a \in R$. Из этого следует, что $(-a)b = a(-b) = -ab$ для всех $a, b \in R$. Простейшими примерами колец являются кольца целых чисел \mathbb{Z} и многочленов $\mathbb{R}[x]$ с вещественными коэффициентами.

Кольцо называется *кольцом с единицей*, если оно имеет мультипликативную единицу, т. е. такой элемент e , что $ae = ea = a$ для любого $a \in R$.

Кольцо называется *коммутативным*, если операция умножения коммутативна.

Два элемента кольца $a \neq 0, b \neq 0$ называются *делителями нуля*, если $ab = 0$. Приведем пример делителей нуля. Рассмотрим кольцо классов вычетов \mathbb{Z}_m по модулю m . Оно состоит из элементов $\bar{0}, \bar{1}, \bar{2}, \dots, \bar{m-1}$. Операция сложения над этими элементами была определена ранее. Аналогично определяется умножение. Выполняем обычное умножение чисел и при необходимости берем остаток от деления на m . Если m – составное, $m = ab$, то делителями нуля будут \bar{a}, \bar{b} .

Кольцо называется *областью целостности*, если оно является коммутативным кольцом с единицей и без делителей нуля.

Коммутативное кольцо называется *полем*, если его ненулевые элементы образуют группу относительно операции умножения. Отметим простейшие свойства полей.

Свойство 3.1. В поле нет делителей нуля.

Равенство $ab = 0$ при $a \neq 0$ влечет $a^{-1}ab = a^{-1}0$, а значит, $b = 0$.

Свойство 3.2. В поле второй закон дистрибутивности вытекает из первого.

Теорема 3.18. Конечная область целостности является полем.

Доказательство. Обозначим через a_1, a_2, \dots, a_n элементы кольца R . Фиксируем ненулевой элемент $a \in R$ и рассмотрим последовательность элементов aa_1, aa_2, \dots, aa_n . Все они попарно различны, поскольку $aa_1 = aa_2$ влечет $a_1 = a_2$. Поэтому в нашей последовательности имеется единичный элемент $e = aa_i$. Тем самым доказана обратимость любого элемента $a \neq 0$, так как $a(a_1 - a_2) = 0$ и нет делителей нуля. \square

Теорема 3.19. Кольцо классов вычетов \mathbb{Z}_m будет областью целостности, а значит и полем, лишь при простом m .

Доказательство. Если m – составное, то в кольце \mathbb{Z}_m есть делители нуля, а значит, оно не может являться полем. Обратно: если $m = p$ – простое, то для любого ненулевого элемента $a \in \mathbb{Z}_m$ можно указать обратный, что легко следует из разрешимости сравнения $ax \equiv 1 \pmod{p}$, так как $a \in \{1, 2, \dots, p-1\}$, т. е. $(a, p) = 1$. \square

Поле \mathbb{Z}_p называется *полем Галуа* порядка p и обозначается через \mathbb{F}_p .

3.8. ПОДКОЛЬЦА

Подмножество S кольца R называется *подкольцом* этого кольца, если оно замкнуто относительно имеющихся операций сложения и умножения и само образует кольцо относительно этих операций.

Подкольцо H кольца R называется *идеалом* (*двусторонним идеалом*) этого кольца, если для всех $a \in H, r \in R$ имеет место $ar \in H, ra \in H$.

Примеры. Множество целых чисел \mathbb{Z} является подкольцом поля рациональных чисел \mathbb{Q} , но не идеалом. Легко подобрать одно целое и одно рациональное число, произведение которых не будет целым.

Пусть R – коммутативное кольцо, $a \in R$. Положим $H = \{ar \mid r \in R\}$. Тогда H – идеал кольца R . В частности, числа, кратные данному модулю m , в кольце \mathbb{Z} образуют идеал $m\mathbb{Z}$.

Пусть R – коммутативное кольцо. Идеал H кольца R называется *главным идеалом* кольца R , если существует элемент $a \in R$ такой, что $H = \{ar \mid r \in R\}$. В этом случае H называют также *главным идеалом, порожденным элементом* a .

Пусть кольцо R содержит единичный элемент. Рассмотрим циклическую подгруппу аддитивной группы кольца, порожденную единицей. Она автоматически будет подкольцом, так как

$$\underbrace{(1 + \dots + 1)}_m \underbrace{(1 + \dots + 1)}_n = \underbrace{(1 + \dots + 1)}_{mn}.$$

Она автоматически будет изоморфна либо аддитивной группе кольца \mathbb{Z} , либо аддитивной группе одного из колец вычетов \mathbb{Z}_m . В первом случае говорят, что *характеристика* кольца R равна нулю, $\text{char } R = 0$. Во втором случае полагают $\text{char } R = m$.

Теорема 3.20. *Характеристика области целостности либо равна нулю, либо является простым числом.*

Доказательство. Пусть циклическая группа, порождаемая единицей кольца R , есть \mathbb{Z}_m . Если бы m было составным, $m = m_1 m_2$, то в силу очевидного равенства

$$\underbrace{(1 + \dots + 1)}_{m_1} \underbrace{(1 + \dots + 1)}_{m_2} = \underbrace{(1 + \dots + 1)}_m = 0$$

это кольцо имело бы делители нуля. \square

Центром кольца R называется множество всех его элементов $a \in R$, для которых $ax = xa$ при всех $x \in R$.

Центр коммутативного кольца R совпадает с R .

Теорема 3.21. *Центр любого кольца является его подкольцом.*

Доказательство. Действительно, если $ax = xa, bx = xb$ при всех x , то

$$(a \pm b)x = ax \pm bx = xa \pm xb = x(a \pm b),$$

$$(ab)x = a(bx) = a(xb) = (ax)b = (xa)b = x(ab).$$

При наличии единицы e верно $ex = xe$. \square

Если H – идеал кольца R , то факторгруппа R/H также наделяется структурой кольца. Определим умножение смежных классов по формуле

$$(a + H)(b + H) = ab + H.$$

Убедимся в корректности определения. Пусть $a + h_1$ и $b + h_2$ – другие представители смежных классов. Тогда $(a + h_1)(b + h_2) = ab + ah_2 + bh_1 + h_1 h_2$. Три последних слагаемых принадлежат идеалу, поэтому имеем тот же смежный класс. Смежные классы по идеалу называют также *классами вычетов по модулю идеала*. При таком определении классы вычетов наследуют аксиомы кольца, и поэтому они образуют кольцо, называемое *факторкольцом* кольца R по идеалу H . Оно обозначается R/H .

Фактически мы уже имели дело с факторкольцом $\mathbb{Z}/m\mathbb{Z}$. Эти факторкольца можно задавать с помощью таблиц Кэли. В случае $p = 3$ имеем таблицы умножения и сложения поля Галуа \mathbb{F}_3 :

$$\begin{array}{c|ccc} \cdot & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 2 \\ 2 & 0 & 2 & 1 \end{array}; \quad \begin{array}{c|ccc} + & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 2 \\ 1 & 1 & 2 & 0 \\ 2 & 2 & 0 & 1 \end{array}.$$

Следует отметить, что факторкольцо наследует не все свойства кольца. Например, в кольце $\mathbb{Z}/6\mathbb{Z}$ есть делители нуля $\bar{2} \cdot \bar{3} = 0$, хотя в самом кольце \mathbb{Z} их нет.

3.9. ГОМОМОРФИЗМЫ КОЛЕЦ

В дальнейшем мы чаще всего будем рассматривать кольца с единицами.

Пусть R и S – кольца. *Гомоморфизмом* $\varphi : R \rightarrow S$ называется отображение, для которого

$$\varphi(a + b) = \varphi(a) + \varphi(b), \varphi(ab) = \varphi(a)\varphi(b), \varphi(e) = e'$$

при всех $a, b \in R$.

Термины «ядро», «образ», «эпиморфизм», «мономорфизм», «изоморфизм», «автоморфизм» и «эндоморфизм» имеют тот же смысл, что и для групп.

Теорема 3.22. *Если φ – гомоморфизм кольца R на кольцо S , то $\ker \varphi$ – идеал кольца R , причем кольцо S изоморфно факторкольцу $R/\ker \varphi$. Обратно: если H – идеал кольца R , то отображение $\bar{\varphi} : R \rightarrow R/H$, определяемое условием $\varphi(a) = a + H$, является эпиморфизмом R на R/H с ядром H .*

Доказательство этой теоремы аналогично доказательству соответствующей теоремы для групп.

Теорема 3.23. *В любой области целостности R положительной характеристики p отображение $x \rightarrow x^p$ является мономорфизмом $\varphi : R \rightarrow R$.*

Доказательство. Воспользуемся формулой бинома:

$$\varphi(a + b) = (a + b)^p = a^p + C_p^1 a^{p-1} b + \dots + C_p^p b^p.$$

Нетрудно показать, что все биномиальные коэффициенты, кроме крайних, делятся на p . Поэтому соответствующие им члены в разложении вообще отсутствуют. Итак, $\varphi(a + b) = a^p + b^p = \varphi(a) + \varphi(b)$. Далее, $\varphi(ab) = (ab)^p = a^p b^p = \varphi(a)\varphi(b)$, $\varphi(e) = e$ в любом коммутативном кольце. \square

В заключение отметим, что в кольце \mathbb{Z} любой идеал главный, так как является аддитивной подгруппой бесконечной циклической группы.

3.10. ЕВКЛИДОВЫ КОЛЬЦА

Теория делимости целых чисел может быть значительно обобщена. В частности, ее можно развить для колец многочленов над произвольными полями.

Евклидовым кольцом называется область целостности R вместе с нормой $\nu : R^* \rightarrow \mathbb{N} \cup \{0\}$ (R^* – множество ненулевых элементов кольца, $\mathbb{N} \cup \{0\}$ – неотрицательные целые числа), которая удовлетворяет следующим условиям:

$$\nu(ab) \geq \nu(a) \text{ для любых } x, y \in R^*;$$

для любых $a \in R$, $b \in R^*$ существуют элементы q и r такие, что

$$a = bq + r, \text{ где либо } r = 0, \text{ либо } \nu(r) < \nu(b).$$

Евклидовым кольцом является кольцо \mathbb{Z} вместе с нормой, являющейся обычным модулем. Кольцо многочленов $F[x]$ над любым полем F также является евклидовым. В качестве нормы следует взять степень многочлена. Менее очевидно, что евклидово кольцо образуют целые гауссовские числа, т. е. числа вида $a + bi$, $a, b \in \mathbb{Z}$. Здесь в качестве нормы следует взять квадрат модуля $\nu(a + bi) = a^2 + b^2$. Эти числа обозначают $\mathbb{Z}[i]$.

Теорема 3.24. В евклидовом кольце R для любых $a, b \in R^*$

$$\nu(ab) = \nu(a),$$

если b обратим, и $\nu(ab) > \nu(a)$ – в противном случае.

Доказательство. Если b обратим, то

$$\nu(a) = \nu(abb^{-1}) \geq \nu(ab) \Rightarrow \nu(ab) = \nu(a),$$

поскольку $\nu(ab) \geq \nu(a)$. Пусть теперь b не обратим. В этом случае ab не делит a . В самом деле, $ab \mid a \Rightarrow a = (ab)q$. После сокращения на a имеем $bq = e$. Поэтому

$$a = (ab)q + r, \text{ где } r \neq 0, \nu(r) < \nu(ab).$$

Кроме того, $r = a - (ab)q = a(e - qb)$. Поэтому $\nu(r) \geq \nu(a)$. Окончательно получаем $\nu(ab) > \nu(r) \geq \nu(a)$. \square

Следствие 3.2. В евклидовом кольце элемент a обратим тогда и только тогда, когда $\nu(a) = \nu(e)$.

Например, обратимые элементы кольца \mathbb{Z} – это ± 1 . В кольце целых гауссовых чисел обратимыми будут четыре элемента: $\pm 1, \pm i$.

Теорема 3.25. В евклидовом кольце все идеалы главные.

Доказательство. Оно аналогично доказательству теоремы о том, что каждая подгруппа циклической группы циклическая. Если H – идеал, то он будет порожден элементом a с наименьшим значением нормы $\nu(a)$, $a \in H$. \square

В силу наличия в евклидовом кольце алгоритма деления с остатком для него можно построить теорию делимости, аналогичную теории делимости для кольца целых чисел. В частности, можно ввести понятия НОК и НОД двух элементов. Для нахождения НОД можно использовать алгоритм Евклида. В евклидовом кольце любой элемент можно разложить в произведение простых элементов. Однако это разложение менее определенное, чем каноническое разложение целого числа, из-за наличия в произвольном евклидовом кольце обратимых элементов. Отметим в связи с этим, что все обратимые элементы евклидова кольца образуют группу по умножению. Она называется *группой единиц кольца* и обозначается через $U(R)$. Например, $U(\mathbb{Z}) = \pm 1$, $U(\mathbb{Z}[i]) = \{\pm 1, \pm i\}$, $U(R[x]) = R^*$. Назовем элемент $p \in R^*$ *простым*, если он не обратим и не раскладывается в произведение двух не обратимых множителей. Два простых элемента называются *ассоциированными*, если они отличаются обратимым множителем, т. е. $p = uq$, где $u \in U(R)$. Например, 5 и -5 – простые ассоциированные элементы в кольце \mathbb{Z} . Отношение ассоциированности является бинарным отношением эквивалентности. Поэтому имеем разбиение простых элементов на непересекающиеся классы ассоциированных.

Теорема 3.26. *В евклидовом кольце всякий ненулевой не обратимый элемент можно представить в виде произведения степеней попарно не ассоциированных простых элементов. В этом разложении классы простых элементов и их степени определены однозначно.*

Доказательство этой теоремы аналогично доказательству теоремы о каноническом разложении натурального числа.

3.11. ПРОСТЫЕ И МАКСИМАЛЬНЫЕ ИДЕАЛЫ

Идеал H кольца R называется *простым*, если $ab \in H \Rightarrow a \in H$ либо $b \in H$. Идеал H кольца R называется *максимальным*, если он не содержится ни в каком большем идеале, кроме самого кольца R .

Теорема 3.27. *Пусть R – коммутативное кольцо с единицей, H – идеал кольца R .*

1. *Идеал H прост тогда и только тогда, когда факторкольцо R/H является областью целостности.*

2. *Идеал H максимальен тогда и только тогда, когда факторкольцо R/H является полем.*

Доказательство. Пусть идеал H – простой. Необходимо доказать отсутствие делителей нуля в факторкольце. Их наличие означало бы существование элементов $a, b \in H$ таких, что $(a + H)(b + H) = H$, где $a \notin H, b \notin H$. Поскольку $(a + H)(b + H) = ab + H \Rightarrow ab \in H$, то получено противоречие с простотой идеала H . Аналогично доказывается обратное утверждение.

Пусть H максимальен в кольце R . Докажем обратимость любого ненулевого элемента факторкольца $a + H$, $a \notin H$. С этой целью рассмотрим множество элементов

$$S = \{h + ax \mid h \in H, x \in R\},$$

которое, как легко убедиться, является идеалом, большим, чем идеал H , поскольку $a \in S$, $a \notin H$. Поэтому $S = R$. Следовательно, $e = h + ax$ для некоторых $h \in H$, $x \in R$. Поэтому

$$e + H = h + ax + H = ax + H = (a + H)(x + H),$$

т. е. в факторкольце $(a + H)^{-1} = (x + H)$. Обратно: пусть R/H – поле. Обозначим через S идеал строго больший, чем H . Пусть $a \in S$, $a \notin H$. В силу того, что R/H – поле, уравнение

$$(a + H)(x + H) = b + H$$

разрешимо для любого $b \in R$, поскольку $H \neq a + H$. Поэтому $ax + H = b + H$. Но $ax + H \subset S$, так как $H \subset S$. Следовательно, $ax \in H$, т. е. $S = b + S$ для любого b . Но это возможно лишь при $S = R$. Поэтому H является максимальным идеалом. \square

Следствие 3.3. *Всякий максимальный идеал прост.*

Доказанная теорема позволяет строить новые конечные поля, отличные от полей типа \mathbb{F}_p .

Теорема 3.28. *В евклидовом кольце идеал $H = aR$, являющийся автоматически главным, максимальен тогда и только тогда, когда a – простой элемент.*

Доказательство. Пусть идеал $M = aR$ максимальен. Если бы при этом элемент a допускал разложение $a = a_1a_2$, то имело бы место строгое включение $aR \subset a_1R$. Обратно: пусть A – простой элемент. Если при этом $H = aR$ содержится в строго большем собственном идеале bR , то имеем делимость $b \mid a$. \square

Пример 3.1. Многочлен $x^2 + x + 1$ является простым элементом кольца многочленов над полем \mathbb{F}_2 , поскольку его приводимость означает наличие корней в этом поле. Здесь, конечно, мы используем то обстоятельство, что он может разлагаться лишь на два множителя первой степени. Представителями классов вычетов являются элементы $0, 1, x, x + 1$. Над ними можно производить сложение и умножение по модулю $x^2 + x + 1$. Тем самым построено поле \mathbb{F}_4 . Приведем таблицы умножения и сложения в этом поле:

.	0	1	x	$x + 1$	+	0	1	x	$x + 1$
0	0	0	0	0	0	0	1	x	$x + 1$
1	0	1	x	$x + 1$;	1	1	0	$x + 1$
x	0	x	$x + 1$	1	x	x	$x + 1$	0	1
$x + 1$	0	$x + 1$	1	x	$x + 1$	$x + 1$	x	1	0

3.12. КОНЕЧНЫЕ РАСШИРЕНИЯ ПОЛЕЙ

Пусть F и P – два поля, причем $F \subset P$. Тогда F называется *подполем* поля P , которое, в свою очередь, называется *надполем* поля F или его *расширением*. Каждое поле содержит так называемое *простое подполе*, т. е. поле, порожденное единицей. Если характеристика поля положительна (равна p), то простое подполе изоморфно \mathbb{F}_p . Если характеристика равна нулю, то простое подполе изоморфно полю рациональных чисел. Таким образом, каждое поле является расширением своего простого подполя.

Поле P можно рассматривать как векторное пространство над полем F . Размерность этого пространства называется *степенью расширения* $F \subset P$ и обозначается $[P : F]$. Например: $[\mathbb{C} : \mathbb{R}] = 2$, $[\mathbb{R} : \mathbb{Q}] = \infty$, $[\mathbb{F}_4 : \mathbb{F}_2] = 2$.

Теорема 3.29. *Пусть даны два расширения $F \subset P$, $P \subset K$ конечной степени. Тогда*

$$[K : F] = [P : F][K : P].$$

Доказательство. Пусть элементы p_1, p_2, \dots, p_s образуют базис P над F , а элементы k_1, k_2, \dots, k_r – базис K над P . Покажем, что rs элементов вида $p_i k_j$ образуют базис K над F . Любой элемент $k \in K$ можно записать в виде $k = \sum_i \alpha_i k_i$, $\alpha_i \in P$. В свою очередь, $\alpha_i = \sum_j \beta_{ij} p_j$. Значит, $k = \sum_{i,j} \beta_{ij} p_j k_i$. Поэтому элементы $p_i k_j$ являются системой образующих пространства K над полем F . Осталось доказать их линейную независимость. Пусть $\sum_{i,j} \alpha_{ij} p_i k_j = 0$

при некоторых $\alpha_{ij} \in F$. Тогда

$$0 = \sum_{i,j} \alpha_{ij} p_i k_j = \sum_j \left(\sum_i \alpha_{ij} p_i \right) k_j \Rightarrow \sum_i \alpha_{ij} f_i = 0 \Rightarrow \alpha_{ij} = 0.$$

□

Элемент $\alpha \in P$ называется *алгебраическим* над полем F , если он является корнем многочлена $f(x) \in F[x]$, $f(\alpha) = 0$. Многочлен $f(x)$ называется *аннулирующим многочленом* элемента α . Среди всех аннулирующих многочленов можно выбрать многочлен наименьшей степени со старшим коэффициентом, равным единице. Такой многочлен называется *минимальным многочленом* элемента α . Минимальный многочлен очевидно неприводим, т. е. не разлагается в произведение многочленов меньшей степени. Покажем, что минимальный многочлен определен однозначно. Пусть $m_1(x)$ и $m_2(x)$ – различные минимальные многочлены. Поскольку их старшие коэффициенты одинаковы (равны единице), а степени равны, то их разность $\varphi(x) = m_1(x) - m_2(x)$ будет ненулевым многочленом меньшей степени. Вместе с тем $\varphi(\alpha) = m_1(\alpha) - m_2(\alpha) = 0$, что противоречит условию.

Теорема 3.30. *Всякое расширение $F \subset P$ конечной степени алгебраично, т. е. все элементы поля P алгебраичны над полем F .*

Доказательство. Для произвольного элемента $\alpha \in P$ рассмотрим его степени 1, $\alpha, \alpha^2, \dots, \alpha^n$, где $n = [P : F]$. Поскольку их количество превышает размерность пространства, они линейно зависимы, т. е. найдутся скаляры $f_0, f_1, \dots, f_n \in F$ такие, что

$$f_0 + f_1\alpha + f_2\alpha^2 + \cdots + f_n\alpha^n = 0,$$

тем самым найден аннулирующий многочлен. \square

Теорема 3.31. Все аннулирующие многочлены для данного элемента $\alpha \in P$ образуют главный идеал в кольце $F[x]$. Он порожден минимальным многочленом $m(x)$, т. е. имеет вид $m(x)F[x]$.

Доказательство. Достаточно показать, что любой аннулирующий многочлен делится на минимальный. Пусть $f(\alpha) = 0$. Разделим $f(x)$ с остатком на $m(x)$. Пусть $f(x) = m(x)g(x) + r(x)$. Тогда $f(\alpha) = m(\alpha)g(\alpha) + r(\alpha) = 0$. Поэтому $r(\alpha) = 0$, что возможно лишь при $r(x) = 0$. \square

Пусть $F \subset P$ – произвольное расширение, $\alpha \in P$. Обозначим через $F[\alpha]$ наименьшее подкольцо поля P , содержащее элемент α и поле F , а через $F(\alpha)$ – наименьшее подполе с аналогичным условием. Очевидно, что

$$F[\alpha] = \{f(\alpha) \mid f(x) \in F[x]\},$$

$$F(\alpha) = \left\{ \frac{f(\alpha)}{g(\alpha)} \mid f(x), g(x) \in F[x], g(x) \neq 0 \right\}.$$

Теорема 3.32. Если элемент α алгебраичен над F , то

$$F[\alpha] = F(\alpha) \cong F[x]/m(x)F[x],$$

где $m(x)$ – минимальный многочлен для α .

Доказательство. Рассмотрим эпиморфизм колец $F[x] \rightarrow F[\alpha]$, определенный подстановкой $x \rightarrow \alpha$. Его ядро, что очевидно, является идеалом $m(x)F[x]$. Осталось применить теорему о гомоморфизме кольца на факторкольцо, а поскольку идеал $m(x)F[x]$ максимальен в силу неприводимости многочлена $m(x)$, то кольцо $F[\alpha]$ является полем и поэтому должно совпасть с $F(\alpha)$. \square

$F \subset F[\alpha]$ называется *простым алгебраическим расширением*. Элемент α называется *примитивным элементом* расширения. Степень этого расширения равна степени минимального многочлена. Пусть $m(x) = \alpha_0 + \alpha_1x + \cdots + \alpha_nx^n$. Тогда n элементов 1, $\alpha, \alpha^2, \dots, \alpha^{n-1}$ образуют базис расширения. В самом деле, они линейно независимы, так как их меньше, чем степень минимального многочлена. Пусть $f(\alpha)$ – произвольный элемент поля $F[\alpha]$. Разделим многочлен $f(x)$ с остатком на $m(x)$. Имеем $f(x) = m(x)g(x) + r(x)$. Поэтому $f(\alpha) = r(\alpha)$, что линейно выражается через 1, $\alpha, \alpha^2, \dots, \alpha^{n-1}$.

3.13. ПОЛЕ РАЗЛОЖЕНИЯ

Пусть $f(x)$ – неприводимый многочлен над полем F . Покажем, что для него существует конечное расширение поля F , содержащее все корни многочлена $f(x)$. Такое расширение называют *полем разложения*. Более точно расширение P поля F называют полем разложения многочлена $f(x)$, если:

- 1) $f(x)$ разлагается на линейные множители в $P[x]$;
- 2) P порождено как поле корнями многочлена $f(x)$ и полем F .

Поле разложения – это наименьшее поле, в котором $f(x)$ разлагается на линейные множители.

Теорема 3.33. *Пусть многочлен $m(x)$ неприводим над полем F . Тогда над расширением $P = F[x]/m(x)F[x]$ многочлен $m(x)$ имеет корень $\bar{x} = \varphi(x)$, где φ – канонический эпиморфизм на факторкольцо $\varphi : F[x] \rightarrow P = F[x]/m(x)F[x]$.*

Доказательство. Достаточно применить канонический эпиморфизм к уравнению $m(x) = 0$. \square

Теорема 3.34. *Пусть F – поле, $f(x)$ – произвольный многочлен $f(x) \in F[x]$. Тогда его поле разложения существует и имеет степень над F не большее, чем $n!$.*

Доказательство. Разложим $f(x) = p_1(x)p_2(x)\dots p_r(x)$ на неприводимые множители. Построим поле $P_1 = F[x]/p_1(x)F[x]$ для первого множителя. Получим поле, которое содержит корень α_1 многочлена $p_1(x)$, а значит, и $f(x)$. Пусть над этим полем $f(x) = (x - \alpha_1)f_1(x)$. Применим тот же прием к многочлену $f_1(x)$ над полем P_1 и т. д. Степени построенных полей не будут превышать соответственно $n, n - 1, \dots, 2, 1$. \square

Теорема 3.35. *Поле разложения данного многочлена над полем F определяется с точностью до изоморфизма.*

Доказательство. Поскольку многочлен $f(x)$ разлагается на неприводимые множители, достаточно рассмотреть случай, когда $f(x)$ – неприводим. Пусть u и v – корни неприводимого многочлена в полях G и H .

Тогда отображение $a_0 + a_1u + \dots + a_{n-1}u^{n-1} \longrightarrow a_0 + a_1v + \dots + a_{n-1}v^{n-1}$ определяет изоморфизм $F(u) \longrightarrow F(v)$: $F(u) \cong F[x]/f(x) \cong F(v)$, где $F(u)$ и $F(v)$ – подполя G и H . Применяя это рассуждение несколько раз, устанавливаем изоморфизм $G \cong H$. \square

Поле разложения можно построить и для любого семейства многочленов из $F[x]$. Поле разложения семейства всех непостоянных многочленов называется алгебраическим замыканием \bar{F} поля F . Поле \bar{F} является алгебраическим расширением поля F и обладает дополнительным свойством: любой многочлен ненулевой степени из $\bar{F}[x]$ разлагается над полем \bar{F} на линейные множители. В случае $F = \mathbb{F}_p$, поле $\bar{\mathbb{F}}_p$ содержит в качестве подполя любое поле \mathbb{F}_{p^n} .

3.14. КОНЕЧНЫЕ ПОЛЯ

Всякое конечное поле P содержит простое подполе \mathbb{F}_p . Поскольку поле P конечное, оно имеет конечную степень n над простым подполем. Обозначим через p_1, p_2, \dots, p_n базис P над \mathbb{F}_p . Тогда любой элемент $x \in P$ можно однозначно записать в виде $x = f_1p_1 + f_2p_2 + \dots + f_np_n$, $f_i \in \mathbb{F}_p$. Для каждого коэффициента f_i имеется p возможностей выбора. Поэтому порядок поля p необходимо равен p^n . Однако мы не знаем, для каких n поле \mathbb{F}_{p^n} действительно существует. Наша ближайшая цель – доказать существование поля Галуа \mathbb{F}_{p^n} для любого простого p и натурального n . Положим $q = p^n$.

Лемма 3.1. *Многочлен $f(x) = x^q - x$ не имеет кратных корней в любом поле характеристики p , в котором он разлагается на линейные множители.*

Доказательство. Вычислим производную этого многочлена и покажем, что она взаимно проста с ним: $(x^q - x)' = qx^{q-1} - 1 = p^n x^{q-1} - 1 = -1$. Поэтому $(f(x), f'(x)) = 1$. \square

Заметим, что для любого ненулевого корня многочлена $x^q - x$ верно $\alpha^{q-1} = 1$.

Теорема 3.36. *Поле разложения многочлена $x^q - x$ содержит в точности p^n элементов.*

Доказательство. Достаточно показать, что p^n корней многочлена $x^q - x$ сами образуют поле. Пусть α, β – ненулевые корни этого многочлена, а значит, и многочлена $x^{q-1} - 1$. Тогда $(\alpha\beta)^{q-1} = \alpha^{q-1}\beta^{q-1} = 1 \cdot 1 = 1$. Вместе с тем в силу доказанной формулы бинома в характеристике p имеем $(\alpha + \beta)^p = \alpha^p + \beta^p$. Следовательно, $(\alpha + \beta)^q = \alpha^q + \beta^q = \alpha + \beta$, т. е. $\alpha + \beta$ также является корнем $x^q - x$. \square

Следствие 3.4. *Существует лишь одно с точностью до изоморфизма поле \mathbb{F}_{p^n} .*

Конечное поле \mathbb{F}_{p^n} также называют полем Галуа и обозначают $GF(\mathbb{F}_{p^n})$.

Теорема 3.37. *Мультиликативная группа любого конечного поля P циклична.*

Доказательство. Эта группа – абелева порядка $q - 1$. Допустим, что она не циклична, тогда по теореме 3.5 НОК порядков ее элементов r будет собственным делителем $q - 1$. Поэтому $x_i^r = 1$ для всех $x_i \in P^*$. Следовательно, $\prod_i (x - x_i) \mid (x^r - 1)$. Но это невозможно, так как произведение имеет большую степень, чем r . \square

Образующий элемент циклической группы $\mathbb{F}_{p^n}^*$ называется *примитивным элементом* этого поля. Его можно взять в качестве примитивного элемента расширения $\mathbb{F}_p \subset \mathbb{F}_{p^n}$. Поэтому верна следующая теорема.

Теорема 3.38. Всякое конечное поле характеристики p является простым алгебраическим расширением поля \mathbb{F}_p .

Минимальный многочлен примитивного элемента поля \mathbb{F}_{p^n} имеет степень n . Поэтому можно сказать, что это поле изоморфно факторкольцу $\mathbb{F}_p[x]/m(x)\mathbb{F}_p[x]$ для любого неприводимого многочлена $m(x)$ степени n . При этом $m(x) \mid (x^{q-1} - 1)$.

Определение 3.1. Пусть K – поле из q элементов и F – расширение K степени t . Следом элемента $a \in F$ над K называется величина

$$\mathrm{Tr}_{F/K}(a) = a + a^q + a^{q^2} + \dots + a^{q^{t-1}}.$$

Если K – простое подполе F , то $\mathrm{Tr}_{F/K}(a)$ называется абсолютным следом и обозначается просто $\mathrm{Tr}_F(a)$ или $\mathrm{Tr}(a)$.

Теорема 3.39. Функция $\mathrm{Tr}_{F/K}$ обладает следующими свойствами:

- 1) $\mathrm{Tr}_{F/K}(a^q) = \mathrm{Tr}_{F/K}(a)^q = \mathrm{Tr}_{F/K}(a)$ для всех $a \in F$;
- 2) $\mathrm{Tr}_{F/K}(a) \in K$ для всех $a \in F$;
- 3) $\mathrm{Tr}_{F/K}(a + b) = \mathrm{Tr}_{F/K}(a) + \mathrm{Tr}_{F/K}(b)$ для всех $a, b \in F$;
- 4) $\mathrm{Tr}_{F/K}(\alpha a) = \alpha \mathrm{Tr}_{F/K}(a)$ для всех $\alpha \in K, a \in F$;
- 5) $\mathrm{Tr}_{F/K}$ является сюръективным отображением $F \rightarrow K$.

Доказательство. По теореме 3.37 для любого $a \in K$ верно $a^q = a$.

Поэтому

$$\begin{aligned} \mathrm{Tr}_{F/K}(a^q) &= a^q + (a^q)^q + \dots + (a^q)^{q^{t-1}} = \left(a + a^q + \dots + a^{q^{t-1}} \right)^q = \\ &= (\mathrm{Tr}_{F/K}(a))^q = \mathrm{Tr}_{F/K}(a). \end{aligned}$$

Тем самым доказано первое утверждение. Поскольку $\mathrm{Tr}_{F/K}(a)$ удовлетворяет уравнению $x^q = x$, то элемент $\mathrm{Tr}_{F/K}(a)$ лежит в поле из q элементов, т. е. в K . Это доказывает второе утверждение. Остальные легко выводятся из доказанного. \square

Теорема 3.40. Линейные отображения $F \rightarrow K$ исчерпываются отображениями вида

$$L_b: a \mapsto \mathrm{Tr}_{F/K}(ab), \quad b \in F.$$

Доказательство. Всего имеется q^m различных линейных отображений $F \rightarrow K$. Линейное отображение $\varphi: F \rightarrow K$ однозначно задается выбором элементов $\varphi(a_1), \dots, \varphi(a_m)$, где a_1, \dots, a_m – базис F над K .

Отображения L_b и L_c различны для $b \neq c$. В самом деле, если $L_b(a(b - c)^{-1}) = L_c(a(b - c)^{-1})$ для всех $a \in F$, то $\mathrm{Tr}_{F/K}(a) = 0$ для всех a . Таким образом, имеется q^m отображений $L_b, b \in F$. \square

3.15. ПОРЯДКИ НЕПРИВОДИМЫХ МНОГОЧЛЕНОВ

Теорема 3.41. Поле \mathbb{F}_{p^n} является полем разложения всякого неприводимого многочлена $f(x)$ степени n над полем \mathbb{F}_p .

Доказательство. Пусть α – корень $f(x)$. Тогда $[\mathbb{F}_p(\alpha) : \mathbb{F}_p] = n$. Поэтому

$$\mathbb{F}_p(\alpha) = \mathbb{F}_{p^n}.$$

Применяя формулу бинома в характеристике p , получаем

$$f(\alpha^p) = (a_0\alpha^n + a_1\alpha^{n-1} + \dots + a_0)^p = (f(\alpha))^p = 0.$$

Следовательно, элементы $\alpha, \alpha^p, \alpha^{p^2}, \dots, \alpha^{p^{n-1}}$ являются корнями многочлена $f(x)$. Остается показать, что все они различны. Пусть $\alpha^{p^j} = \alpha^{p^k}$, где $0 < j < k \leq n - 1$. Возведя это равенство в степень p^{n-k} , получаем $\alpha^{p^{n-k+j}} = \alpha^{p^n} = \alpha$. Следовательно, неприводимый многочлен $f(x)$ делит $x^{n-k+j} - x$. Это означает, что поле \mathbb{F}_{p^n} является подполем поля \mathbb{F}_{p^l} , где $l = n - k + j < n$. Противоречие. \square

Следствие 3.5. Порядки всех корней неприводимого многочлена равны.

Доказательство. Корни многочлена $\alpha, \alpha^p, \dots, \alpha^{p^{n-1}}$ имеют одинаковый порядок в мультиликативной группе поля \mathbb{F}_{p^n} в силу теоремы 3.7. \square

Назовем порядком многочлена $f(x) \in \mathbb{F}_p[x]$, $f(0) \neq 0$, наименьшее натуральное e , при котором многочлен $f(x)$ делит $x^e - 1$. Порядок обозначается $\text{ord } f(x)$.

Следствие 3.6. Пусть $f(x)$ – неприводимый многочлен степени n над полем \mathbb{F}_p . Порядок этого многочлена совпадает с порядком любого его корня в мультиликативной группе поля разложения.

Минимальный многочлен образующего элемента группы \mathbb{F}_q^* имеет степень n и неприводим над \mathbb{F}_p . Другими словами, это такой неприводимый многочлен f_x , что

$$f_x | x^{q-1} - 1 \quad \text{и} \quad f_x \nmid x^l - 1,$$

где $l < p^n - 1$, т. е. $\text{ord } f(x) = p^n - 1$. Такой многочлен называется примитивным. Далее будем считать, что он является нормированным, т. е. его старший коэффициент равен 1.

Теорема 3.42. Имеется всего $\frac{1}{n} \varphi(p^n - 1)$ примитивных многочленов степени n над полем \mathbb{F}_p .

Доказательство. У циклической группы $\mathbb{F}_{p^n}^*$ имеется $\varphi(p^n - 1)$ образующих. Корнями одного примитивного многочлена будут ровно n из них. \square

3.16. ЧИСЛО НЕПРИВОДИМЫХ МНОГОЧЛЕНОВ

Сейчас мы хотим найти число нормированных неприводимых множеств степени n над полем \mathbb{F}_p . Это число обозначают через $N_p(n)$.

Мы видим, что поле \mathbb{F}_q , $q = p^n$, является полем разложения всякого неприводимого многочлена $f(x)$ степени n . Это означает, что $f(x)|(x^q - x)$. Поскольку $k|n \Rightarrow \mathbb{F}_{p^k} \subset \mathbb{F}_{p^n}$, то поле \mathbb{F}_q содержит поле разложения всякого неприводимого многочлена, степень которого делит n . Отсюда получается следующая теорема.

Теорема 3.43. $x^q - x = \prod_{\deg f(x)|n} f(x)$, где $f(x)$ – неприводимый нормированный многочлен.

Доказательство. В каноническом разложении многочлена $x^q - x$ встречаются все многочлены $f(x)$ ровно по одному разу, так как многочлен $x^q - x$ не имеет кратных корней.

Переходя к степеням в каноническом разложении многочлена $x^q - x$, получаем некоторую информацию о числе $N_p(n)$. \square

Следствие 3.7. $p^n = \sum_{d|n} dN_p(d)$, где d – натуральный делитель n .

Чтобы получить явную формулу для $N_p(n)$, нам потребуется так называемая функция Мёбиуса:

$$\mu(n) = \begin{cases} 1, & n = 1, \\ (-1)^k, & n = p_1 p_2 \dots p_k, \\ 0, & p^2|n. \end{cases}$$

Теорема 3.44. $\sum_{d|n} \mu(d) = \begin{cases} 1, & n = 1, \\ 0, & n > 1. \end{cases}$

Доказательство. Пусть $n = p_1^{s_1} p_2^{s_2} \dots p_k^{s_k}$. Тогда

$$\begin{aligned} \sum_{d|n} \mu(d) &= \mu(1) + \sum_{i=1}^k \mu(p_i) + \sum_{1 \leq i_1 < i_2 \leq k} \mu(p_{i_1} p_{i_2}) + \dots = \\ &= 1 + C_k^1(-1) + C_k^2(-1)^2 + \dots = (1 + (-1))^k = 0. \end{aligned}$$

\square

Теорема 3.45. Пусть целочисленные функции натурального аргумента связаны соотношением $H(n) = \sum_{d|n} h(d)$. Тогда

$$h(n) = \sum_{d|n} \mu\left(\frac{n}{d}\right) H(d) = \sum_{d|n} \mu(d) H\left(\frac{n}{d}\right).$$

Доказательство.

$$\begin{aligned} \sum_{d|n} \mu\left(\frac{n}{d}\right) H(d) &= \sum_{d|n} \mu(d) H\left(\frac{n}{d}\right) = \sum_{d|n} \mu(d) \sum_{c|\frac{n}{d}} h(c) = \\ &= \sum_{c|n} \sum_{d|\frac{n}{c}} \mu(d) h(c) = \sum_{c|n} h(c) \sum_{d|\frac{n}{c}} \mu(d) = h(n), \text{ поскольку } \sum_{d|\frac{n}{c}} \mu(d) = \begin{cases} 1, & c = n; \\ 0, & c < n. \end{cases} \end{aligned}$$

□

Теорема 3.46. Число нормированных неприводимых многочленов степени n над полем \mathbb{F}_p находится по формуле

$$N_p(n) = \frac{1}{n} \sum_{d|n} \mu\left(\frac{n}{d}\right) p^d = \frac{1}{n} \sum_{d|n} \mu(d) p^{\frac{n}{d}}.$$

Доказательство. Достаточно применить предыдущую теорему, полагая $h(n) = nN_p(n)$, $H(n) = p^n$. □

3.17. ЛИНЕЙНЫЕ РЕКУРРЕНТНЫЕ ПОСЛЕДОВАТЕЛЬНОСТИ

Последовательность элементов s_0, s_1, \dots поля \mathbb{F}_q , удовлетворяющих условию

$$s_{n+k} = a_{k-1}s_{n+k-1} + a_{k-2}s_{n+k-2} + \dots + a_0s_k, \quad (3.1)$$

где $a_{k-1}, a_{k-2}, \dots, a_0$ – фиксированные элементы поля, называется *линейной рекуррентной* (ЛРП) k -го порядка над полем \mathbb{F}_q . Эта последовательность полностью определяется вектором начального состояния $S_0 = (s_0, s_1, \dots, s_{k-1})$ и коэффициентами $a_{k-1}, a_{k-2}, \dots, a_0$.

С линейной рекуррентной можно связать матрицу

$$A = \begin{pmatrix} 0 & 0 & \dots & 0 & a_0 \\ 1 & 0 & \dots & 0 & a_1 \\ 0 & 1 & \dots & 0 & a_2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & a_{k-1} \end{pmatrix}.$$

Рассмотрим последующие состояния ЛРП $S_1 = (s_1, s_2, \dots, s_k)$, $S_2 = (s_2, s_3, \dots, s_{k+1})$, \dots . Определение (3.1) можно переписать в виде

$$S_i = S_{i-1}A, \quad i = 1, 2, \dots \quad (3.2)$$

Далее рассматриваются лишь ЛРП с условием $a_0 \neq 0$. В этом случае матрица A является элементом группы $GL(k, \mathbb{F}_q)$ всех невырожденных матриц k -го порядка с элементами из поля \mathbb{F}_q . Поскольку эта группа конечна, то матрица A имеет конечный порядок как элемент группы.

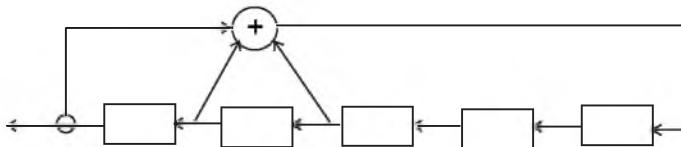
Теорема 3.47. Любая линейная рекуррента при $a_0 \neq 0$ является чисто периодической последовательностью.

Доказательство. Существует натуральное l такое, что $S^l = I_k$, где I_k – единичная матрица. Следовательно, $S_{i+l} = S_i A^l = S_i I_k = S_i$. Это означает, что период ЛРП не превосходит l . \square

Линейные рекурренты можно генерировать с помощью регистров сдвига с обратной связью. Особенно просто такой регистр сдвига устроен в случае двоичного поля. Например, для рекурренты

$$a_{n+5} = a_{n+2} + a_{n+1} + a_n, \quad n = 0, 1, 2, \dots,$$

над полем \mathbb{F}_2 соответствующий регистр имеет вид, представленный на следующем рисунке.



Заполнением триггеров после i -го такта является вектор i -го состояния $A_i = (a_i, a_{i+1}, \dots, a_{i+k-1})$.

Поле $\mathbb{F}_{q=p^n}$ мы строили как факторкольцо $\mathbb{F}_p[x](f(x))$, где $f(x) \in \mathbb{F}_p[x]$ – неприводимый многочлен степени. Элементы факторкольца – многочлены из $\mathbb{F}_p[x]$ степени $< n$. При этом многочлен $\alpha = x$ – корень $f(x)$: $f(\alpha) = f(x) \equiv 0 \pmod{f(x)}$.

Корнями f в \mathbb{F}_q являются также элементы $\alpha^p, \alpha^{p^2}, \dots, \alpha^{p^{n-1}}$. Действительно, по лемме о степени суммы $f(\alpha^{p^i}) = f(\alpha)^{p^i} = 0$.

Лемма 3.2 (о базисе). Набор $\alpha, \alpha^2, \dots, \alpha^n$ является базисом \mathbb{F}_q над \mathbb{F}_p .

Доказательство. Предположим, что указанные элементы линейно зависимы: $\alpha \sum_{i=0}^{n-1} b_i \alpha^i = 0$, где $(b_0, b_1, \dots, b_{n-1}) \neq (0, 0, \dots, 0)$. Тогда α – корень ненулевого многочлена $\sum_{i=0}^{n-1} b_i x^i$.

Пусть $g \in \mathbb{F}_p[x]$ – ненулевой многочлен, для которого α является корнем и который имеет минимальную степень среди всех таких многочленов. Сказанное выше означает, что $\deg g < n$. Выполним деление f на g :

$$f(x) = g(x)h(x) + r(x), \quad \deg r < \deg g.$$

Тогда $r(\alpha) = 0$ и по построению $r = 0$. Но тогда f не является неприводимым. Противоречие. \square

Рассмотрим характеристический многочлен $f(x) = |xE - A|$ матрицы A . Легко подсчитать, что $f(x) = x^k - a_{k-1}x^{k-1} - \dots - a_1x - a_0$. Можно показать, что он будет также и минимальным многочленом матрицы A . Этот многочлен называется характеристическим многочленом ЛРП (3.1).

Теорема 3.48. *Пусть характеристический многочлен $f(x)$ ЛРП s_0, s_1, s_2, \dots неприводим и α – его корень в расширении \mathbb{F}_{p^k} поля \mathbb{F}_p . Тогда существует однозначно определенный элемент $\beta \in \mathbb{F}_{p^k}$ такой, что*

$$s_n = \text{Tr}(b\alpha^n), \quad n = 0, 1, \dots.$$

Доказательство. Легко показать, что если элементы $\{\alpha, \dots, \alpha^k\}$ образуют базис \mathbb{F}_{p^k} над \mathbb{F}_p , то существует однозначно определенное линейное отображение $L: \mathbb{F}_{p^k} \rightarrow \mathbb{F}_p$ такое, что

$$L(\alpha^n) = s_n, \quad n = 0, 1, \dots, k.$$

Но из предыдущей теоремы следует, что имеется однозначно определенный элемент b такой, что $L(a) = \text{Tr}(ba)$ для всех $a \in \mathbb{F}_{p^k}$. Поэтому

$$s_n = \text{Tr}(b\alpha^n), \quad n = 0, 1, \dots, k.$$

Остается проверить 3.1. Для каждого $n = 0, 1, \dots$ имеем

$$\begin{aligned} s_{n+k} - a_{k-1}s_{n+k-1} - \dots - a_1s_{n+1} - a_0s_n &= \text{Tr}(b\alpha^{n+k}) - \sum_{i=0}^{k-1} a_i \text{Tr}(b\alpha^{n+i}) = \\ &= \text{Tr}(b\alpha^n (\alpha^k + a_{k-1}\alpha^{k-1} + \dots + a_1\alpha + a_0)) = \\ &= \text{Tr}(b\alpha^n f(\alpha)) = 0, \end{aligned}$$

что и требовалось установить. \square

3.18. ПОСЛЕДОВАТЕЛЬНОСТИ МАКСИМАЛЬНОГО ПЕРИОДА

Под периодом ЛРП будем понимать ее минимальный период.

Рассмотрим теперь ЛРП с характеристическим многочленом $f(x)$. Нас будет интересовать вопрос о том, когда период ЛРП максимален. Ответ напрямую зависит от многочлена $f(x)$.

Теорема 3.49. *Пусть многочлен $f(x) = |xE - A|$ неприводим. Тогда его порядок совпадает с порядком матрицы S как элемента группы $GL(k, \mathbb{F}_q)$.*

Доказательство. Корни многочлена $f(x)$ являются характеристическими числами матрицы S . Если l – порядок матрицы, то $S^l = E$, и значит, для любого корня α многочлена выполняется $\alpha^l = 1$. Поэтому $\text{ord } f(x) \leq l$. С другой стороны, у матрицы $S^{\text{ord } f(x)}$ все собственные значения равны 1, а ее минимальный многочлен неприводим. Поэтому $S^{\text{ord } f(x)} = E \Rightarrow l \leq \text{ord } f(x)$. \square

Теорема 3.50. *Минимальный период линейной рекурренты с неприводимым характеристическим многочленом $f(x)$, $f(0) \neq 0$, при ненулевом начальном состоянии равен порядку многочлена $f(x)$.*

Доказательство. Пусть $l = \text{ord } f(x) = \text{ord } S$. По теореме 3.47 l – период. Необходимо лишь установить его минимальность. Пусть, например, $A_i = A_j$, $i < j$, $j - i < l$. Это означает, что $A_0 S^i = A_0 S^j$, т. е. $A_0 S^{j-i} = A_0$. Противоречие: у матрицы S^{j-i} не может быть единичного собственного значения. \square

Данная теорема показывает, как для генерации последовательностей максимального периода можно использовать примитивные многочлены.

3.19. ЗАДАНИЯ

1. Выяснить, обладают ли свойствами ассоциативности и коммутативности операции $*$ на множестве A , если:

$$\begin{array}{ll} A = N, \quad x * y = x + 2y; & A = N; \quad x * y = x^y; \\ A = N, \quad x * y = 3xy; & A = N; \quad x * y = \text{НОД}(x, y); \\ A = Z, \quad x * y = x - y; & A = Z; \quad x * y = x^2 + y^2; \\ A = R, \quad x * y = \sin x \cos y; & A = R, \quad x * y = x^{|y|}. \end{array}$$

2. Какие из указанных числовых множеств являются группами относительно заданных операций:

- 1) множество степеней данного вещественного числа с целыми показателями относительно умножения;
- 2) множество комплексных чисел с фиксированным модулем относительно операции умножения;
- 3) множество положительных действительных чисел относительно операции умножения;
- 4) отрезок $[0, 1]$ относительно операции умножения;
- 5) отрезок $[0, 1]$ относительно операции $\alpha * \beta = \{\alpha + \beta\}$;
- 6) корни всех степеней из единицы относительно умножения?

3. Какие из указанных множеств квадратных матриц фиксированного порядка образуют группу относительно операции умножения:

- 1) множество симметрических матриц с вещественными элементами;
- 2) множество невырожденных матриц с вещественными элементами;
- 3) множество целочисленных матриц с определителем равным ± 1 ;
- 4) множество верхних треугольных матриц с вещественными элементами;
- 5) множество ортогональных матриц;
- 6) множество диагональных матриц?

4. Доказать, что множество функций вида $y = (ax + b)(cx + d)^{-1}$, где $a, b, c, d \in \mathbb{R}$, $ad - bc \neq 0$, является группой относительно операции композиции функций.

5. Доказать, что если в группе G выполнено условие $x^2 = e$, $x \in G$, то группа G коммутативна.

6. Доказать, что в любой группе $(ab)^{-1} = b^{-1}a^{-1}$ и вообще $(a_1a_2 \cdots a_n)^{-1} = a_n^{-1}a_{n-1}^{-1} \cdots a_1^{-1}$.

7. Доказать, что множество всех квадратных матриц данного порядка, в каждой строке и каждом столбце которых один элемент равен 1, а остальные – 0, образует группу.

8. Составить таблицу Кэли для циклической группы пятого порядка.

9. Найти с точностью до изоморфизма все группы порядков 3, 4, 6. Составить их таблицы Кэли.

10. Доказать, что в конечной группе любое подмножество H , замкнутое по умножению ($h_1, h_2 \in H \implies h_1h_2 \in H$), будет подгруппой.

11. Доказать, что для нетривиального смежного класса $gH(e \notin gH)$ выполнено условие $h_1, h_2 \in gH \implies h_1h_2 \notin H$.

12. Доказать, что пересечение двух подгрупп будет подгруппой.

13. Доказать, что в любой группе подстановок, содержащей хотя бы одну нечетную подстановку, количество четных подстановок равно количеству нечетных.

14. Найти все подгруппы групп S_3 , Z_6 , Z_{24} .

15. Разложить:

1) аддитивную группу вещественных чисел по подгруппе целых чисел;

2) аддитивную группу комплексных чисел по подгруппе целых гауссовых чисел;

3) симметрическую группу S_n по подгруппе подстановок, оставляющей элемент 1 на месте;

4) группу Z_6 по своим подгруппам;

5) группу S_3 по своим подгруппам.

16. Пусть $a \in G$. Доказать, что множество элементов $\{x; xa = ax, x \in G\}$, называемое централизатором элемента a , является подгруппой.

17. Доказать, что множество элементов $\{x|xa = ax, \forall a \in G\}$, называемое центром группы, является нормальной подгруппой.

18. Найти централизаторы элементов

$$\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \begin{pmatrix} 4 & 3 \\ 5 & 4 \end{pmatrix}, \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

в группе $GL(2, R)$.

19. Найти центр группы $GL(2, R)$.
 20. Показать, что порядки элементов a и a^{-1} равны.
 21. Показать, что порядки элементов ab и ba равны.
 22. Доказать, что аддитивные группы вещественных и рациональных чисел не являются циклическими.
 23. Пусть A и B нормальны в G , $A \cap B = e$. Доказать, что каждый элемент $a \in A$ перестановчен с каждым элементом $b \in B$.
 24. Подгруппа, порожденная элементами вида $aba^{-1}b^{-1}$ (коммутаторами), называется коммутантом. Доказать, что:
 - 1) $aba^{-1}b^{-1} = e \iff ab = ba$;
 - 2) $K \triangleleft G$;
 - 3) факторгруппа G/K – абелева;
 - 4) если G/H – абелева, то $K \subset H$.
 25. Определить с точностью до изоморфизма все абелевы группы порядка 8.
 26. Пусть H – подмножество группы G с условием $h_1, h_2 \in H \implies h_1h_2 \notin H$. Доказать, что $|H| \leq 0,5|G|$.
 27. Вычислить следующие произведения подстановок:
 - 1) $(1, 2)(1, 3)(1, 4)(2, 3)(3, 5)$;
 - 2) $(125)(124)(129)$;
 - 3) f^{100} , где $f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 3 & 5 & 4 & 6 & 9 & 7 & 1 & 10 & 8 & 2 \end{pmatrix}$.
 28. Найти подстановку X , если $AXB^2 = C$,
- $$A = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 7 & 3 & 2 & 1 & 6 & 5 & 4 \end{pmatrix};$$
- $$B = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 3 & 1 & 2 & 7 & 4 & 5 & 6 \end{pmatrix}; \quad C = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 5 & 1 & 3 & 6 & 4 & 7 & 2 \end{pmatrix}.$$
29. Доказать, что две подстановки сопряжены тогда и только тогда, когда имеют одинаковое количество циклов каждой длины.
 30. Доказать, что в булевом кольце $(x^2 = x)$:
 - 1) умножение коммутативно;
 - 2) $x + x = 0$;
 - 3) кольцо не является областью целостности.
 31. Доказать, что в определении кольца с единицей не обязательно требовать, чтобы операция «+» была коммутативной.

32. Доказать, что следующие подмножества являются подкольцами в кольце $M_n(R)$:

- 1) диагональные матрицы;
- 2) верхнетреугольные матрицы.

33. Найти все подкольца колец вычетов $\mathbb{Z}_7, \mathbb{Z}_{10}, \mathbb{Z}_{12}$.

34. Может ли в кольце, не являющемся полем, содержаться некоторое поле?

35. Показать, что эндоморфизмы абелевой группы образуют кольцо.

36. Показать, что биекция $a + b\sqrt{2} \longleftrightarrow a + b\sqrt{3}$ не является изоморфизмом полей $Q(\sqrt{2}), Q(\sqrt{3})$ и что эти поля вообще не изоморфны.

37. Доказать, что $Q(\sqrt{2} + \sqrt{3}) = Q(\sqrt{2} - \sqrt{3}) = Q(\sqrt{2}, \sqrt{3})$.

38. Установить, что I – идеал кольца K . Является ли факторкольцо полем?

- 1) $K = \{a + bi \mid a, b \in \mathbb{Z}\}, I = \{a + bi \mid a, b \in 3\mathbb{Z}\}$.
- 2) $K = \{a + b\sqrt{2} \mid a, b \in \mathbb{Z}\}, I = \{a + b\sqrt{2} \mid a, b \in 3\mathbb{Z}\}$.
- 3) $K = \mathbb{Z}_3[x], I = (x^2 + 1)$.
- 4) $K = \mathbb{Z}_2[x], I = (x^2 + x)$.
- 5) $K = \mathbb{R}[x], I = (x^2 + 1)$.

39. Найдите идеал, порожденный множеством M , если:

- 1) $M = \{3, 5\}$ в кольце \mathbb{Z} ;
- 2) $M = \{4, 10\}$ в кольце \mathbb{Z} ;
- 3) $M = \{x^6 - 1, x^4 - 1\}$ в кольце $\mathbb{R}[x]$;
- 4) $M = \{x, x + 1\}$ в кольце $\mathbb{R}[x]$.

40. Доказать, что факторкольцо $R[x]/(x^4 + x^3 + x + 1)$ не может быть полем ни для какого коммутативного кольца R .

41. Вычислить образ $(2x + 1)^{-1}$ в факторкольце $F[x]/(x^3 - 2)$, где

- 1) $F = Q$;
- 2) $F = \mathbb{Z}_5$;
- 3) $F = \mathbb{F}_7$.

42. Доказать, что $(x^m - 1) \mid (x^n - 1) \iff m \mid n$ над любым полем коэффициентов.

43. Является ли $C[0, 1]$ областью целостности? Показать, что отображение $f \rightarrow f(a)$ – эпиморфизм, а ядро – максимальный идеал.

44. Показать, что если $p(x)$ приводим, то идеал $(p(x))$ немаксимальен.

45. Показать, что $x^2 + x + 1, x^3 + x + 1, x^4 + x + 1$ неприводимы над \mathbb{F}_2 и что нет других неприводимых многочленов 2-й и 3-й степени.

46. Написать таблицы умножения для колец $\mathbb{Z}_2[x]/(x^2 + x + 1)$, $\mathbb{Z}_2[x]/(x^3 + x + 1)$, $\mathbb{Z}_2[x]/(x^3 + x^2 + 1)$.

47. Применив алгоритм Евклида, найти наибольшие общие делители многочленов с коэффициентами из поля F :

- 1) $F = \mathbb{F}_2$, $x^7 + 1, x^5 + x^3 + 1$;
- 2) $F = \mathbb{F}_2$, $x^5 + x + 1, x^6 + x^5 + x^4 + 1$;
- 3) $F = \mathbb{F}_3$, $x^8 + 2x^5 + x^3 + x^2 + 1, 2x^6 + x^5 + 2x^3 + 2x^2 + 2$.

48. Вычислить $f(3)$, если $f(x) = x^{214} + 3x^{152} + 2x^{47} + 2 \in \mathbb{F}_5[x]$.

49. Решить, если возможно, сравнения:

- 1) $(x^2 + 1)f(x) = 1 \pmod{(x^3 + 1)}$ в $\mathbb{F}_3[x]$;
- 2) $(x^4 + x^3 + x^2 + 1)f(x) = x^2 + 1 \pmod{(x^3 + 1)}$ в $\mathbb{F}_2[x]$.

50. Пусть $f(x) \in \mathbb{F}_p[x]$, тогда $(f(x))^p = f(x^p)$. Доказать.

51. Доказать, что в коммутативном кольце характеристики p $(a + b)^{p^n} = a^{p^n} + b^{p^n}$.

52. Доказать неприводимость многочленов $x^2 + 1$, $x^2 + x + 4$ над полем \mathbb{F}_{11} и построить изоморфизм факторкольца

$$\mathbb{F}_{11}[x]/(x^2 + 1) \cong \mathbb{F}_{11}[x]/(x^2 + x + 4).$$

53. Найти порядки многочленов: $x^{10} + x^9 + x^3 + x^2 \in \mathbb{F}_2[x]$, $x^8 + x^7 + x^3 + x + 1 \in \mathbb{F}_2[x]$, $x^7 + x^6 + x^4 - x^2 + x \in \mathbb{F}_3[x]$, $(x^2 + x + 1)^5(x^3 + x + 1) \in \mathbb{F}_2[x]$. Какие из них неприводимы?

54. Найти примитивный многочлен степени 6 над полем \mathbb{F}_2 .

55. Найти примитивный элемент α в $\mathbb{Z}[x]/(x^2 - 2)$, представить степени $\alpha^2, \dots, \alpha^8$ в виде $a + b\sqrt{2}$, $a, b \in \mathbb{F}_3$. Однозначно ли такое представление?

56. Можно ли вложить \mathbb{F}_4 в \mathbb{F}_8 ?

57. При каком условии поле \mathbb{F}_{p^n} можно изоморфно вложить в поле \mathbb{F}_{p^m} ?

58. Найти все примитивные элементы полей \mathbb{F}_7 , \mathbb{F}_9 , \mathbb{F}_{17} .

59. Найти базис поля \mathbb{F}_{25} над простым подполем. Разложить все элементы по этому базису. Найти примитивный элемент α этого поля и для любого элемента $\beta \in \mathbb{F}_{25}$ найти n такое, что $\alpha = \beta^n$.

60. Показать, что каждый элемент конечного поля \mathbb{F}_{p^n} имеет в нем только один корень p -й степени.

61. Доказать, что если I – идеал кольца K , то $I[x]$ – идеал кольца $K[x]$.

62. Показать, что элемент $\sqrt{2} + i$ имеет степень 4 над Q и степень 2 над R . Найти его минимальные многочлены.

63. Доказать, что если многочлен $F(x)$ неприводим в $\mathbb{F}_q[x]$, то $F(ax + b)$ также неприводим, где $a, b \in \mathbb{F}_q$, $a \neq 0$.

64. Разложить многочлены на неприводимые множители: $x^9 + x + 1$ над \mathbb{F}_2 , $x^7 + x^6 + x^5 - x^3 + x^2 - x - 1$ над \mathbb{F}_3 .

65. С помощью матриц дать представление для элементов поля \mathbb{F}_8 , используя многочлен $x^3 + x + 1$. Дать аналогичное представление для \mathbb{F}_{16} , \mathbb{F}_9 .

66. Доказать неприводимость многочлена $x^4 + x + 1$ над \mathbb{F}_2 и построить таблицы операций для его поля корня.

67. Показать, что поле корня $x^3 + x + 1$ над \mathbb{F}_2 есть поле разложения.

68. Найти поля разложения $(x^2 - 3)(x^3 + 1)$ над Q , $(x^2 - 3)(x^2 - 2x - 2)$ над Q , $x^3 + x + 1$ над \mathbb{F}_2 .

69. Найти изоморфизм полей разложения $x^3 + 2x + 1$ и $x^3 + 2x + 2$ над \mathbb{F}_3 .

70. Найти степени неприводимых множителей многочлена $x^{17} - 1$ над \mathbb{F}_2 и его поле разложения.

71. Установить примитивность многочленов $x^6 + x^5 + x^2 + x + 1$ над \mathbb{F}_2 , $x^5 - x + 1$ над \mathbb{F}_3 .

72. Найти хотя бы один примитивный многочлен степени 3 над полем \mathbb{F}_4 .

73. Установить неприводимость, примитивность и порядок $x^4 + x^3 + x^2 - x - 1$ над полем \mathbb{F}_3 .

74. Пусть A – любое множество автоморфизмов поля F . Показать, что элементы, инвариантные относительно всех $\alpha \in A$, образуют подполе $S(A) \subset F$.

75. Над полем \mathbb{F}_2 рассматривается рекуррентное уравнение $s_i - s_{i-2} + s_{i-3} = 0$. Вычислить периоды всех 8-ми последовательностей и их внутрипериодные значения.

76. Построить регистр сдвига с обратной связью, реализующий соотношение из предыдущей задачи.

77. Вычислить порядки матриц

$$A_1 = \begin{pmatrix} 0 & 0 & -1 \\ 1 & 0 & 1 \\ 0 & 1 & -1 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}$$

над полями \mathbb{F}_2 и \mathbb{F}_3 .

78. Вычислить характеристический и минимальный многочлены матрицы S из п. 3.18.

79. Построить последовательность максимального периода над полем \mathbb{F}_3 с периодом не меньше 50.

80. Найти примитивный многочлен четвертой степени над \mathbb{F}_2 . Построить соответствующий регистр сдвига и последовательность периода 15.

Г л а в а 4

ЭЛЛИПТИЧЕСКИЕ КРИВЫЕ

4.1. УРАВНЕНИЕ ВЕЙЕРШТРАССА ЭЛЛИПТИЧЕСКОЙ КРИВОЙ

В последнее время в криптографии все чаще используются эллиптические кривые. Цель данной главы – ознакомить читателя с основными фактами теории эллиптических кривых. К сожалению, из-за ограниченности объема книги дать полное изложение вопроса невозможно. Пропущенные доказательства можно найти в книге [156]. Для понимания содержания главы необходимо знакомство с понятием алгебраического замыкания поля K (определенного как расширение K , в котором каждое алгебраическое уравнение имеет корень). Алгебраические замыкания и их свойства подробно описаны, например, в учебнике Б. Л. ван дер Вардена [7].

Определение 4.1. Эллиптической кривой над полем K называется алгебраическая кривая, которая задается уравнением Вейерштрасса:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \quad (4.1)$$

где $a_1, a_2, a_3, a_4, a_6 \in K$.

Для обозначения эллиптической кривой используют запись E или E/K , чтобы подчеркнуть, что кривая определена над K . Точки, лежащие на эллиптической кривой, являются решениями (x, y) уравнения (4.1) (это так называемые *аффинные точки*); единственная лежащая на E (в проективной плоскости) точка на бесконечности в дальнейшем обозначается \mathcal{O} .

Определение 4.2. Точка эллиптической кривой, координаты которой принадлежат F , где F – некоторое расширение поля K , называется *F-рациональной*. Множество всех *F-рациональных* точек эллиптической кривой E будем обозначать $E(F)$. Под E будем также понимать множество \bar{K} -рациональных точек, где \bar{K} – алгебраическое замыкание K .

Определение 4.3. Точка $P = (x_0, y_0) \in E$ называется *невырожденной* (гладкой), если для многочлена $f(x, y) = y^2 + a_1xy + a_3y - (x^3 + a_2x^2 + a_4x + a_6)$ верно по крайней мере одно из условий:

$$\frac{\partial f}{\partial x}(x_0, y_0) \neq 0 \quad \text{или} \quad \frac{\partial f}{\partial y}(x_0, y_0) \neq 0. \quad (4.2)$$

Эллиптическая кривая E/K называется *невырожденной* (гладкой), если каждая афинная точка $E(\bar{K})$ является невырожденной.

4.2. j -ИНВАРИАНТ И ДИСКРИМИНАНТ ЭЛЛИПТИЧЕСКОЙ КРИВОЙ

Определение 4.4. Дискриминантом эллиптической кривой E называется величина

$$\Delta = -d_2^2 d_8 - 8d_4^3 - 27d_6^2 + 9d_2 d_4 d_6.$$

При $\Delta \neq 0$ j -инвариантом эллиптической кривой E называется величина

$$j(E) = c_4^3 / \Delta,$$

где

$$\begin{aligned} d_2 &= a_1^2 + 4a_2, & d_4 &= 2a_4 + a_1 a_3, & d_6 &= a_3^2 + 4a_6, \\ d_8 &= a_1^2 a_6 + 4a_2 a_6 - a_1 a_3 a_4 + a_2 a_3^2 - a_4^2, & c_4 &= d_2^2 - 24d_4. \end{aligned}$$

Теорема 4.1. Эллиптическая кривая E невырождена тогда и только тогда, когда $\Delta(E) \neq 0$.

Будем рассматривать только невырожденные эллиптические кривые. Уравнение (4.1) является наиболее общим, однако в зависимости от характеристики поля, над которым задана кривая, оно может быть приведено к более простому виду путем линейной замены переменных. При изучении эллиптических кривых допускаются не всякие линейные замены, а лишь те, которые осуществляют изоморфизм эллиптических кривых как проективных многообразий. Определим изоморфизм эллиптических кривых следующим образом.

Определение 4.5. Две эллиптические кривые E и \tilde{E} над полем K , заданные уравнениями

$$\begin{aligned} E : y^2 + a_1 xy + a_3 y &= x^3 + a_2 x^2 + a_4 x + a_6, \\ \tilde{E} : y^2 + \tilde{a}_1 xy + \tilde{a}_3 y &= x^3 + \tilde{a}_2 x^2 + \tilde{a}_4 x + \tilde{a}_6, \end{aligned}$$

будем называть изоморфными над K , если существуют $u, r, s, t \in K$, $u \neq 0$, такие, что замена переменных

$$(x, y) \mapsto (u^2 x + r, u^3 y + u^2 s x + t) \tag{4.3}$$

переводит кривую E в \tilde{E} .

Эллиптические кривые, заданные над полем K , могут быть неизоморфны над этим полем, но могут стать изоморфными над расширением. Изоморфность эллиптических кривых также можно описать в терминах j -инварианта.

Теорема 4.2. Две эллиптические кривые E_1 и E_2 изоморфны над алгебраическим замыканием поля K тогда и только тогда, когда $j(E_1) = j(E_2)$.

Замена переменных (4.3) в уравнении (4.1) называется *допустимой*. Легко проверить, что это преобразование обратимо и обратное преобразование также допустимо. Кроме того, допустима тождественная замена переменных, поэтому всякая эллиптическая кривая изоморфна самой себе. Также несложно проверить, что композиция допустимых преобразований является допустимой.

Таким образом, изоморфность эллиптических кривых является отношением эквивалентности, и множество эллиптических кривых разбивается на классы эквивалентных, причем каждый класс эллиптических кривых, изоморфных над алгебраическим замыканием данного поля, однозначно определяется величиной j -инварианта.

4.3. СЛОЖЕНИЕ ТОЧЕК ЭЛЛИПТИЧЕСКОЙ КРИВОЙ

Существует ряд эквивалентных способов для описания группового закона сложения точек на эллиптической кривой. Геометрически этот закон можно сформулировать следующим образом: *три коллинеарные (лежащие на одной прямой) точки на эллиптической кривой дают нулевую сумму*. Известно, что операция сложения точек на эллиптической кривой, введенная по данному правилу, превращает эллиптическую кривую в коммутативную группу. Далее будут рассмотрены формулы сложения точек эллиптической кривой, которые вытекают из данного правила.

В качестве нулевого элемента выбирается точка \mathcal{O} на бесконечности. Таким образом, для любой точки $P \in E$ имеют место равенства $P + \mathcal{O} = \mathcal{O} + P = \mathcal{O}$.

Вертикальные прямые в плоскости xy проходят через точку \mathcal{O} . Точка \mathcal{O} задается в проективной плоскости координатами $(0 : 1 : 0)$; всякая прямая в проективной плоскости имеет вид $ax + by + cz = 0$; таким образом, так как \mathcal{O} лежит на прямой, то $b = 0$, т. е. прямая вертикальная. Пусть $P \neq \mathcal{O}$ – точка эллиптической кривой E и l – вертикальная прямая, проходящая через точку $P = (x_1, y_1)$. Эта прямая пересекает E еще и в точке Q (с учетом кратности). Таким образом, $-P = Q$. Пусть координаты точки Q равны (x_2, y_2) . Поскольку она лежит на l , то $x_2 = x_1$. Тогда y_2 является решением уравнения

$$y^2 + f_1(x_1)y - f_3(x_1) = 0, \quad (4.4)$$

где $f_1(x) = a_1x + a_3$, $f_3(x) = x^3 + a_2x^2 + a_4x + a_6$.

Уравнение (4.4) имеет два корня: y_1 и y_2 . По теореме Виета $y_1 + y_2 = -f_1(x_1)$, откуда получаем $y_2 = -y_1 - f_1(x_1)$. Итак,

$$-(x_1, y_1) = (x_1, -y_1 - a_1x_1 - a_3). \quad (4.5)$$

Может оказаться, что $y_2 = y_1$ и, соответственно, $-P = P$. Это возможно, если проходящая через P вертикальная прямая касается кривой E в точке P .

Рассмотрим, как вычисляется сумма двух различных точек $P_1 = (x_1, y_1)$ и $P_2 = (x_2, y_2)$ – на эллиптической кривой. Предположим, что $P_1, P_2 \neq \mathcal{O}$ и $P_1 \neq \pm P_2$. Тогда $x_1 \neq x_2$. Пусть l – прямая, проходящая через точки P_1 и P_2 . Уравнение l имеет вид

$$y = \alpha x + \beta. \quad (4.6)$$

Коэффициенты α и β – решения системы

$$\begin{cases} y_1 = \alpha x_1 + \beta, \\ y_2 = \alpha x_2 + \beta. \end{cases}$$

Тогда $\alpha = \frac{y_1 - y_2}{x_1 - x_2}$, $\beta = y_1 - \alpha x_1 = y_2 - \alpha x_2$.

Прямая l пересекает кривую E в некоторой третьей точке $Q = -(P_1 + P_2) = (\tilde{x}_3, \tilde{y}_3)$. Подставляя (4.6) в уравнение Вейерштрасса, получаем

$$(\alpha x + \beta)^2 + (\alpha x + \beta)f_1(x) - f_3(x) = 0.$$

Это кубическое уравнение имеет три корня: x_1, x_2, \tilde{x}_3 . По теореме Виета сумма $x_1 + x_2 + \tilde{x}_3$ равна $\alpha^2 + a_1\alpha - a_2$. Таким образом,

$$\begin{cases} \tilde{x}_3 = -x_1 - x_2 + \alpha^2 + a_1\alpha - a_2, \\ \tilde{y}_3 = \alpha \tilde{x}_3 + \beta = y_1 - \alpha(x_1 - \tilde{x}_3). \end{cases}$$

Используя (4.5), находим

$$P_1 + P_2 = (x_3, y_3), \text{ где } \begin{cases} x_3 = -x_1 - x_2 + \alpha^2 + a_1\alpha - a_2, \\ y_3 = -y_1 + \alpha(x_1 - x_3) - a_1x_3 - a_3. \end{cases} \quad (4.7)$$

Найдем правило вычисления удвоенной точки $P + P$. Будем предполагать, что $P \neq -P$. Чтобы найти $P + P$, проведем касательную через точку P . Эта прямая задается линейным уравнением (4.6), в котором коэффициент α равен взятому с обратным знаком отношению частных производных многочлена, задающего в неявном виде эллиптическую кривую:

$$\alpha = \frac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3}. \quad (4.8)$$

Таким образом, удвоенная точка также вычисляется по формулам (4.7), но с α , найденным по (4.8).

Теорема 4.3. Пусть K – поле, E/K – эллиптическая кривая. Тогда для любого расширения F поля K относительно введенной выше операции «+» множество $E(F)$ образует абелеву группу. Операция сложения задается по следующим правилам.

1. Для любой точки $P \in E$ верно $P + \mathcal{O} = \mathcal{O} + P = P$.

2. Для любой точки $P = (x, y) \neq 0$ точка $-P$ находится по формуле $-P = (x, -y - a_1x - a_3)$.

3. Для любых $P_1, P_2 \neq 0$ с координатами $P_i = (x_i, y_i)$, $i = 1, 2$, таких, что $P_1 \neq -P_2$, сумма этих точек равна точке $P_1 + P_2 = (x_3, y_3)$ такой, что

$$\left\{ \begin{array}{l} \alpha = \begin{cases} \frac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3}, & \text{если } x_1 = x_2, \\ \frac{y_1 - y_2}{x_1 - x_2}, & \text{если } x_1 \neq x_2, \end{cases} \\ x_3 = -x_1 - x_2 + \alpha^2 + a_1\alpha - a_2, \\ y_3 = -y_1 + \alpha(x_1 - x_3) - a_1x_3 - a_3. \end{array} \right. \quad (4.9)$$

Следствие 4.1. Для кривой, заданной уравнением

$$y^2 = x^3 + ax + b, \quad (4.10)$$

формулы (4.9) принимают упрощенный вид:

$$\left\{ \begin{array}{l} \alpha = \begin{cases} \frac{3x_1^2 + a}{2y_1}, & \text{если } x_1 = x_2, \\ \frac{y_1 - y_2}{x_1 - x_2}, & \text{если } x_1 \neq x_2, \end{cases} \\ x_3 = -x_1 - x_2 + \alpha^2, \\ y_3 = -y_1 + \alpha(x_1 - x_3). \end{array} \right. \quad (4.11)$$

Кроме того, противоположная к $P = (x, y)$ точка равна $(x, -y)$.

4.4. ЭЛЛИПТИЧЕСКИЕ КРИВЫЕ НАД КОНЕЧНЫМИ ПОЛЯМИ НЕЧЕТНОЙ ХАРАКТЕРИСТИКИ

Теорема 4.4. Всякая эллиптическая кривая над полем характеристики $\neq 2$ изоморфна некоторой эллиптической кривой вида

$$y^2 = x^3 + b_2x^2 + b_4x + b_6, \quad (4.12)$$

Доказательство. Пусть K – поле характеристики $\neq 2$, E – эллиптическая кривая, заданная уравнением (4.1). Тогда замена переменных

$$(x, y) \mapsto \left(x, y - \frac{a_1}{2}x - \frac{a_3}{2} \right)$$

переводит кривую E в кривую (4.12). \square

Теорема 4.5. Всякая эллиптическая кривая над полем характеристики > 3 изоморфна некоторой эллиптической кривой вида (4.10).

Доказательство. Пусть K – поле характеристики > 3 , E – эллиптическая кривая, заданная уравнением (4.1). Согласно теореме 4.4 данная эллиптическая кривая изоморфна кривой (4.12). Тогда замена переменных

$$(x, y) \mapsto (x - b_2/3, y)$$

переводит кривую (4.12) в (4.10). \square

В случае эллиптической кривой, заданной уравнением (4.10), формулы для дискриминанта и j -инварианта принимают вид

$$\begin{aligned} \Delta &= -16(4a^3 + 27b^2), \\ j(E) &= -1728 \frac{64a^3}{\Delta} = 1728 \frac{4a^3}{4a^3 + 27b^2}. \end{aligned} \quad (4.13)$$

4.5. ЭЛЛИПТИЧЕСКИЕ КРИВЫЕ НАД КОНЕЧНЫМИ ПОЛЯМИ ХАРАКТЕРИСТИКИ 2

Теорема 4.6. Всякая эллиптическая кривая над полем характеристики 2 изоморфна некоторой эллиптической кривой вида

$$y^2 + xy = x^3 + b_2x^2 + b_6 \quad (4.14)$$

и некоторой другой эллиптической кривой, имеющей вид

$$y^2 + b_3y = x^3 + b_4x + b_6. \quad (4.15)$$

Доказательство. Пусть K – поле характеристики 2, E – эллиптическая кривая, заданная уравнением (4.1). Если $a_1 \neq 0$, то замена переменных $(x, y) \mapsto (a_1^2x + a_1^{-1}a_3, a_1^3y)$ переводит E в кривую

$$y^2 + xy = x^3 + \tilde{a}_2x^2 + \tilde{a}_4x + \tilde{a}_6, \quad (4.16)$$

которая переводится в (4.14) заменой $(x, y) \mapsto (x, y + \tilde{a}_4)$.

Если $a_1 = 0$, то замена переменных $(x, y) \mapsto (x + a_2, y)$ переводит E в кривую вида (4.15). \square

4.6. МНОГОЧЛЕНЫ ДЕЛЕНИЯ

Как следует из формулы для нахождения суммы двух точек на кривой, координаты $P_1 + P_2$ являются рациональными функциями от координат P_1 и P_2 . Отсюда несложно получить, что координаты кратной точки $[m](x, y)$ представляются в виде рациональных функций от x и y . Конкретнее: имеет место результат [79, с. 39], заключенный в следующей лемме.

Лемма 4.1. Пусть E – эллиптическая кривая, определенная над полем K , и пусть $m \in \mathbb{N}$. Тогда существуют многочлены $\psi_m, \theta_m, \omega_m \in K[x, y]$ такие, что для любой точки $P = (x, y) \in E(\bar{K})$ такой, что $[m]P \neq \mathcal{O}$, верно равенство

$$[m]P = \left(\frac{\theta_m(x, y)}{\psi_m(x, y)^2}, \frac{\omega_m(x, y)}{\psi_m(x, y)^3} \right).$$

Многочлены $\theta_m, \omega_m, \psi_m$ задаются однозначно по модулю уравнения кривой.

Многочлен $\psi_m(x, y)$ называют m -м многочленом деления кривой E . Также несложно показать, что последовательности θ_m и ω_m могут быть выражены через ψ_m .

Пусть уравнение для эллиптической кривой E имеет вид $y^2 = x^3 + ax + b$. Тогда многочлены $\psi_m(x, y)$ можно найти рекуррентно из следующих равенств:

$$\begin{aligned} \psi_0 &= 0, & \psi_1 &= 1, & \psi_2 &= 2y; \\ \psi_3 &= 3x^4 + 6ax + 12bx - a^2; \\ \psi_4 &= 4y(x^6 + 5ax^4 + 20bx^3 - 5a^2x^2 - 4abx - 8b^2 - a^3); \\ \psi_{2m+1} &= \psi_{m+2}\psi_m^3 - \psi_{m-1}\psi_{m+1}^3, & m &\geq 2; \\ \psi_{2m} &= (\psi_{m+2}\psi_{m-1}^2 - \psi_{m-2}\psi_{m+1}^2)\psi_m/2y, & m &> 2. \end{aligned}$$

Поскольку возможно производить все действия с ψ_m по модулю уравнения кривой, то мы можем добиться, чтобы y в многочлены деления входил не более чем в первой степени. Кроме того, как несложно показать, ψ_{2m} делится на ψ_2 , и многочлены ψ_{2m+1} и ψ_{2m}/ψ_2 не будут зависеть от y . Обозначим такие многочлены через $f_m(x)$:

$$f_m(x) = \begin{cases} \psi_m(x), & \text{при } m \equiv 1 \pmod{2}, \\ \psi_m(x)/(2y), & \text{при } m \equiv 0 \pmod{2}. \end{cases}$$

Из рекурсивных формул для ψ_m несложно получить, что степень f_m не превосходит $(m^2 - 1)/2$ при нечетных m и не превосходит $(m^2 - 4)/2$ при четных. Кроме того, степень f_m будет совпадать с указанным значением в случае, если $\text{char}(K)$ не делит m при нечетном m и $m/2$ – при четном m . Многочлены $f_m(x)$ тесно связаны с $\psi_m(x, y)$ и иногда также называются многочленами деления. Именно эти многочлены деления будут в дальнейшем использоваться для вычислений.

Многочлены деления дают критерий принадлежности точки P на данной эллиптической кривой к m -кручению этой кривой.

Лемма 4.2. Пусть P – произвольная точка из множества $E(\bar{K}) \setminus \{\mathcal{O}\}$. В этом случае $P \in E[m]$ (то есть $[m]P = \mathcal{O}$) тогда и только тогда, когда $\psi_m(P) = 0$.

Если это утверждение переписать для многочленов f_m , то получим следующее утверждение.

Лемма 4.3. *Пусть $P = (x, y)$ – произвольная точка из множества $E(\overline{K}) \setminus \{O\}$ такая, что $[2]P \neq O$. В этом случае $P \in E[m]$ тогда и только тогда, когда $f_m(x) = 0$.*

Эта лемма дает один из возможных способов того, как алгоритмически сгенерировать ненулевую точку $P \in E[m]$. Для этого надо решить уравнение $f_m(x) = 0$, что дает первую координату точки, и затем найти вторую координату точки из уравнения кривой. Данный метод может быть реализован за полиномиальное время.

4.6.1. Вычисление многочленов деления

В алгоритмах вычисления порядка группы точек на эллиптической кривой (алгоритм Шуфа и его модификации) необходимо вычислить многочлены f_p , где p – небольшое простое число (для нас будет достаточно $p < 150$). Если записать рекурсивные формулы для вычисления Ψ_m в терминах f_m , то получим:

$$f_0 = 0, \quad f_1 = 1, \quad f_2 = 1, \quad f_3 = \Psi_3, \quad f_4 = \Psi_4/\Psi_2,$$

$$f_{2m+1} = \begin{cases} f_{m+2}f_m^3 - F^2 f_{m-1}f_{m+1}^3 & \text{при нечетном } m, \\ F^2 f_{m+2}f_m^3 - f_{m-1}f_{m+1}^3 & \text{при четном } m, \end{cases} \quad m \geq 3,$$

$$f_{2m} = (f_{m+2}f_{m-1}^2 - f_{m-2}f_{m+1}^2) f_m, \quad m > 2,$$

где $F = 4x^3 + 4ax + 4b$.

Поскольку операция нахождения f_m является довольно трудоемкой, произведем ее оптимизацию. В качестве оценки скорости работы алгоритма будем использовать количество вычисляемых произведений.

Во-первых, отметим, что для вычисления f_m не обязательно находить все f_i , $i = \overline{1, n}$. Нам достаточно знать только $f_{[m/2]}$, $f_{[m/2]-1}$, $f_{[m/2]+1}$, $f_{[m/2]+2}$ и при четном m еще $f_{[m/2]-2}$. Таким образом, если у нас есть некоторый набор $M \subset \mathbb{N}$ и требуется вычислить все f_m , $m \in M$, то несложно построить множество \overline{M} тех номеров m , для которых нам придется вычислить f_m , чтобы решить поставленную задачу. Поскольку алгоритм нахождения \overline{M} легко реализуется и работает очень быстро, то сначала можно построить это множество. Так, например, если в качестве M взять все простые числа от 2 до 150, то нам потребуется еще вычислить f_m для всех m от 1 до 76, кроме f_{60} и f_{72} .

Во-вторых, заметим, что $F_2(x)$ нам достаточно вычислить один раз в самом начале работы алгоритма.

Оценим трудоемкость операции нахождения f_m . Будем полагать, что для нахождения произведения двух многочленов степени a и b требуется ab операций умножения. Произведем операции следующим образом:

а) $f_{2m+1} = (f_{m+2}f_m)(f_m^2) - ((F^2 f_{m-1})f_{m+1})(f_{m+1}^2)$, если m нечетное. Поскольку нам известны степени многочленов f_m , то мы можем найти трудоемкость этой операции. Она равна

$$\frac{27}{4}m^4 + 6m^3 - \frac{93}{2}m^2 - 12m + \frac{183}{4};$$

б) $f_{2m+1} = ((F^2 f_{m+2})f_m)(f_m^2) - (f_{m-1}f_{m+1})(f_{m+1}^2)$, если m нечетное.

Трудоемкость:

$$\frac{27}{4}m^4 + 21m^3 - 24m^2 - 72m + 12;$$

в) $f_{2m} = (f_{m+2}(f_{m-1}^2) - f_{m-2}(f_{m+1}^2))f_m$.

Трудоемкость:

$$\frac{9}{4}m^4 - \frac{21}{2}m^2 - \frac{15}{4}, \text{ если } m \text{ нечетное, и}$$

$$\frac{9}{4}m^4 - 9m^2, \text{ если } m \text{ четное.}$$

Если обозначить через $K(n)$ трудоемкость операции нахождения всех f_m , где $m \leq 4n$, то получим

$$K(n) = \frac{18}{5}n^5 + \frac{81}{4}n^4 + \frac{15}{2}n^3 - \frac{291}{4}n^2 - \frac{123}{5}n + 66 = O(n^5).$$

Теперь заметим, что для вычисления f^2 , $\deg(f) = a$, можно обойтись $\frac{a(a+1)}{2}$ операциями умножения. Если это учесть, то получим

$$K(n) = \frac{17}{5}n^5 + 19n^4 + \frac{11}{2}n^3 - 73n^2 - \frac{329}{10}n + 78.$$

Далее, можно не вычислять f_{m-1}^2 и f_{m-1}^3 , а пользоваться уже вычисленными на предыдущих шагах. Это также ускорит работу алгоритма, тем не менее его сложность будет по-прежнему оцениваться как $O(n^5)$.

4.7. ИЗОГЕНИИ И ЭНДОМОРФИЗМ ФРОБЕНИУСА

Пусть E_1 и E_2 – две эллиптические кривые над полем K . Рациональным отображением из E_1 в E_2 называется отображение вида $f = (f_1 : f_2 : f_3)$, где $f_1, f_2, f_3 \in \overline{K}(E_1)$ такое, что для всех точек $P \in E_1$, в которых f регулярно, выполнено $(f_1(P) : f_2(P) : f_3(P)) \in E_2$. При этом говорят, что f регулярно в точке P , если существует функция $g \in \overline{K}(E_1)$ такая, что $gf_i(P)$ определены для всех i и для некоторого i $gf_i(P) \neq 0$.

Рациональное отображение из E_1 в E_2 , регулярное в каждой точке E_1 , называется *морфизмом*.

Известно, что всякий морфизм эллиптических кривых является либо постоянным, либо сюръективным [156, с. 24]. Заметим, что непостоянный морфизм может не быть сюръекцией между множествами K -рациональных точек эллиптических кривых. Здесь имеется в виду, что он отображает множество \overline{K} -рациональных точек на одной кривой на множество \overline{K} -рациональных точек на другой кривой.

Пусть E_1 и E_2 – две эллиптические кривые над полем K . Морфизм из E_1 в E_2 , который переводит \mathcal{O} в \mathcal{O} , называется *изогенией*. Если существует ненулевая изогения из E_1 в E_2 , то эти эллиптические кривые называются *изогенными*.

Рассмотрим некоторые свойства изогений, а также введем еще некоторые понятия, связанные с ними.

Предположим, что изогения g не постоянная, т. е. $g(E_1) \neq \{\mathcal{O}\}$. Тогда g порождает следующую инъекцию полей функций:

$$g^* : \begin{cases} \overline{K}(E_2) & \rightarrow \overline{K}(E_1), \\ f & \mapsto f \circ g. \end{cases}$$

Определим *степень изогении* следующим образом. Для постоянной изогении ее степень положим равной нулю, а в противном случае положим

$$\deg g = [\overline{K}(E_1) : g^*\overline{K}(E_2)].$$

Предложение. Для каждой непостоянной изогении существует и притом единственная двойственная ей изогенения $\hat{g} : E_2 \rightarrow E_1$, такая, что $\hat{g} \circ g$ совпадает с умножением на n в E_1 , где $n = \deg(g)$, и $g \circ \hat{g}$ совпадает с умножением на n в E_2 .

Из этого предложения сразу следует, что отношение изогенности является отношением эквивалентности.

Двойственность изогений обладает рядом свойств [156, с. 86]:

$$\begin{aligned} \hat{g} \circ g &= [n]_{E_1}, & g \circ \hat{g} &= [n]_{E_2}, & \deg \hat{g} &= \deg g, \\ \hat{g} &= g, & \widehat{g_1 + g_2} &= \hat{g}_1 + \hat{g}_2, & \widehat{g_1 \circ g_2} &= \hat{g}_2 \circ \hat{g}_1, \\ \widehat{[n]} &= [n], & \deg[n] &= n^2. \end{aligned}$$

Известно, что для эллиптических кривых каждая изогения является гомоморфизмом групп и что ядро непостоянной изогении из E – всегда конечная подгруппа в E , которую обозначают $E[\varphi]$. Пусть $P \in E[\varphi]$ и $\deg(\varphi) = n$. Тогда $[n]P = \hat{\varphi} \circ \varphi(P) = \mathcal{O}$. Таким образом, $E[\varphi] \subset E[n]$.

4.7.1. Эндоморфизмы и автоморфизмы эллиптической кривой

Изогении из кривой E в себя называются эндоморфизмами эллиптической кривой E . Множество эндоморфизмов образует кольцо относительно операций поточечного сложения и композиции. Обычно ее обозначают $\text{End}(E)$ и называют *кольцом эндоморфизмов кривой E* .

Подгруппа $\text{End}(E)$ обратимых по умножению эндоморфизмов состоит из автоморфизмов E в себя и называется группой автоморфизмов, обозначается $\text{Aut}(E)$.

Все отображения $[n]$ (т. е. операции скалярного умножения на n) являются эндоморфизмами эллиптической кривой. Поэтому $\mathbb{Z} \subset \text{End}(E)$.

Если $\text{End}(E) \neq \mathbb{Z}$, то говорят, что кривая E имеет *комплексное умножение*.

Известно, что группа $\text{End}(E)$ изоморфна либо группе \mathbb{Z} , либо порядку в некотором мнимом квадратичном расширении \mathbb{Q} , либо максимальному порядку в алгебре кватернионов над \mathbb{Q} .

Структура группы автоморфизмов также хорошо описана: если $\text{char}(K) \neq 2, 3$, то $\text{Aut}(E) = \begin{cases} \mu_2, & \text{если } j(E) \neq 0,1728, \\ \mu_4, & \text{если } j(E) = 1728, \\ \mu_6, & \text{если } j(E) = 0, \end{cases}$ где μ_n – группа корней n -й степени из единицы в поле \overline{K} .

4.8. ВЫЧИСЛЕНИЕ ПОРЯДКА ГРУППЫ ТОЧЕК ЭЛЛИПТИЧЕСКОЙ КРИВОЙ

Здесь речь пойдет о проблеме подсчета точек на эллиптической кривой E над конечным полем \mathbb{F}_q . Эта задача является важной для обеспечения работы ряда криптосистем цифровой подписи. Безопасность таких криптосистем основана на трудности вычисления дискретного логарифма в подгруппе группы точек некоторой эллиптической кривой. Сложность дискретного логарифмирования в группе точек эллиптической кривой оценивается квадратным корнем из наибольшего простого делителя порядка этой группы [79]. Поэтому в вопросах безопасности криптосистем на эллиптических кривых важно знать порядок группы, а также некоторые его делители.

4.8.1. Метод больших и малых шагов для вычисления порядка

Рассмотрим следующий вопрос: как определить, является ли данное число t порядком эллиптической кривой. Далее рассмотрим метод, позволяющий найти числа – «кандидаты» на то, чтобы быть порядком. Важно уметь выбрать среди них то единственное число, которое и является порядком группы точек эллиптической кривой.

Пусть $\bar{\mathbb{F}}_q$ – алгебраическое замыкание \mathbb{F}_q . Напомним, что для поля K , $\mathbb{F}_q \subset K \subset \bar{\mathbb{F}}_q$, $E(K)$ обозначает множество K -рациональных точек.

Если число $m = \#E(\mathbb{F}_q)$, то должны выполняться следующие условия:

$$\begin{cases} q + 1 - 2\sqrt{q} \leq m \leq q + 1 + 2\sqrt{q}, \\ P \in E(\mathbb{F}_q) \Rightarrow [m]P = \mathcal{O}. \end{cases} \quad (4.17)$$

Таким образом, чтобы отсеять лишних кандидатов из списка, в котором наверняка есть правильный ответ, можно проверять условия (4.17), перебирая разные точки P . Единственное оставшееся число в этом списке составит искомый порядок.

Рассмотрим алгоритм, позволяющий найти сравнительно небольшое количество кандидатов на порядок группы. По теореме Хассе

$$\#E(\mathbb{F}_q) = q + 1 - t, |t| \leq 2\sqrt{q}. \quad (4.18)$$

Чтобы установить порядок эллиптической кривой, требуется найти число t , для которого имеется $2[2\sqrt{q}] + 1$ возможностей.

Идея алгоритма состоит в нахождении для точки $P \in E(\mathbb{F}_q)$, выбранной случайным образом, всех чисел $m \in (q + 1 - 2\sqrt{q}, q + 1 + 2\sqrt{q})$ таких, что $[m]P = \mathcal{O}$. Если такое число в заданном интервале единственное, то по теореме Хассе $m = \#E(\mathbb{F}_q)$. Если таких чисел несколько, то можно уменьшить количество кандидатов из полученного списка на целый порядок методом, описанным выше.

Итак, точка $P \in E(\mathbb{F}_q)$ выбирается случайно. Положим $s = \lceil \sqrt[4]{q} \rceil$, где $[x] = \min\{a \in \mathbb{Z} : a \geq x\}$. Сначала составляется и сортируется таблица из $2s + 1$ точек:

$$\mathcal{O}, \pm P, \dots, \pm[s]P. \quad (4.19)$$

Далее, определим точки

$$Q = [2s + 1]P, \quad R = [q + 1]P. \quad (4.20)$$

Пусть $m = \#E(\mathbb{F}_q) = q + 1 - t$. Тогда

$$\mathcal{O} = [m]P = [q + 1 - t]P = R - [t]P. \quad (4.21)$$

Легко видеть, что существует единственное целое число j такое, что $|j| \leq s$ и $t \equiv j \pmod{2s + 1}$. Тогда число $i = (j - t)/(2s + 1)$ также целое, причем

$$|i| \leq \frac{s + 2\sqrt{q}}{2s + 1} \leq \frac{\sqrt[4]{q} + 1 + 2\sqrt{q}}{2\sqrt[4]{q} + 1} \leq \sqrt[4]{q} < \lceil \sqrt[4]{q} \rceil = s.$$

Это значит, что найдутся целые i, j такие, что $|i|, |j| \leq s$ и $-t = (2s + 1)i - j$. Подставляя это выражение в (4.21), получаем

$$\mathcal{O} = R + [i(2s + 1)]P - [j]P = R + [i]Q - [j]P. \quad (4.22)$$

Следовательно, вычисляя точки вида $R \pm [i]Q$, где $i = 0, \dots, s$, и проверяя, содежится ли такая точка в массиве (4.19), мы найдем все пары i, j такие, что выполнено (4.22). В частности, одной из этих пар будет соответствовать искомый порядок m .

4.8.2. Метод Шуфа

Этот метод основан на использовании свойств эндоморфизма Фробениуса:

$$\begin{aligned}\varphi_E : E(\overline{\mathbb{F}}_q) &\rightarrow E(\overline{\mathbb{F}}_q), \\ (x, y) &\mapsto (x^q, y^q), \\ \mathcal{O} &\mapsto \mathcal{O}.\end{aligned}$$

Согласно теореме Хассе [79] эндоморфизм Фробениуса удовлетворяет уравнению $f(\varphi_E) = 0$ в кольце эндоморфизмов $\text{End}(E)$, где $f(x) = x^2 - tx + q \in \mathbb{Z}[x]$, $t = q + 1 - \#E(\mathbb{F}_q)$, $t \leq 2\sqrt{q}$. Число t называется *следом Фробениуса*. Таким образом, для любой точки $P = (x, y) \in E$

$$\varphi_E^2(P) - [t]\varphi_E(P) + [q]P = \mathcal{O}. \quad (4.23)$$

Идея метода Шуфа сводится к определению остатков числа t по простым модулям l , $l \leq l_{\max}$, где l_{\max} – наименьшее простое такое, что

$$\prod_{2 \leq l \leq l_{\max}} l > 4\sqrt{q}.$$

Когда все такие остатки найдены, восстановить число t по китайской теореме об остатках не представляет особого труда.

При $l = 2$ легко определить $t \pmod{2}$. Если характеристика поля нечетная, $t \equiv \#E(\mathbb{F}_q) \pmod{2}$, то $\#E(\mathbb{F}_q) \equiv 1 \pmod{2} \Leftrightarrow x^3 + ax + b$ неразложим над \mathbb{F}_q , что эквивалентно $(x^3 + ax + b, x^q - x) = 1$. Если $\text{char}(\mathbb{F}_q) = 2$ и кривая не является вырожденной, то $t \equiv 1 \pmod{2}$.

Рассмотрим случай $l > 2$.

Множество $E[l] \subset E$, состоящее из точек эллиптической кривой, порядок которых делит l , т. е. из точек $P \in E(\overline{\mathbb{F}}_q)$ таких, что $[l]P = \mathcal{O}$, называется *l-кручением кривой* E .

Известно, что $E[l] \cong (\mathbb{Z}/l\mathbb{Z}) \times (\mathbb{Z}/l\mathbb{Z})$. Также легко проверяется, что $E[l]$ – подгруппа E .

Пусть $q_l = q \pmod{l}$. Для нахождения $t_l = t \pmod{l}$ выбирается точка $P = (x, y) \in E[l] \setminus \{\mathcal{O}\}$ и ищется $\tau \in 0, 1, \dots, l - 1$ такое, что

$$\varphi_E^2(P) + [q_l]P = [\tau]\varphi_E(P). \quad (4.24)$$

Такое τ единственное, так как l – простое, $P \neq 0$. Кроме того, в силу выбора точки P равенство (4.24) выполнено при $\tau = t_l$. Поэтому решение (4.24) существует и единственno. Найти его можно даже любым переборным методом (например, тем же самым методом больших и малых шагов). Для проверки соотношения (4.24) используются вышеописанные многочлены деления. Именно эти многочлены позволяют неявно оперировать с точками из подгруппы l -кручения, так как именно в них обращаются в 0 знаменатели l -го многочлена деления. Таким образом, проверка соотношения (4.24) осуществляется посредством вычисления левой и правой частей по модулю l -го многочлена деления.

4.9. ЗАДАНИЯ

1. Доказать, что всякая эллиптическая кривая над полем характеристики 3 изоморфна некоторой эллиптической кривой вида $y^2 = x^3 + ax^2 + b$.
2. Найти дискриминант и j -инвариант для кривых (4.14) и (4.15) над полем характеристики 2.
3. Найти дискриминант и j -инвариант для кривых (4.12) вида, определенного в задании 1, над полем характеристики 3.
4. Найти формулы сложения и обратной точки для кривых, заданных уравнениями (4.14) и (4.15), над полями характеристики 2.
5. Найти формулы сложения и обратной точки для кривых вида (4.12) и вида, определенного в задании 1, над полями характеристики 3.
6. Найти количество \mathbb{F}_{2^r} -рациональных точек на кривой $y^2 + y = x^3$, заданной над \mathbb{F}_2 , $r = 1, 2, 3, \dots$.
7. Доказать, что две эллиптические кривые $y^2 = x^3 + ax + b$ и $y^2 = x^3 + a'x + b'$ изоморфны над полем K тогда и только тогда, когда $a' = u^4a$, $b' = u^6b$ для некоторого $u \in K^*$.
8. Пусть E – эллиптическая кривая над \mathbb{F}_q , E' – скручивание эллиптической кривой E над \mathbb{F}_q . Доказать, что E и E' не изоморфны над \mathbb{F}_q , но изоморфны над \mathbb{F}_{q^2} .
9. Пусть E – эллиптическая кривая над \mathbb{F}_q , а E' и E'' – скручивания эллиптической кривой E над \mathbb{F}_q . Доказать, что E' и E'' изоморфны над \mathbb{F}_q .
10. Пусть P – точка эллиптической кривой E/\mathbb{F}_q и порядок этой точки $> 4\sqrt{q}$. Сколько различных целых чисел m удовлетворяет условиям (4.17)?
11. Пусть задана эллиптическая кривая E/\mathbb{F}_q и $E(\mathbb{F}_q)$ – циклическая группа. Оценить вероятность того, что для случайно взятой точки $P \in E$ найдется по крайней мере два различных целых числа m , удовлетворяющих условиям (4.17).

Г л а в а 5

ВЕРОЯТНОСТНЫЕ МОДЕЛИ СЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

5.1. ДИСКРЕТНЫЕ ВРЕМЕННЫЕ РЯДЫ, ИХ МОДЕЛИ И ВЕРОЯТНОСТНЫЕ ХАРАКТЕРИСТИКИ

При функционировании любой криптографической системы в ее узлах и на выходе формируются хаотические последовательности символов из некоторого дискретного множества. Удобной математической моделью для их описания является *дискретная случайная последовательность*, или *дискретный временной ряд* (*ДВР*).

Определение 5.1. *Дискретный временной ряд x_t есть упорядоченная по индексу $t \in \mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$ последовательность дискретных случайных величин:*

$$x_t = x(\omega, t) \in \mathcal{A}, \omega \in \Omega, t \in \mathbb{Z}, \quad (5.1)$$

определенных на одном и том же вероятностном пространстве $(\Omega, \mathcal{F}, \mathbf{P})$ и принимающих значения из некоторого дискретного множества \mathcal{A} , где Ω – пространство элементарных событий; \mathcal{F} – σ -алгебра подмножеств из Ω , называемых случайными событиями; $\mathbf{P}(\cdot)$ – вероятностная мера, определенная на \mathcal{F} .

В (5.1) индекс $t \in \mathbb{Z}$ интерпретируется как дискретное время, а \mathcal{A} называется *пространством состояний* (алфавитом) временного ряда. В существующих криптографических системах алфавит \mathcal{A} конечен, поэтому далее без потери общности условимся полагать

$$\mathcal{A} = \{0, 1, \dots, N - 1\},$$

где $2 \leq N < +\infty$ – мощность алфавита \mathcal{A} , т. е. число различных возможных значений дискретного временного ряда x_t . Если $N = 2$, то $x_t \in \mathcal{A} = \{0, 1\}$ принято называть *двоичным* (бинарным) временным рядом.

Определение 5.2. *Отсчетом ДВР в некоторый фиксированный момент времени $t = s \in \mathbb{Z}$ называется дискретная случайная величина $x_s \in \mathcal{A}$. Реализация (траектория) ДВР – это упорядоченная совокупность всех отсчетов при фиксированном элементарном событии (исходе случайного эксперимента) $\omega \in \Omega$:*

$$X_\omega = \{x(\omega, t) : t \in \mathbb{Z}\}.$$

Совокупность всех реализаций $X = \{X_\omega : \omega \in \Omega\}$ называется ансамблем реализаций ДВР.

Вероятностные модели ДВР задаются на основе их вероятностных характеристик. Определим основные вероятностные характеристики ДВР.

Определение 5.3. Пусть зафиксированы произвольные натуральное число $n \in \mathbb{N}$ и упорядоченные моменты времени $t_1 < t_2 < \dots < t_n$ ($t_1, \dots, t_n \in \mathbb{Z}$). Совместное дискретное распределение вероятностей отсчетов $x_{t_1}, \dots, x_{t_n} \in \mathcal{A}$

$$P_n(a_1, \dots, a_n; t_1, \dots, t_n) = \mathbf{P}\{x_{t_1} = a_1, \dots, x_{t_n} = a_n\}, \quad a_1, \dots, a_n \in \mathcal{A}, \quad (5.2)$$

называется n -мерным распределением вероятностей временного ряда x_t .

Функция (5.2) обладает следующими свойствами:

- область значений $P_n(\cdot)$ есть $[0, 1]$;
- свойство нормировки:

$$\sum_{a_1, \dots, a_n \in \mathcal{A}} P_n(a_1, \dots, a_n; t_1, \dots, t_n) \equiv 1;$$

- свойство согласованности ($1 \leq k < n$):

$$\sum_{a_{k+1}, \dots, a_n \in \mathcal{A}} P_n(a_1, \dots, a_n; t_1, \dots, t_n) \equiv P_k(a_1, \dots, a_k; t_1, \dots, t_k).$$

Заметим, что в последнем соотношении k -мерное распределение вероятностей $P_k(\cdot)$ называется *маргинальным распределением вероятностей* по отношению к исходному $P_n(\cdot)$.

Известно [59], что семейство всевозможных конечномерных распределений вероятностей (5.2) однозначно задает *вероятностную модель ДВР*. Отметим еще, что в силу конечности N ДВР x_t имеет ограниченные моменты любого порядка $s > 0$: $\mathbf{E}\{x_t^s\} < +\infty$.

Определение 5.4. Математическим ожиданием (средним) и дисперсией ДВР x_t называются функции времени

$$m_t = \mathbf{E}\{x_t\} = \sum_{a \in \mathcal{A}} a P_1(a; t), \quad t \in \mathbb{Z},$$

$$d_t = \mathbf{D}\{x_t\} = \mathbf{E}\{(x_t - m_t)^2\} = \sum_{a \in \mathcal{A}} (a - m_t)^2 P_1(a; t) \geq 0, \quad t \in \mathbb{Z}, \quad (5.3)$$

соответственно.

Определение 5.5. Ковариационная и корреляционная функция ДВР x_t являются функциями двух переменных

$$\begin{aligned} \sigma(t_1, t_2) &= \mathbf{Cov}\{x_{t_1}, x_{t_2}\} = \mathbf{E}\{(x_{t_1} - m_{t_1})(x_{t_2} - m_{t_2})\} = \\ &= \sum_{a_1, a_2 \in \mathcal{A}} (a_1 - m_{t_1})(a_2 - m_{t_2}) P_2(a_1, a_2; t_1, t_2), \quad t_1, t_2 \in \mathbb{Z}, \end{aligned}$$

$$\rho(t_1, t_2) = \mathbf{Corr}\{x_{t_1}, x_{t_2}\} = \frac{\sigma(t_1, t_2)}{\sqrt{d_{t_1} d_{t_2}}} \in [-1, +1], \quad t_1, t_2 \in \mathbb{Z}, \quad (5.4)$$

соответственно.

Отметим, что $\sigma(t, t) = d_t$, $t \in \mathbb{Z}$.

Определение 5.6. Дискретный временной ряд x_t называется стационарным в узком смысле (сильно стационарным), если любое его n -мерное распределение вероятностей (5.2) инвариантно относительно сдвига времени, т. е. для любого $n \in \mathbb{N}$, любых $t_1, \dots, t_n \in \mathbb{Z}$ ($t_1 < t_2 < \dots < t_n$) и любого сдвига времени $\tau \in \mathbb{Z}$ выполняется соотношение

$$\begin{aligned} P_n(a_1, \dots, a_n; t_1 + \tau, \dots, t_n + \tau) = \\ = P_n(a_1, \dots, a_n; t_1, \dots, t_n), \quad a_1, \dots, a_n \in \mathcal{A}. \end{aligned} \quad (5.5)$$

Соотношение (5.5) означает, что для стационарного в узком смысле дискретного временного ряда n -мерное распределение вероятностей $P_n(\cdot; t_1, \dots, t_n)$ зависит лишь от взаиморасположения моментов времени t_1, \dots, t_n (т. е. от разностей $t_2 - t_1, t_3 - t_2, \dots, t_n - t_{n-1}$). В частности, из (5.5) следует, что одномерное распределение (при $n = 1$) вообще не зависит от времени:

$$P_1(a; t) \equiv P_1(a), \quad (5.6)$$

а двумерное распределение (при $n = 2$) зависит лишь от $t_2 - t_1$:

$$P_2(a_1, a_2; t_1, t_2) \equiv P_2(a_1, a_2; t_2 - t_1). \quad (5.7)$$

Определение 5.7. Дискретный временной ряд x_t называется стационарным в широком смысле (слабо стационарным), если его математическое ожидание не зависит от времени t :

$$m_t \equiv m, \quad m \in \mathbb{R}, \quad (5.8)$$

а его ковариационная функция зависит лишь от разности моментов времени $t_2 - t_1 \in \mathbb{Z}$:

$$\sigma(t_1, t_2) \equiv \sigma(t_2 - t_1), \quad (5.9)$$

где $\sigma(u)$, $u \in \mathbb{Z}$, – некоторая четная неотрицательно определенная функция.

Теорема 5.1. Если N конечно, то всякий стационарный в узком смысле ДВР x_t является одновременно и стационарным в широком смысле дискретным временным рядом.

Доказательство. Как отмечалось выше, в силу конечности N для ДВР x_t существуют ограниченные моменты любого порядка. Следовательно, существуют моменты первого и второго порядка (5.3), (5.4). Поэтому из (5.6) вытекает (5.8), а из (5.7) следует (5.9). \square

Отметим, что, как видно из теоремы 5.1, из стационарности в узком смысле следует стационарность в широком смысле; обратное, вообще говоря, не верно.

Как известно [52], ковариационная функция $\sigma(u)$ стационарного в широком смысле ДВР x_t характеризует уровень линейной стохастической зависимости отсчетов x_t и x_{t+u} , отстоящих на u единиц времени. На практике временные ряды имеют «затухающую память» [26]:

$$P_2(a_1, a_2; t_1, t_2) - P_1(a_1; t_1)P_1(a_2; t_2) \rightarrow 0, \quad a_1, a_2 \in \mathcal{A},$$

при $|t_2 - t_1| \rightarrow +\infty$, влекущую в силу (5.4), (5.7) убывание ковариационной функции стационарного ДВР:

$$\sigma(u) \rightarrow 0, |u| \rightarrow +\infty. \quad (5.10)$$

Определение 5.8. Пусть (5.10) выполняется в усиленном смысле, так что сходится ряд

$$\sum_{u=-\infty}^{+\infty} |\sigma(u)| < +\infty. \quad (5.11)$$

Спектральной плотностью $S(\lambda)$, $\lambda \in [-\pi, \pi]$, стационарного в широком смысле ДВР x_t , имеющего ковариационную функцию $\sigma(u)$, $u \in \mathbb{Z}$, называется *дискретное преобразование Фурье* ($\mathcal{D}\Pi\Phi$) от ковариационной функции

$$S(\lambda) = \frac{1}{2\pi} \sum_{u=-\infty}^{+\infty} \sigma(u) e^{i\lambda u} \equiv \frac{1}{2\pi} \sum_{u=-\infty}^{+\infty} \sigma(u) \cos(\lambda u), \quad (5.12)$$

где i – мнимая единица.

Отметим некоторые свойства спектральной плотности $S(\cdot)$:

- неотрицательность: $S(\lambda) \geq 0$, $\lambda \in [-\pi, \pi]$;
- четность $S(-\lambda) = S(\lambda)$, $\lambda \in [-\pi, \pi]$;
- $S(\cdot)$ и $\sigma(\cdot)$ взаимнооднозначно связаны парой $\mathcal{D}\Pi\Phi$ (5.12), и

$$\sigma(u) = \int_{-\pi}^{\pi} S(\lambda) e^{i\lambda u} d\lambda \equiv \int_{-\pi}^{\pi} S(\lambda) \cos(\lambda u) d\lambda, \quad u \in \mathbb{Z}.$$

Как уже отмечалось, вероятностная модель ДВР полностью определяется заданием системы конечномерных распределений вероятностей

$$\mathcal{P} = \{P_n(a_1, \dots, a_n; t_1, \dots, t_n) : a_1, \dots, a_n \in \mathcal{A}; t_1, \dots, t_n \in \mathbb{Z}\}. \quad (5.13)$$

Система распределений вероятностей \mathcal{P} , определяемая (5.13), бесконечна, поэтому для ее задания необходимы некоторые конструктивные функциональные соотношения. Эти функциональные соотношения для \mathcal{P} определяют конкретный вид вероятностной модели ДВР и излагаются в последующих пунктах данной главы.

5.2. РАВНОМЕРНО РАСПРЕДЕЛЕННАЯ СЛУЧАЙНАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ И ЕЕ СВОЙСТВА

Простейшей вероятностной моделью ДВР является *равномерно распределенная случайная последовательность (PPCP)*, которая в криптологии иногда называется «чисто случайная последовательность».

Определение 5.9. *PPCP – это случайная последовательность $x_1, x_2, \dots, x_t, x_{t+1}, \dots$ со значениями в дискретном множестве $\mathcal{A} = \{0, 1, \dots, N - 1\}$, определенная на вероятностном пространстве $(\Omega, \mathcal{F}, \mathbf{P})$ и удовлетворяющая двум свойствам – C₁ и C₂.*

Свойство C₁. Для любого $n \in \mathbb{N}$ и произвольных значений индексов $t_1, \dots, t_n \in \mathbb{Z}$, $t_1 < \dots < t_n$, случайные величины $x_{t_1}, \dots, x_{t_n} \in \mathcal{A}$ независимы в совокупности

$$\mathbf{P}\{x_{t_1} = i_1, \dots, x_{t_n} = i_n\} = \prod_{k=1}^n \mathbf{P}\{x_{t_k} = i_k\}, \quad i_1, \dots, i_n \in \mathcal{A}.$$

Свойство C₂. Для любого номера $t \in \mathbb{N}$ случайная величина x_t имеет дискретное равномерное на \mathcal{A} распределение вероятностей:

$$\mathbf{P}\{x_t = i\} = \frac{1}{N}, \quad i \in \mathcal{A}.$$

Из базовых свойств C₁, C₂ и определений легко доказываются следующие дополнительные свойства.

Свойство C₃. Если $\{x_t\}$ – PPCP, то для любого $n \in \mathbb{N}$ и любой фиксированной последовательности индексов $t_1, \dots, t_n \in \mathbb{Z}$, $t_1 < \dots < t_n$, n -мерное дискретное распределение вероятностей случайного вектора $(x_{t_1}, x_{t_2}, \dots, x_{t_n}) \in \mathcal{A}^n$ является равномерным:

$$P_n(i_1, \dots, i_n; t_1, \dots, t_n) = \mathbf{P}\{x_{t_1} = i_1, \dots, x_{t_n} = i_n\} = \frac{1}{N^n}, \quad i_1, \dots, i_n \in \mathcal{A}.$$

Свойство C₄. Если $x_t \in \mathcal{A}$ – элемент PPCP, то $\forall k \in \mathbb{N}$ справедливы следующие выражения его начального и центрального моментов k -го порядка («моменты PPCP»):

$$\alpha_k = \mathbf{E}\{x_t^k\} = \frac{1}{N(k+1)} \sum_{l=0}^k \binom{k+1}{l} B_l N^{k+1-l};$$

$$\mu_k = \mathbf{E}\{(x_t - \alpha_1)^k\} = k! \sum_{l=0}^k \frac{B_l}{l!} \sum_{s=1}^{k+1-l} \frac{N^{s-1}}{s!} \times \frac{(-(N-1)/2)^{k+1-l-s}}{(k+1-l-s)!},$$

где $\{B_l\}$ – числа Бернулли [23]. В частности, математическое ожидание

$$m_t = \mathbf{E}\{x_t\} = \frac{N-1}{2}, \quad \text{а дисперсия} \quad d_t = \mathbf{D}\{x_t\} = \frac{N^2-1}{12}.$$

Свойство С₅. Для ковариационной функции и спектральной плотности РРСП { x_t } справедливы следующие выражения:

$$r(\tau) = \mathbf{E} \{(x_t - \alpha_1)(x_{t+\tau} - \alpha_1)\} = \frac{N^2 - 1}{12} \delta_{\tau,0}, \quad \tau \in \mathbb{Z};$$

$$S(\lambda) = \frac{1}{2\pi} \sum_{\tau=-\infty}^{+\infty} r(\tau) \cos(\lambda\tau) = \frac{N^2 - 1}{24\pi}, \quad \lambda \in [-\pi, +\pi],$$

где $\delta_{i,j}$ – символ Кронекера.

Свойство С₆ (воспроизведимость при прореживании). Для любой фиксированной последовательности моментов времени $t_1, \dots, t_n \in \mathbb{Z}$, $t_1 < \dots < t_n < t_{n+1} < \dots$, при «прореживании» РРСП { x_t } возникает подпоследовательность

$$y_1 = x_{t_1}, \dots, y_n = x_{t_n}, y_{n+1} = x_{t_{n+1}}, \dots,$$

которая также является РРСП.

Свойство С₇ (воспроизведимость при суммировании). Если $x_t \in \mathcal{A}$ – РРСП, а $\xi_t \in \mathcal{A}$ – произвольная неслучайная либо случайная последовательность, не зависящая от { x_t }, то случайная последовательность

$$y_t = (x_t + \xi_t) \bmod N$$

также является РРСП.

Свойство С₈. Если { x_t } – РРСП, то $\forall n \in \mathbb{N}$ количество информации по Шеннону, содержащейся в отрезке последовательности $X_n = (x_1, \dots, x_n) \in \mathcal{A}^n$, о будущем элементе x_{n+1} равно нулю:

$$\mathbf{I}\{x_{n+1}, X_n\} = 0,$$

поэтому для любого алгоритма прогнозирования $\hat{x}_{n+1} = f(X_n): \mathcal{A}^n \rightarrow \mathcal{A}$ вероятность ошибки прогнозирования не может быть меньше, чем для «угадывания по жребию»:

$$\mathbf{P}\{\hat{x}_{n+1} \neq x_{n+1}\} \geq 1 - \frac{1}{N}.$$

Свойство С₉. Если { x_t } – РРСП, то для любого $k \in \mathbb{N}$ и произвольной интегрируемой борелевской функции k переменных $y = f(z_1, \dots, z_k)$, $z_1, \dots, z_k \in \mathbb{R}$, при $n \rightarrow \infty$ имеет место сходимость почти наверное:

$$\frac{1}{n} \sum_{\tau=1}^n f(x_{(\tau-1)k+1}, \dots, x_{\tau k}) \xrightarrow{\text{п.н.}} \frac{1}{N^k} \sum_{i_1, \dots, i_k \in \mathcal{A}} f(t_{i_1}, \dots, i_{i_k}).$$

Свойство С₁₀. Если { x_t } – равномерно распределенная последовательность порядка $k = \infty$ в смысле Г. Вейля [18, 49], то { x_t } – РРСП.

5.3. ЦЕПЬ МАРКОВА И ЕЕ СВОЙСТВА

В криптологии широкое применение получил класс ДВР, обладающих марковскими свойствами: цепи Маркова с дискретным временем.

Определение 5.10. *ДВР $x_t \in \mathcal{A}$, $t \in \mathbb{Z}$, определенный на вероятностном пространстве $(\Omega, \mathcal{F}, \mathbf{P})$, называется цепью Маркова (ЦМ) с пространством состояний \mathcal{A} , если для любого $n \in \mathbb{N}$ и любых $t_1, \dots, t_{n+1} \in \mathbb{Z}$ таких, что $t_1 < t_2 < \dots < t_{n+1}$, выполняется марковское свойство*

$$\mathbf{P} \left\{ x_{t_{n+1}} = i_{n+1} | x_{t_n} = i_n, \dots, x_{t_1} = i_1 \right\} = \mathbf{P} \left\{ x_{t_{n+1}} = i_{n+1} | x_{t_n} = i_n \right\},$$

$$i_1, \dots, i_{n+1} \in \mathcal{A}. \quad (5.14)$$

Соотношение (5.14) означает, что условное распределение вероятностей состояний $x_{t_{n+1}} \in \mathcal{A}$ в будущий момент времени t_{n+1} при условии, что известна некоторая предыстория состояний процесса $\{x_{t_n} = i_n, \dots, x_{t_1} = i_1\}$ в предыдущие моменты времени, зависит на самом деле не от всей этой предыстории, а лишь от состояния процесса $\{x_{t_n} = i_n\}$ в самый близкий к t_{n+1} прошлый момент времени t_n .

Пусть $t = 0$ – начальный момент времени; $p(0) = (p_0(0), \dots, p_{N-1}(0))'$ $\in \mathbb{R}^N$ – вектор-столбец начального распределения вероятностей состояний цепи Маркова x_t :

$$p_i(0) = P_1(i; 0) = \mathbf{P} \{x_0 = i\}, \quad i \in \mathcal{A}; \quad \sum_{i \in \mathcal{A}} p_i(0) = 1; \quad (5.15)$$

$P(t) = (p_{ij}(t)) \in \mathbb{R}^{N \times N}$ – матрица вероятностей одношаговых переходов:

$$p_{ij}(t) = \mathbf{P} \{x_{t+1} = j | x_t = i\}, \quad i, j \in \mathcal{A}, \quad t \in \mathbb{N}_0 = \mathbb{N} \cup \{0\}. \quad (5.16)$$

Матрица $P(t)$ относится к классу стохастических матриц, так как обладает свойствами:

$$p_{ij}(t) \in [0, 1], \quad i, j \in \mathcal{A}; \quad \sum_{j \in \mathcal{A}} p_{ij}(t) = 1, \quad i \in \mathcal{A}.$$

Теорема 5.2. Для любого $n \in \mathbb{N}_0$ $(n + 1)$ -мерное распределение вероятностей цепи Маркова x_t , $t \geq 0$, однозначно выражается через начальное распределение вероятностей $p(0)$ и матрицу вероятностей одношаговых переходов $P(0)$:

$$P_{n+1}(i_0, i_1, \dots, i_n; 0, 1, \dots, n) = \mathbf{P} \{x_0 = i_0, x_1 = i_1, \dots, x_n = i_n\} =$$

$$= p_{i_0}(0) \prod_{t=0}^{n-1} p_{i_t, i_{t+1}}(t). \quad (5.17)$$

Доказательство. Воспользуемся (5.2) и обобщенной формулой умножения вероятностей:

$$\begin{aligned} P_{n+1}(i_0, \dots, i_n; 0, \dots, n) &= \\ &= \mathbf{P}\{x_0 = i_0\} \cdot \prod_{t=0}^{n-1} \mathbf{P}\{x_{t+1} = i_{t+1} | x_t = i_t, \dots, x_0 = i_0\}. \end{aligned}$$

Применяя к правой части марковское свойство (5.14) и обозначения (5.15), (5.16), приходим к (5.17). \square

Заметим, что, используя (5.17) и свойство согласованности из п. 5.1, легко получить t -мерные распределения для любой цепочки моментов времени $0 \leq t_1 < t_2 < \dots < t_m$.

Определение 5.11. Если матрица вероятностей одностадийных переходов $P(t)$, задаваемая (5.16), не зависит от времени $t \in \mathbb{Z}$,

$$P(t) \equiv P = (p_{ij}) \in \mathbb{R}^{N \times N},$$

где P – некоторая стохастическая матрица, то x_t называется однородной цепью Маркова (ОЦМ), в противном случае – неоднородной.

Обозначим по аналогии с (5.15), (5.16):

$$p(t) = (p_0(t), \dots, p_{N-1}(t))', p_i(t) = P_1(i; t) = \mathbf{P}\{x_t = i\}, i \in \mathcal{A}, -$$

распределение вероятностей состояний ОЦМ в момент времени $t \in \mathbb{Z}$;

$$P^{(s)} = \left(p_{ij}^{(s)} \right), p_{ij}^{(s)} = \mathbf{P}\{x_{t+s} = j | x_t = i\}, i, j \in \mathcal{A}, -$$

матрица вероятностей s -шаговых переходов ($s \in \mathbb{N}$).

Следствие 5.1. Для ОЦМ x_t справедливы следующие формулы:

$$\begin{aligned} P_{n+1}(i_0, i_1, \dots, i_n; 0, 1, \dots, n) &= p_{i_0}(0) \prod_{t=0}^{n-1} p_{i_t, i_{t+1}}; \\ P^{(s+m)} &= P^{(s)} \cdot P^{(m)}, s, m \in \mathbb{N} \text{ (формула Колмогорова – Чепмена);} \\ P^{(t)} &= P^t, p(t) = (P^t)'p(0). \end{aligned}$$

Введем ряд необходимых понятий по классификации состояний ОЦМ [52].

Определение 5.12. Состояние $i \in \mathcal{A}$ называется несущественным, если найдется состояние $j \neq i$ и $s \in \mathbb{N}$ такие, что $p_{ij}^{(s)} > 0$, но $p_{ji}^{(t)} = 0$ для любого $t \in \mathbb{N}$; в противном случае состояние называется существенным.

Определение 5.13. Состояние $j \in \mathcal{A}$ называется достижимым из состояния $i \in \mathcal{A}$, если существует $s \in \mathbb{N}_0$ такое, что $p_{ij}^{(s)} > 0$ (обозначается $i \rightarrow j$); если $i \rightarrow j$, а $j \rightarrow i$, то состояния i, j называются сообщающимися (обозначаются $i \leftrightarrow j$). Бинарное отношение $i \leftrightarrow j$ разбивает множество всех существенных состояний на непересекающиеся неразложимые классы сообщающихся состояний: $S_1, \dots, S_L \subset \mathcal{A}$; если некоторый класс S_l состоит из единственного состояния $a^* \in \mathcal{A}$, то это состояние называется поглощающим (при попадании в него ОЦМ навсегда остается в этом состоянии).

Определение 5.14. ОЦМ, множество состояний которой \mathcal{A} образует один класс существенных сообщающихся состояний, называется неразложимой.

Определение 5.15. Пусть $d_i = \text{НОД} \left\{ n \in \mathbb{N} : p_{ii}^{(n)} > 0 \right\}$. Если $d_i > 1$, то состояние $i \in \mathcal{A}$ называется периодическим с периодом d_i ; если $d_i = 1$, то состояние $i \in \mathcal{A}$ называется непериодическим.

Обозначим:

$$q_i(n) = \mathbf{P} \{x_n = i, x_{n-1} \neq i, \dots, x_1 \neq i | x_0 = i\}, \quad i \in \mathcal{A}; \quad P_i = \sum_{m=1}^{+\infty} q_i(m),$$

где $q_i(n)$ – вероятность события, состоящая в том, что ОЦМ x_t , выйдя из начального состояния i , впервые вернется в него на n -м шаге; P_i – вероятность того, что ОЦМ, выйдя из состояния i , вновь вернется в него когда-нибудь.

Определение 5.16. Состояние $i \in \mathcal{A}$ называется возвратным, если $P_i = 1$, в противном случае – невозвратным.

Теорема 5.3 (критерий возвратности). Состояние $i \in \mathcal{A}$ возвратно тогда и только тогда, когда расходится ряд

$$Q_i = \sum_{n=1}^{+\infty} p_{ii}^{(n)} = +\infty.$$

Для невозвратного состояния i вероятность возврата $P_i = Q_i / (1 + Q_i)$.

Доказательство. Достаточно воспользоваться формулой полной вероятности и построить производящие функции для $\{p_{ii}^{(n)} : n \in \mathbb{N}\}$ и $\{p_i(n) : n \in \mathbb{N}\}$ [52]. \square

Определение 5.17. Обозначим $\mu_i = \sum_{n=1}^{+\infty} np_i(n) \geq 1$ – среднее время возвращения в i -е состояние ОЦМ, начавшей свое движение из i -го начального состояния. Состояние $i \in \mathcal{A}$ называется положительным, если $\mu_i^{-1} > 0$ (т. е. среднее время возвращения конечно: $\mu_i < +\infty$) и нулевым, если $\mu_i^{-1} = 0$ (т. е. $\mu_i = +\infty$).

Теорема 5.4 (теорема солидарности). В неприводимой ОЦМ все состояния солидарно обладают одним и тем же свойством: если хотя бы одно состояние возвратно, то и все возвратны; если хотя бы одно периодично с некоторым периодом d , то и все периодичны с периодом d ; если хотя бы одно состояние непериодично (т. е. $d = 1$), то и все состояния непериодичны (при этом ОЦМ называется непериодической).

Теорема 5.5 (теорема о циклических подклассах). Если ОЦМ x_t – неприводимая и периодическая с некоторым периодом $d > 1$, то множество \mathcal{A} ее состояний разбивается на d циклических подклассов: D_0, D_1, \dots, D_{d-1} таких, что с вероятностью единица за один шаг ОЦМ переходит из некоторого класса D_k ($k < d - 1$) в класс D_{k+1} , а из класса D_{d-1} – в D_0 .

Доказательство теорем 5.4, 5.5 приведено в [52].

Подводя итог представленной выше классификации состояний цепи Маркова, представим эту классификацию графически на рис. 5.1, 5.2 [59].

Определение 5.18. ОЦМ $x_t, t \in \mathbb{N}_0$, называется эргодической, если для любых $i, j \in \mathcal{A}$ существуют независимые от i положительные пределы

$$\lim_{n \rightarrow +\infty} p_{ij}^{(n)} = \pi_j^* > 0;$$

при этом вектор-столбец $\pi^* = (\pi_0^*, \pi_1^*, \dots, \pi_{N-1}^*)'$ называется стационарным распределением вероятностей ОЦМ.

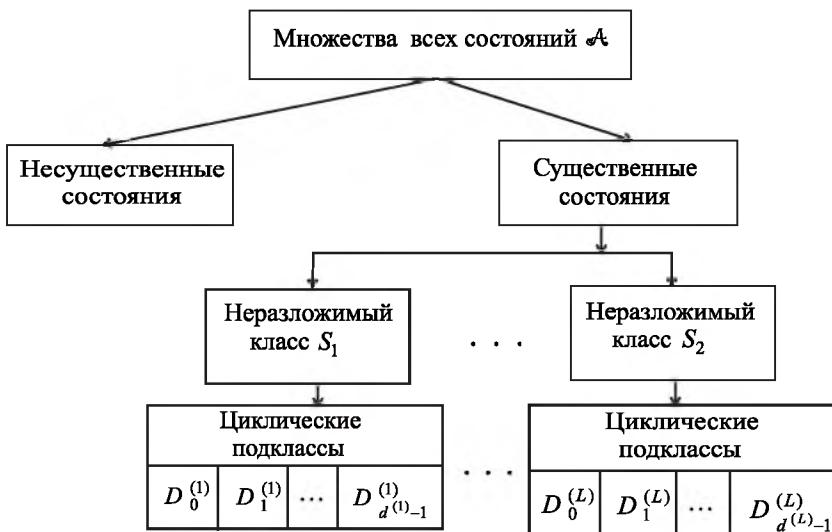


Рис. 5.1. Классификация состояний ОЦМ по арифметическим свойствам вероятностей $\{p_{ij}^{(n)}\}$

Стационарное распределение вероятностей ОЦМ x_t является единственным решением системы линейных алгебраических уравнений

$$\begin{cases} P'\pi = \pi, \\ \sum_{i=0}^{N-1} \pi_i = 1. \end{cases} \quad (5.18)$$



Рис. 5.2. Классификация состояний ОЦМ по асимптотическим свойствам вероятностей $\{p_{ii}^{(n)}\}$

Теорема 5.6 (о критерии эргодичности [59]). Свойство эргодичности ОЦМ x_t эквивалентно каждому из следующих утверждений:

- 1) ОЦМ x_t неразложима и непериодическая;
- 2) $\forall i, j \in \mathcal{A}$ существуют положительные пределы

$$\lim_{n \rightarrow +\infty} p_{ij}^{(n)} = \frac{1}{\mu_j} > 0,$$

где μ_j задано в определении 5.17;

3) найдется такое достаточно большое число шагов n_0 , что для всех $n \geq n_0$ все элементы матрицы P^n положительны:

$$\min_{i, j \in \mathcal{A}} p_{ij}^{(n)} > 0.$$

Теорема 5.7. Пусть для ОЦМ x_t , $t \in \mathbb{N}_0$, выполнено условие эргодичности и $\pi^* = (\pi_i^*)$ – стационарное распределение вероятностей, удовлетворяющее (5.18). Тогда, если начальное распределение вероятностей совпадает со стационарным:

$$p(0) = \pi^*, \quad (5.19)$$

то справедливы следующие два результата:

- 1) распределение вероятностей $p(t) \equiv \pi^*$ и не зависит от времени t ;
- 2) ОЦМ x_t является стационарным в узком смысле ДВР.

Доказательство. В силу последнего соотношения следствия 5.1, (5.18) и (5.19) приходим к справедливости первого утверждения:

$$p(t) = (P^t)' P(0) = (P^t)' \pi^* = (P^{t-1})' P' \pi^* = (P^{t-1})' \pi^* = \dots = \pi^*.$$

Для доказательства второго утверждения достаточно воспользоваться определением 5.6 и аналогично (5.17) вычислить $(n+1)$ -мерное распределение для ОЦМ x_t при произвольном сдвиге времени $\tau \in \mathbb{N}$ с учетом только что доказанного результата:

$$\begin{aligned} & \mathbf{P}\{x_\tau = i_0, x_{\tau+1} = i_1, \dots, x_{\tau+n} = i_n\} = \\ &= \mathbf{P}\{x_\tau = i_0\} \cdot \prod_{t=0}^{n-1} \mathbf{P}\{x_{\tau+t+1} = i_{t+1} | x_{\tau+t} = i_t\} = \\ &= \pi_{i_0}^* \prod_{t=0}^{n-1} p_{i_t, i_{t+1}} = \mathbf{P}\{x_0 = i_0, x_1 = i_1, \dots, x_n = i_n\}. \end{aligned} \quad \square$$

Заметим, что в силу определения 5.18 существует предел $\lim_{t \rightarrow +\infty} P^t = \Pi^*$,

где $\Pi^* = (\pi^* : \pi^* : \dots : \pi^*)'$ – $(N \times N)$ -матрица, все строки которой одинаковы и совпадают с π^* . Поэтому из следствия 5.1 заключаем, что при любом начальном распределении $p(0)$ имеет место сходимость (с экспоненциальной скоростью):

$$p(t) \rightarrow \pi^* \text{ при } t \rightarrow +\infty.$$

Следовательно, каково бы ни было начальное распределение вероятностей $p(0)$, если исключить из рассмотрения начальный фрагмент $\{x_0, x_1, \dots, x_T\}$, то при достаточно большом T (которое можно легко оценить) ОЦМ $\{x_{T+1}, x_{T+2}, \dots\}$ мало отличается от стационарного ДВР.

Отметим еще, что методы компьютерного моделирования цепей Маркова представлены в [47, 55].

В заключение приведем некоторые полезные формулы расчета важнейших вероятностных характеристик для эргодических ОЦМ [19]. Примем следующие обозначения: $D = \text{diag}\{1/\pi_0^*, \dots, 1/\pi_{N-1}^*\}$ – диагональная $(N \times N)$ -матрица; \mathbf{I}_N – единичная $(N \times N)$ -матрица; $Z = (z_{ij}) = (\mathbf{I}_N - P + \Pi^*)^{-1}$ – так называемая фундаментальная $(N \times N)$ -матрица; Z_{diag} – матрица, полученная из Z заменой всех внедиагональных элементов нулями; $\mathbf{1}_N$ – $(N \times 1)$ -вектор-столбец, все элементы которого равны единице; $\mathbf{1}_{N \times N}$ – $(N \times N)$ -матрица, все элементы которой равны единице; $\bar{P} = (\bar{p}_{ij})$ – матрица вероятностей однопашевых переходов для обращенной стационарной цепи Маркова, т. е. для ОЦМ,

наблюдаемой в обратном времени; $M = (m_{ij})$, m_{ij} – математическое ожидание случайного времени 1-го достижения состояния j , исходя из состояния i ($i, j \in \mathcal{A}$); $W = (w_{ij})$, w_{ij} – дисперсия случайного времени 1-го достижения j из i ; $y_i^{(n)}$ – суммарное случайное время пребывания в состоянии i за первые n шагов; $C = (c_{ij})$,

$$c_{ij} = \lim_{n \rightarrow +\infty} n^{-1} \mathbf{Cov} \left\{ y_i^{(n)}, y_j^{(n)} \right\}, \quad i, j \in \mathcal{A}.$$

Справедливы следующие полезные соотношения [19]:

$$\begin{aligned} \bar{P} &= DP'D^{-1}; \\ M &= (\mathbf{I}_N - Z + \mathbb{1}_{N \times N} Z_{\text{diag}}) D; \quad \bar{M} = M - M_{\text{diag}}; \\ W &= M (2Z_{\text{diag}} D - \mathbf{I}_N) + 2 (ZM - \mathbb{1}_{N \times N} (ZM)_{\text{diag}}); \\ c_{ij} &= \pi_i^* z_{ij} + \pi_j^* z_{ji} - \pi_i^* \delta_{ij} - \pi_i^* \pi_j^*, \quad i, j \in \mathcal{A}; \\ m_{ii} &= 1/\pi_i^*; \quad w_{ii} = 2z_{ii}(\pi_i^*)^{-2} = (\pi_i^*)^{-1}; \\ \pi_0^* M &= \mathbb{1}_N' Z_{\text{diag}} D; \quad M \pi^* = C \mathbb{1}_N; \\ P &= \mathbf{I}_N + (D - \mathbb{1}_{N \times N})(\bar{M})^{-1}. \end{aligned}$$

В [19] приводятся также полезные формулы вычисления вероятностных характеристик для ОЦМ при наличии поглощающих состояний.

5.4. ЦЕПЬ МАРКОВА ПОРЯДКА s

В криптологии для моделирования ДВР с глубиной памяти $s \in \mathbb{N}$ используется цепь Маркова порядка s (ЦМ(s)), обобщая модель простой цепи Маркова из п. 5.3.

Определение 5.19. Цепь Маркова $x_t \in \mathcal{A}$, $t \in \mathbb{N}$, порядка s с пространством состояний \mathcal{A} , определенная на вероятностном пространстве $(\Omega, \mathcal{F}, \mathbf{P})$ и временной области \mathbb{N} , характеризуется обобщенным марковским свойством [13]:

$$\begin{aligned} &\mathbf{P} \{x_{t+1} = i_{t+1} | x_t = i_t, \dots, x_1 = i_1\} = \\ &= \mathbf{P} \{x_{t+1} = i_{t+1} | x_t = i_t, \dots, x_{t-s+1} = i_{t-s+1}\} = \\ &= p_{i_{t-s+1}, \dots, i_t, i_{t+1}}(t), \quad t \geq s, \quad i_1, \dots, i_{t+1} \in \mathcal{A}. \end{aligned} \tag{5.20}$$

Соотношение (5.20) означает, что условное распределение вероятностей будущих состояний $x_{t+1} \in \mathcal{A}$ при фиксированной предыстории зависит не от всей этой предыстории, а лишь от ближайшей на глубину s предыстории $(x_t, \dots, x_{t-s+1}) \in \mathcal{A}^s$. Если $s = 1$, то (5.20) эквивалентно соотношению (5.14) и ЦМ(1) называют *простой цепью Маркова*. Если $s = 0$, то ЦМ(0), по сути, является схемой независимых испытаний.

Цепь Маркова ЦМ(s) характеризуется s -мерным начальным распределением вероятностей

$$\pi_{i_1, \dots, i_s} = \mathbf{P}\{x_1 = i_1, \dots, x_s = i_s\}, i_1, \dots, i_s \in \mathcal{A},$$

и $(s+1)$ -мерной матрицей вероятностей одношаговых переходов в момент времени $t \geq s$:

$$\begin{aligned} P(t) &= \left(p_{i_1, \dots, i_{s+1}}(t) \right), p_{i_1, \dots, i_{s+1}}(t) = \\ &= \mathbf{P}\{x_{t+1} = i_{t+1} | x_t = i_t, \dots, x_{t-s+1} = i_{t-s+1}\}, \\ &i_1, \dots, i_{s+1} \in \mathcal{A}. \end{aligned} \quad (5.21)$$

Если $P(t)$ не зависит от времени t : $P(t) = P = (p_{i_1, \dots, i_{s+1}})$, $i_1, \dots, i_{s+1} \in \mathcal{A}$, то имеем однородную цепь Маркова s -го порядка (ОЦМ(s)). Матрица P удовлетворяет условиям нормировки

$$\sum_{i_{s+1} \in \mathcal{A}} p_{i_1, \dots, i_{s+1}} \equiv 1.$$

Оказывается, расширением пространства состояний ОЦМ(s) можно привести к простой цепи Маркова. Примем обозначения: $X_t = (x_{t-s+1}, \dots, x_t) \in \mathcal{A}^s$ – s -предыстория к моменту t ($t \geq s$); $\langle X \rangle = x_{t-s+1}N^{s-1} + \dots + x_{t-1}N + x_t \in \{0, 1, \dots, N^s - 1\}$ – число, N -ичная запись которого есть X_t .

Теорема 5.8. Если x_t , $t \geq 1$, есть ОЦМ(s) с матрицей вероятностей одношаговых переходов $P = (p_{i_1, \dots, i_{s+1}})$, $i_1, \dots, i_{s+1} \in \mathcal{A}$, то ДВР $y_t = \langle X_t \rangle$ есть простая цепь Маркова с пространством состояний $\mathbf{B} = \{0, 1, \dots, N^s - 1\}$ и матрицей вероятностей одношаговых переходов $P^X = (p_{\langle I \rangle, \langle J \rangle}^X)$, $I = (i_0, \dots, i_{s-1})$, $J = (j_0, \dots, j_{s-1}) \in \mathcal{A}^s$:

$$p_{\langle I \rangle, \langle J \rangle}^X = \begin{cases} p_{i_0, \dots, i_{s-1}, j_{s-1}}, & \text{если } j_0 = i_1, j_1 = i_2, \dots, j_{s-2} = i_{s-1}, \\ 0 & \text{в противном случае.} \end{cases} \quad (5.22)$$

Доказательство. Достаточно проверить марковское свойство первого порядка для ДВР X_t с использованием (5.21) и принятых обозначений. \square

Теорема 5.8 позволяет перенести всю теорию простых цепей Маркова из п. 5.3 на ОЦМ(s). В частности, получим один из критериев эргодичности ОЦМ(s).

Следствие 5.2. ОЦМ(s) эргодична тогда и только тогда, когда найдется такое число $n_0 \in \mathbb{N}$, что для любого $n \geq n_0$

$$\min_{I, J \in \mathcal{A}^s} ((P^X)^n)_{\langle I \rangle, \langle J \rangle} > 0,$$

где матрица P^X определяется (5.22).

В заключение отметим, что число независимых параметров, определяющих матрицу вероятностей одношаговых переходов $P = (p_{i_1, \dots, i_s+1})$ для ОЦМ(s), равно $D_{\text{ОЦМ}(s)} = N^s(N - 1) = Q(N^{s+1})$.

При увеличении глубины памяти s число параметров экспоненциально возрастает (см. табл.). Для идентификации такой модели требуется наблюдать реализацию x_1, x_2, \dots, x_n не всегда доступной на практике длительности $n > D_{\text{ОЦМ}(s)}$.

Число параметров ОЦМ(s) при $N=2$

s	1	2	8	16	32	64	128	256
$D_{\text{ОЦМ}(s)}$	2	4	256	$6,4 \cdot 10^4$	$4,3 \cdot 10^9$	$1,8 \cdot 10^{19}$	$3,4 \cdot 10^{38}$	$1,2 \cdot 10^{77}$

В связи с этим в криптологии начинают активно использоваться так называемые *малопараметрические модели цепей Маркова высокого порядка* [14, 46, 54], для которых матрица P задается числом параметров $D \ll \ll D_{\text{ОЦМ}(s)}$. Некоторые из таких малопараметрических моделей рассматриваются в следующих пунктах данной главы.

5.5. МОДЕЛЬ ДЖЕКОБСА – ЛЬЮИСА

Определение 5.20. Модель Джекобса – Льюиса [110] для ДВР x_t , $t \geq 1$, определяется стохастическим разностным уравнением порядка $s \geq 2$ со случайным запаздыванием:

$$x_t = \mu_t x_{t-\eta_t} + (1 - \mu_t) \xi_t, \quad t > s, \quad (5.23)$$

где $\{x_1, \dots, x_s\}, \{\xi_t, \eta_t, \mu_t : t > s\}$ – независимые в совокупности дискретные случайные величины на $(\Omega, \mathcal{F}, \mathbf{P})$ с распределениями вероятностей:

$$\begin{aligned} \mathbf{P}\{\xi_t = i\} &= \pi_i, \quad i \in \mathcal{A}; \quad \sum_{i \in \mathcal{A}} \pi_i = 1; \\ \mathbf{P}\{\eta_t = j\} &= \lambda_j, \quad j \in \{1, \dots, s\}; \quad \sum_{j=1}^s \lambda_j = 1, \quad \lambda_s \neq 0; \\ \mathbf{P}\{\mu_t = 1\} &= 1 - \mathbf{P}\{\mu_t = 0\} = \rho; \quad \mathbf{P}\{x_k = i\} = \pi_i, \quad i \in \mathcal{A}, \quad k \in \{1, \dots, s\}. \end{aligned} \quad (5.24)$$

Модель (5.23), (5.24) дает вероятностное описание криптографическому генератору случайной последовательности x_t , схема которого представлена на рис. 5.3.

Генератор состоит из трех простейших генераторов G_1 (двоичной последовательности ξ_t), G_2 (последовательности η_t), G_3 (двоичной последовательности μ_t), регистра сдвига, на каждом такте сдвигающего содержимое s своих ячеек на одну позицию влево, теряя последний бит x_{t-s} , и селектора, выбирающего один из своих входных сигналов, $x_{t-\eta_t}$ или ξ_t , в зависимости от значения μ_t .

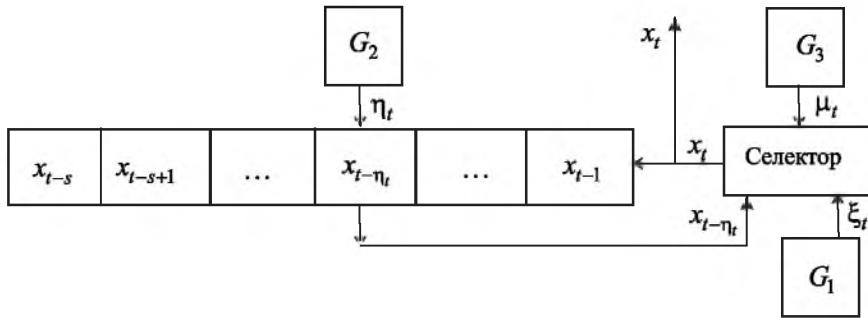


Рис. 5.3. Криптографический генератор последовательности x_t

Теорема 5.9. [44] *ДВР x_t , определяемый (5.23), (5.24), является ОЦМ(s) с начальным распределением вероятностей $\pi_{i_1, \dots, i_s} = \pi_{i_1} \cdot \dots \cdot \pi_{i_s}$ и $(s+1)$ -мерной матрицей вероятностей одношаговых переходов $P(\pi, \lambda, \rho) = (p_{i_1, \dots, i_{s+1}})$:*

$$p_{i_1, \dots, i_{s+1}} = (1 - \rho)\pi_{i_{s+1}} + \rho \sum_{j=1}^s \lambda_j \delta_{i_{s-j+1}, i_{s+1}}, \quad i_1, \dots, i_{s+1} \in \mathcal{A}. \quad (5.25)$$

Модель (5.23) предложена английскими статистиками П. Джекобсом и П. Льюисом в 1978 г. Малопараметрическое представление (5.25) $(s+1)$ -мерной матрицы вероятностей одношаговых переходов для модели Джекобса – Льюиса характеризуется числом параметров

$$D_{\text{JL}} = N + s - 1.$$

Видно, что вместо экспоненциальной зависимости числа параметров для общей модели ЦМ(s) ($D_{\text{ОЦМ}(s)} = N^s(N - 1)$) для модели Джекобса – Льюиса число параметров D_{JL} линейно зависит от s ; это создает существенный выигрыш в вычислительной сложности алгоритмов идентификации модели (см. гл. 6).

5.6. MTD-МОДЕЛЬ РАФТЕРИ

Эта модель предложена американским статистиком А. Рафтери [145] в 1985 г.

Определение 5.21. *MTD-модель (от англ. *Mixture Transition Distribution*) Рафтери для ДВР x_t , $t \geq 1$, задается следующим малопараметрическим представлением $(s+1)$ -мерной матрицы вероятностей одношаговых переходов $P = (p_{i_1, \dots, i_{s+1}})$ на основе смеси вероятностных распределений:*

$$p_{i_1, \dots, i_{s+1}} = \sum_{j=1}^s \lambda_j \cdot q_{i_j, i_{s+1}}, \quad i_1, \dots, i_{s+1} \in \mathcal{A}, \quad (5.26)$$

где $Q = (q_{i,k})$ – двумерная стохастическая $(N \times N)$ -матрица, $i, k \in \mathcal{A}$; $\lambda = (\lambda_1, \dots, \lambda_s)' - s$ -вектор-столбец дискретного распределения вероятностей смеси, для которого $\lambda_1 > 0$, $\lambda_2, \dots, \lambda_s \geq 0$, $\lambda_1 + \dots + \lambda_s = 1$.

Эта модель имеет $D_{\text{MTD}} = N(N-1)/2 + s - 1$ параметров. Обобщением модели (5.26) является MTDg-модель [145], в которой для каждого из s прошлых моментов времени используется «своя» матрица вероятностей переходов:

$$p_{i_1, \dots, i_{s+1}} = \sum_{j=1}^s \lambda_j q_{i_j, i_{s+1}}^{(j)}, \quad i_1, \dots, i_{s+1} \in \mathcal{A}, \quad (5.27)$$

где $Q^{(j)} = (q_{i,k}^{(j)})$ – j -я двумерная стохастическая матрица, соответствующая запаздыванию $s-j$, при этом число параметров $D_{\text{MTDg}} = s(N-1)/2 + 1 - 1$.

Приведем два результата о вероятностных свойствах моделей (5.26), (5.27) [44].

Теорема 5.10. Для того чтобы ОЦМ(s), определяемая моделью (5.26), была эргодической, необходимо и достаточно, чтобы существовало число $n_0 \in \mathbf{N}_0$ такое, что при любом $n \geq n_0$ все элементы матрицы Q^n положительны.

Примем обозначения: $\Pi^* = (\pi_{i_1, \dots, i_s}^*)$, $i_1, \dots, i_s \in \mathcal{A}$, – s -мерное стационарное распределение вероятностей эргодической цепи; $\pi^* = (\pi_0^*, \dots, \pi_{N-1}^*)'$ – одномерное (маргинальное) стационарное распределение вероятностей.

Теорема 5.11. Если x_t – эргодическая ОЦМ(s), удовлетворяющая модели (5.27), то ее стационарное распределение вероятностей удовлетворяет соотношению

$$\pi_{i_1, \dots, i_s}^* = \prod_{l=0}^{s-1} \left(\pi_{i_{s-l}}^* + \sum_{j=l+1}^s \lambda_j \left(q_{i_{j-l}, i_{s-l}}^{(j)} - \sum_{r=0}^{N-1} q_{r, i_{s-l}}^{(j)} \pi_r^* \right) \right).$$

Следствие 5.3. Если для модели (5.26) выполнено условие эргодичности, то для стационарного двумерного маргинального случайного вектора $(x_{t-m}, x_t)' \in \mathcal{A}^2$, $1 \leq m \leq s$, справедливо соотношение

$$\pi_{k,i}^*(m) = \pi_k^* \pi_i^* + \pi_k^* \lambda_{s-m+1} (q_{k,i} - \pi_i^*), \quad i, k \in \mathcal{A}.$$

5.7. ЦЕПЬ МАРКОВА С ЧАСТИЧНЫМИ СВЯЗЯМИ ЦМ(s, r)

Эта модель разработана в Белорусском государственном университете в 2003 г. [51]. Пусть, как и в п. 5.4, x_t – однородная ЦМ s -го порядка на $(\Omega, \mathcal{F}, \mathbf{P})$ с некоторой $(s+1)$ -мерной матрицей вероятностей переходов $P = (p_{i_1, \dots, i_{s+1}})$, $i_1, \dots, i_{s+1} \in \mathcal{A}$; $r \in \{1, \dots, s\}$ – параметр, который называется числом связей; $M_r^0 = (m_1^0, \dots, m_r^0) \in M$ – произвольный целочисленный r -вектор с упорядоченными компонентами $1 = m_1^0 < m_2^0 < \dots < m_r^0 \leq s$, который называется шаблоном связей; M – множество всевозможных таких векторов с r компонентами, имеющее мощность $K = |M| = C_{s-1}^{r-1}$; $Q^0 = (q_{j_1, \dots, j_r, j_{r+1}}^0)$, $j_1, \dots, j_{s+1} \in \mathcal{A}$, – некоторая $(r+1)$ -мерная стохастическая матрица.

Определение 5.22. Цепь Маркова x_t называется цепью Маркова s -го порядка с r частичными связями [51] и обозначается ЦМ(s, r), если ее вероятности одношаговых переходов имеют вид

$$p_{i_1, \dots, i_s, i_{s+1}} = q_{i_{m_1^0}, \dots, i_{m_r^0}, i_{s+1}}^0, \quad i_1, \dots, i_{s+1} \in \mathcal{A}. \quad (5.28)$$

Соотношение (5.28) означает, что вероятность перехода процесса в состояние i_{s+1} в момент времени $t > s$ зависит не от всех s предыдущих состояний процесса i_1, \dots, i_s , а лишь от r выбранных состояний $i_{m_1^0}, \dots, i_{m_r^0}$.

Таким образом, вместо $D_{\text{ЦМ}(s)} = N^s(N-1)$ параметров модель (5.28) полностью определяется $D_{\text{ЦМ}(s, r)} = N^r(N-1) + r - 1$ параметрами Q^0, M_r^0 . Выигрыш в числе параметров может оказаться весьма существенным: например, если $N = 2, s = 32, r = 3$, то $D_{\text{ЦМ}(s)} \approx 4,3 \cdot 10^9$, в то время как $D_{\text{ЦМ}(s, r)} = 10$.

Заметим, что если $s = r, M_r^0 = (1, \dots, s)$, то $P = Q^0$ и ЦМ(s, s) есть полносвязная цепь Маркова s -го порядка: ЦМ(s, s) = ЦМ(s). Конструктивным примером ЦМ(s, s) является бинарная ($N = 2$) авторегрессия s -го порядка с r ненулевыми коэффициентами, частным случаем которой является линейная рекуррентная последовательность над кольцом Z_2 , порожденная многочленом степени s с r ненулевыми коэффициентами.

Примем обозначения: $J_s = (j_1, \dots, j_s) = (J_{s-1}, j_s) \in \mathcal{A}^s$ – мультииндекс s -го порядка; δ_{J_s, J'_s} – символ Кронекера для мультииндексов J_s, J'_s ; $S_t(X_n; M_r) = (x_{t+m_1-1}, \dots, x_{t+m_r-1}) \in \mathcal{A}^r$ – функция $A^n \times M \rightarrow \mathcal{A}^r$, которую условимся называть селектором r -го порядка с параметрами $M_r \in M$ и $t \in \{1, \dots, n-s+1\}$.

Теорема 5.12. ЦМ(s, r), определяемая (5.28), эргодична тогда и только тогда, когда найдется целое число $l \geq 0$ такое, что

$$\min_{J_s, J'_s \in \mathcal{A}^s} \sum_{K_l \in \mathcal{A}^l} \prod_{i=1}^{s+l} q_{S_i((J_s, K_l, J'_s); M_{r+1}^0)}^0 > 0.$$

Доказательство основано на преобразовании ЦМ(s, r) в специальную векторную цепь Маркова первого порядка $\langle X \rangle$ (см. п. 5.4).

5.8. ДРУГИЕ МАЛОПАРАМЕТРИЧЕСКИЕ МОДЕЛИ ЦЕПЕЙ МАРКОВА ВЫСОКОГО ПОРЯДКА

В этом пункте дадим краткое описание еще некоторых малопараметрических моделей ОЦМ(s).

Определение 5.23. *Дискретная авторегрессия порядка s ($DAR(s)$) задается следующим стохастическим разностным уравнением [31, 44]:*

$$x_t = (\theta_1 x_{t-1} + \dots + \theta_s x_{t-s} + \xi_t) \bmod N, \quad t \geq s,$$

где $\theta_1, \dots, \theta_s \in \mathcal{A}$ – коэффициенты авторегрессии, причем $\theta_s \neq 0$; $\{\xi_t\}$ – независимые одинаково распределенные на \mathcal{A} случайные величины с некоторым дискретным распределением вероятностей:

$$\mathbf{P}\{\xi_t = k\} = p_k, \quad k \in \mathcal{A}.$$

Число параметров модели $DAR(s)$: $D_{DAR(s)} = N + s - 1$.

Определение 5.24. *Модель дискретного скользящего среднего порядка q ($DMA(q)$) имеет вид $x_t = (\alpha_0 \xi_t + \alpha_1 \xi_{t-1} + \dots + \alpha_q \xi_{t-q}) \bmod N$, $t \geq q$, где $\alpha_0 = 1$, $\alpha_1, \dots, \alpha_q \in \mathcal{A}$ – коэффициенты скользящего среднего.*

Обобщением моделей $DAR(s)$ и $DMA(q)$ является модель дискретной авторегрессии и скользящего среднего ($DARMA(s, q)$):

$$x_t = (\theta_1 x_{t-1} + \dots + \theta_s x_{t-s} + \xi_t + \alpha_1 \xi_{t-1} + \dots + \alpha_q \xi_{t-q}) \bmod N;$$

число параметров этой модели $D_{DARMA(s, q)} = N + s + q - 1$.

Определение 5.25. *Цепь Маркова переменного порядка (variable length Markov chain) определяется следующим малопараметрическим представлением $(s+1)$ -мерной матрицы одношаговых переходов [81] $P = (p_{i_1, \dots, i_{s+1}})$:*

$$p_{i_1, \dots, i_{s+1}} = q_{i_{s-l+1}, \dots, i_{s+1}},$$

где $l = l(i_1, \dots, i_s) : \mathcal{A}^s \rightarrow \{1, \dots, s\}$ – некоторая заданная дискретная функция.

Если $l(i_1, \dots, i_s) \equiv s$, то получаем полносвязную цепь Маркова ЦМ(s). Задание функции $l(\cdot)$ эквивалентно заданию так называемой *контекстной функции*:

$$c(i_1, \dots, i_s) = (i_{s-l+1}, \dots, i_s) : \mathcal{A}^s \rightarrow \mathcal{A}^l,$$

являющейся аналогом функции-селектора для модели ЦМ(s, r) (см. п. 5.7). Контекстная функция определяет *контекстное дерево*:

$$\tau = \{u : u = c(i_1, \dots, i_s), (i_1, \dots, i_s) \in \mathcal{A}^s\}.$$

5.9. ЗАДАНИЯ

1. Получить выражения для математического ожидания, дисперсии, ковариационной и корреляционной функций, а также спектральной плотности для стационарной двоичной ($N = 2$) цепи Маркова ЦМ(1) с матрицей вероятностей одношаговых переходов $P = \begin{pmatrix} p_0 & 1 - p_0 \\ 1 - p_1 & p_1 \end{pmatrix}$, $p_0, p_1 \in [0, 1]$.
2. Пусть $\{\xi_t : t \in \mathbb{Z}\}$ – независимые одинаково распределенные случайные величины Бернулли ($N = 2$), причем $\mathbf{P}\{\xi_t = 1\} = 1 - \mathbf{P}\{\xi_t = 0\} = p$. Доказать, что $\eta_n = \xi_1 \oplus \xi_2 \oplus \dots \oplus \xi_n$ является двоичной ЦМ(1), и найти ее матрицу вероятностей одношаговых переходов.
3. Пусть ξ_t , $t \in \mathbb{Z}$, есть ЦМ(s). Доказать, что векторная случайная последовательность $\eta_t = (\xi_t, \xi_{t-1}, \dots, \xi_{t-s+1})'$ образует ЦМ(1).
4. Доказать, что при обращении времени марковский характер последовательности сохраняется. Найти матрицу вероятностей одношаговых переходов для обращенной цепи Маркова.
5. Доказать, что марковский характер случайной последовательности при прореживании сохраняется.
6. Доказать, что если ξ_t, η_t , $t \in \mathbb{Z}$, – независимые эргодические ЦМ(1), то $\zeta_t = (\xi_t, \eta_t)'$ также является эргодической ЦМ(1).
7. Пусть $\xi_t \in \mathcal{A}$, $t \in \mathbb{Z}$, – однородная ЦМ(1) с матрицей вероятностей одношаговых переходов P . Доказать, что $\eta_t = (\xi_t, \xi_{t-1})' \in \mathcal{A}^2$ – также однородная ЦМ(1). Найти матрицу вероятностей одношаговых переходов для η_t .
8. Пусть $\xi_t \in \mathcal{A}$, $t \in \mathbb{Z}$, – однородная ЦМ(1) с матрицей вероятностей одношаговых переходов P , $\mathbf{B} \subset \mathcal{A}$ – некоторое подмножество в пространстве состояний, а $\eta_t \in \mathbf{B}$ – случайный процесс ξ_t , наблюдаемый только в те моменты времени, когда он находится в \mathbf{B} , т. е. $\xi_t \in \mathbf{B}$. Доказать, что η_t – также однородная ЦМ(1). Найти ее матрицу вероятностей одношаговых переходов.
9. Разработать алгоритм моделирования ЦМ(s), оценить его вычислительную сложность, протестировать программу на численных примерах.
10. Доказать свойства $C_3 - C_{10}$ для РРСП.
11. Реализовать на компьютере криптографический генератор, представленный на рис. 5.3 (зафиксировав некоторые простейшие генераторы G_1, G_2, G_3 , например, как LFSR), и провести его исследование с помощью статистического моделирования, а также теоретически с помощью модели Джекобса – Льюиса из п. 5.5.
12. Реализовать на компьютере МТД-модель ЦМ(s) из п. 5.6 и исследовать ее методом статистического моделирования.
13. Реализовать на компьютере модель ЦМ(s, r) из п. 5.7 и исследовать ее методом статистического моделирования.

Г л а в а 6

СТАТИСТИЧЕСКОЕ ТЕСТИРОВАНИЕ СЛУЧАЙНЫХ И ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

6.1. ПРОБЛЕМА СТАТИСТИЧЕСКОГО ТЕСТИРОВАНИЯ В КРИПТОЛОГИИ И БАТАРЕИ ТЕСТОВ

Случайные и псевдослучайные последовательности, а также их генераторы являются неотъемлемыми элементами современных систем криптографической защиты информации. Они применяются для решения следующих основных задач:

- 1) генерации гаммы в поточных крипtosистемах;
- 2) генерации сеансовых и других ключей в крипtosистемах;
- 3) генерации «случайных значений» параметров для многих систем ЭЦП;

4) формирования «случайных запросов» при реализации большинства существующих криптографических протоколов выработки общего секретного ключа и аутентификации, в том числе в протоколе Kerberos.

Генерация случайной последовательности с произвольным законом распределения вероятностей сводится к генерации равномерно распределенной случайной последовательности (РРСП). Как уже отмечалось в п. 5.2, РРСП – это последовательность дискретных случайных величин $x_1, x_2, \dots \in \mathcal{A} = \{0, 1, \dots, N - 1\}$ из конечного алфавита \mathcal{A} мощности $2 \leq N < +\infty$, обладающая двумя свойствами (гипотеза H_0):

C_1) для любого числа $n \in \mathbb{N}$ и произвольных индексов $1 \leq t_1 < \dots < t_n$ случайные биты $x_{t_1}, x_{t_2}, \dots, x_{t_n}$ независимы в совокупности;

C_2) для любого $t \in \mathbb{N}$ случайная величина x_t имеет равномерное на \mathcal{A} распределение вероятностей:

$$\mathbf{P}\{x_t = i\} = N^{-1}, \quad i \in \mathcal{A}.$$

Важнейшим требованием, предъявляемым к генераторам случайных последовательностей, является требование «статистической безопасности». Оно заключается в том, что при используемой длине n генерируемой реализации ее невозможно отличить от реализации РРСП на заданном уровне значимости $\varepsilon \in (0, 1)$. Для проверки этого требования используются статистические тесты (критерии). Статистический тест – это решающее правило, позволяющее по наблюдаемой реализации $x_1, x_2, \dots, x_n \in \mathcal{A}$ длины n (или конечной выборке таких реализаций) с заданным уровнем значимости осуществить проверку гипотезы $H_0 = \{\{x_t\} \text{ есть РРСП}\}$ против некоторой альтернативы

$H_1 = \{\text{нарушено свойство } C_1 \text{ или свойство } C_2\}$. Существуют различные типы альтернатив H_1 , каждый из которых порождает свой собственный статистический тест. Поэтому на практике при проверке требования «статистической безопасности» используют батареи статистических тестов [23, 123, 152, 165, 166]. Рассмотрим наиболее известные батареи тестов.

Исторически первой батареей тестов считается батарея статистических тестов, предложенная Д. Кнутом [23]. Она включает в себя следующие тесты: частотный тест, тест серий, тест интервалов, покер-тест, тест «собирателя купонов», тест перестановок, тест на монотонность, тест подпоследовательностей, тест «наибольшего из t ». Большинство этих тестов являются частными случаями универсального алгоритма тестирования (см. п. 6.2).

Батареей тестов, предъявляющей «более жесткие требования», чем батарея тестов Д. Кнута, является «DIEHARD»-батарея [123], которая предложена Дж. Марсалли в 1985 г. Статистические тесты, входящие в эту батарею, могут быть классифицированы согласно рис. 6.1.

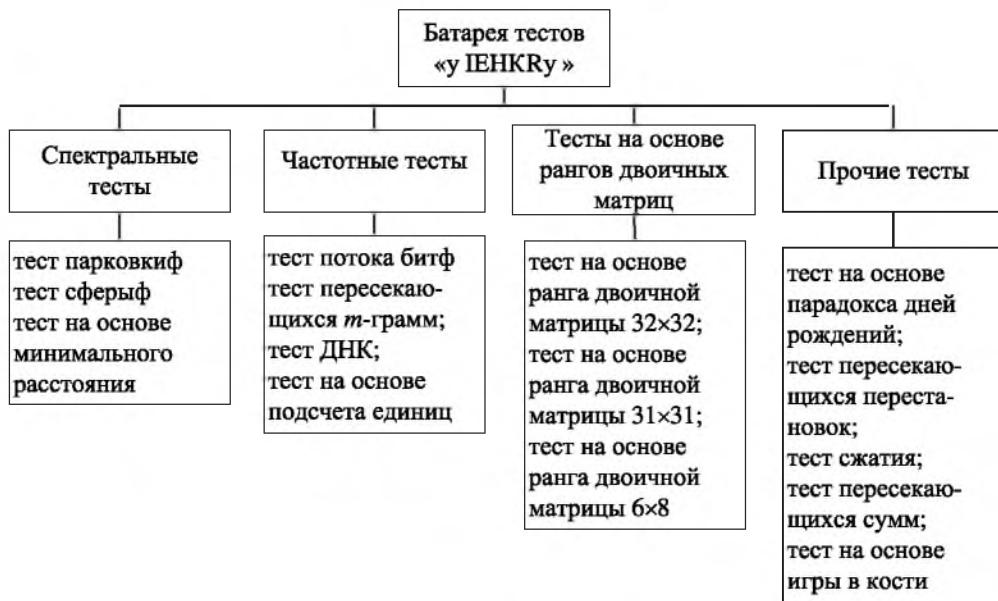


Рис. 6.1. Классификация статистических тестов батареи «DIEHARD»

Третья батарея тестов предложена NIST [152]. Она включает в себя следующие тесты: частотный тест; частотный тест внутри блока; тест серий; тест, основанный на рангах двоичных матриц; спектральный тест; универсальный статистический тест Маурера; тест, основанный на алгоритме сжатия Лемпеля – Зива; тест, основанный на нелинейной сложности; тест на основе аппроксимации энтропии и другие.

Батарея тестов CRYPT-X разработана австралийским Институтом безопасности информации [165]. Она классифицируется согласно рис. 6.2.

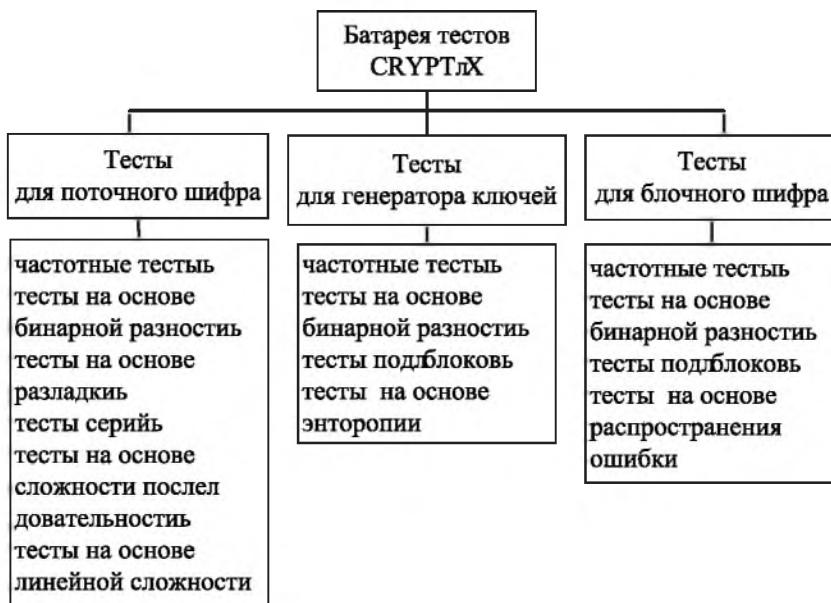


Рис. 6.2. Классификация статистических тестов батареи CRYPT-X

Еще одна батарея тестов, используемая в конкурсе криптоалгоритмов NESSIE [166], включает в себя следующие тесты: фильтрующий тест; корреляционный тест; тест Неймана – Пирсона; тест серий; тест на основе линейной аппроксимации; частотный тест; тест на основе парадокса дней рождения; универсальный статистический тест Маурера; покер-тест; тест, основанный на алгоритме сжатия Лемпеля – Зива; тест, основанный на нелинейной сложности; спектральный тест и другие.

Проведенный в [48, 50] обзор существующих статистических тестов показывает:

- 1) многие из существующих тестов ориентированы на проверку не главных свойств C_1 , C_2 , а лишь их частных случаев;
- 2) многие из известных тестов построены «эвристически» и не фиксируют семейство альтернатив H_1 ;
- 3) многие тесты не имеют оценок мощности;
- 4) при включении нескольких тестов в батарею не удается оптимизировать «составной» тест, используя вероятности ошибок первого и второго рода.

В связи с этим актуальны задачи разработки адекватных вероятностных моделей для описания отклонений H_1 от моделей РРСП, построения алгоритмов статистического анализа для обнаружения и оценивания таких отклонений. В пп. 6.8–6.17 приводится описание классических тестов, а пп. 6.19–6.22 содержат тесты, построенные на специальных моделях дискретных временных рядов, представляющих альтернативу H_1 .

6.2. УНИВЕРСАЛЬНЫЙ АЛГОРИТМ СТАТИСТИЧЕСКОГО ТЕСТИРОВАНИЯ СЛУЧАЙНЫХ И ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

В современных криптосистемах, как отмечалось в п. 6.1, интенсивно используются равномерно распределенные случайные последовательности или имитирующие их псевдослучайные последовательности. Оказывается, что стойкость криптосистем существенно зависит от того, насколько точно соответствует модели РРСП используемая последовательность $x_t \in \mathcal{A}$ ($t = 1, 2, \dots$). Проверка близости $\{x_t\}$ к модели РРСП осуществляется методами статистического тестирования и состоит в проверке гипотез выполнения базовых свойств C_1, C_2 , определяющих РРСП (см. п. 6.1). Иногда вместо базовых свойств C_1, C_2 проверяют выполнение свойств $C_3 - C_{10}$ РРСП, следующих из C_1 и C_2 .

Сформулируем *универсальный алгоритм статистического тестирования* случайных и псевдослучайных последовательностей. Пусть имеется генератор, порождающий случайную или псевдослучайную последовательность

$$x_1, x_2, \dots \in \mathcal{A} = \{0, 1, 2, \dots, N-1\}$$

необходимой длины. Пусть определены две гипотезы: H_{0n} – о том, что наблюдаемая последовательность $x_1, x_2, \dots, x_n \in \mathcal{A}$ заданной длины $n \in \mathbb{N}$ является РРСП и имеет n -мерное дискретное равномерное распределение вероятностей (см. свойство C_3 из п. 6.1); $H_{1n} = \overline{H_{0n}}$ – альтернатива общего вида, утверждающая, что свойство C_3 нарушено и $\{x_t\}$ не соответствует модели РРСП. Согласно принятой в математической статистике классификации гипотеза H_{0n} является простой, а H_{1n} – сложной.

Пусть для $m \in \mathbb{N}$ построена некоторая m -мерная векторная статистика

$$t = \begin{pmatrix} t_1 \\ \vdots \\ t_m \end{pmatrix} = t(x_1, \dots, x_n) = \begin{pmatrix} t_1(x_1, \dots, x_n) \\ \vdots \\ t_m(x_1, \dots, x_n) \end{pmatrix} \in \mathbb{R}^m, \quad (6.1)$$

удовлетворяющая двум условиям:

- 1) найдено дискретное распределение вероятностей

$$p_{0j} = \mathbf{P}_{H_{0n}} \{t(x_1, \dots, x_n) = u_j\} > 0, \quad j = \overline{1, L}, \quad (6.2)$$

статистики (6.1) при верной гипотезе H_{0n} , где $U = \{u_1, \dots, u_L\} \subset \mathbb{R}^m$ – конечное множество L упорядоченных (в лексикографическом порядке) возможных значений статистики (6.1);

2) при верной альтернативе H_{1n} дискретное распределение вероятностей

$$p_{1j} = \mathbf{P}_{H_{1n}} \{t(x_1, \dots, x_n) = u_j\}, \quad j = \overline{1, L}, \quad (6.3)$$

статистики (6.1) отличается от распределения (6.2).

Универсальный алгоритм тестирования основан на выявлении различия дискретных вероятностных распределений (6.2) и (6.3) по наблюдаемой выборке с помощью χ^2 -критерия согласия Пирсона и состоит из пяти этапов.

1. Генерируем подпоследовательность длиной $n_+ = Mn$ и разбиваем ее на M непересекающихся фрагментов:

$$X = (x_1, \dots, x_n, x_{n+1}, \dots, x_{2n}, \dots, x_{Mn}) = (X_1 \parallel X_2 \parallel \dots \parallel X_M),$$

где $X_i = (x_{(i-1)n+1}, x_{(i-1)n+2}, \dots, x_{in}) \in \mathcal{A}^n$ – i -й фрагмент $i \in \{1, 2, \dots, M\}$ длиной n ; M – количество фрагментов.

2. Для i -го ($i = \overline{1, M}$) фрагмента X_i вычисляем значение m -мерной статистики (6.1):

$$t_i = t(x_{(i-1)n+1}, \dots, x_{in}) \in U.$$

3. По сформированной таким образом выборке объема M вычисляем частоты встречаемости всех возможных значений статистики $t(\cdot)$:

$$\nu_j = \sum_{i=1}^M \delta_{t_i, u_j}, \quad j = \overline{1, L}, \quad (6.4)$$

где δ_{kl} – символ Кронекера.

4. Вычисляем χ^2 -статистику Пирсона:

$$\chi^2 = \sum_{j=1}^L \frac{(\nu_j - Mp_{0j})^2}{Mp_{0j}} \geq 0. \quad (6.5)$$

5. Выносим решение по правилу:

$$\text{принимается гипотеза } \begin{cases} H_{0n}, & \text{если } P > \varepsilon, \\ H_{1n}, & \text{если } P \leq \varepsilon, \end{cases} \quad (6.6)$$

где $P = 1 - G_{L-1}(\chi^2)$ – так называемое P -значение для статистики (6.5); $G_f(\cdot)$ – функция χ^2 -распределения с f степенями свободы [4]; $\varepsilon \in (0, 1)$ – задаваемый уровень значимости теста.

Замечание 6.1. При увеличении количества фрагментов ($M \rightarrow \infty$) асимптотический размер теста (6.4)–(6.6), т. е. вероятность ошибки первого рода, совпадает с заданным уровнем значимости ε :

$$\lim_{M \rightarrow \infty} \mathbf{P}_{H_{0n}} \{ \text{принимается } H_{1n} \} = \varepsilon$$

и тест является состоятельным, т. е. его мощность стремится к единице (вероятность ошибки второго рода стремится к нулю):

$$\lim_{M \rightarrow \infty} \mathbf{P}_{H_{1n}} \{ \text{принимается } H_{1n} \} = 1.$$

Замечание 6.2. Для практического применения теста (6.4)–(6.6) выбирается количество фрагментов M так, чтобы $M \min_j p_{0j} > 5$.

Замечание 6.3. Возможна модификация описанного теста, состоящая в проверке гипотезы о том, что статистика (6.5) имеет χ^2 -распределение с $L - 1$ степенями свободы. Для проверки такой гипотезы целесообразно s -кратно повторить выборку и использовать критерий Колмогорова [5], основанный на эмпирической функции распределения.

6.3. ТЕСТ n -СЕРИЙ

Классический тест n -мерной равномерности – это *тест n -серий*. Он входит в *батарею тестов Кнута* [23] и является частным случаем универсального алгоритма тестирования (6.4)–(6.6), описанного в п. 6.2:

$$\begin{aligned} U &= \mathcal{A}^n, \quad L = N^n; \quad t_k(x_1, \dots, x_n) = x_k, \quad k = \overline{1, n}; \\ m &= n; \quad p_{0j} = \frac{1}{N^n}, \quad j = \overline{1, L}. \end{aligned} \tag{6.7}$$

При этом вычисляемая согласно (6.4) статистика v_j – это частота встречаемости j -й возможной n -серии $(i_1, i_2, \dots, i_n) \in \mathcal{A}^n$:

$$j = 1 + \sum_{k=1}^n i_k \times N^{k-1}.$$

Отметим, что серии, отличающиеся только перестановкой символов, считаются различными, поэтому общее количество серий $L = N^n$.

Отметим, что с учетом (6.7) и замечания 6.2 для применения теста (6.4)–(6.6) требуется, чтобы количество M фрагментов исследуемой последовательности $\{x_t\}$ было не менее критической величины

$$M_{\min} = 5 \times N^n.$$

Таким образом, при увеличении исследуемого порядка равномерности n требуемое количество фрагментов растет экспоненциально быстро. В этом заключается главный недостаток теста n -серий.

6.4. ТЕСТ ИНТЕРВАЛОВ

Еще один классический тест n -мерной равномерности, входящий в батарею тестов Кнута [23], – *тест интервалов*. В этом тесте исследуются частоты случайных событий, состоящих в попадании элементов наблюдаемой последовательности $\{x_t\}$ в заданный числовой промежуток $[\alpha, \beta)$, где $\alpha, \beta \in \mathcal{A}$, $\alpha < \beta$. При этом оценивается случайная длина серии t непопаданий в $[\alpha, \beta)$ до первого попадания. Тест интервалов является частным случаем общего алгоритма тестирования (см. п. 6.2) при следующих значениях определяющих его характеристик:

$$n = \infty; \quad m = 1; \quad U = \mathbb{N}_0 = \{0, 1, 2, \dots\}; \quad L = \infty;$$

$$t(x_1, x_2, \dots) = \begin{cases} 0, & x_1 \in [\alpha, \beta), \\ 1, & x_1 \notin [\alpha, \beta), \quad x_2 \in [\alpha, \beta), \\ \dots & \\ t, & x_1, \dots, x_t \notin [\alpha, \beta), \quad x_{t+1} \in [\alpha, \beta), \\ \dots & \end{cases} \quad (6.8)$$

$$p_{0j} = p(1-p)^j; \quad j \in \mathbb{N}_0, \quad p = \frac{\beta - \alpha}{N}.$$

Отличительной особенностью данного теста является отсутствие заранее фиксированного размера фрагмента и, следовательно, заранее фиксированной длины n_+ генерируемой последовательности $\{x_t\}$; величина n_+ определяется как суммарная длина M фрагментов:

$$n_+ = \sum_{k=1}^M (t^{(k)} + 1), \quad (6.9)$$

где $t^{(k)} \in \mathbb{N}_0$ – зарегистрированная длина серии непопаданий в k -м фрагменте.

Заметим, что в силу (6.8) и (6.9) и свойств геометрического распределения вероятностей $\{p_{0j}\}$ средняя длина последовательности, необходимой для построения теста интервалов (6.4)–(6.6), (6.8), равна

$$\bar{n}_+ = \mathbf{E}\{n_+\} = \frac{MN}{\beta - \alpha},$$

а среднеквадратическое отклонение

$$\sigma_{n_+} = \sqrt{\mathbf{D}\{n_+\}} = \frac{N}{\beta - \alpha} \sqrt{M \left(1 - \frac{\beta - \alpha}{N}\right)}.$$

6.5. ОБОБЩЕННЫЙ ПОКЕР-ТЕСТ

Обобщенный покер-тест и его специальный случай – *классический покер-тест* – также относятся к классическим тестам n -мерной равномерности, входящим в батарею тестов Кнута [23]. В этом тесте вида (6.4)–(6.6) исследуются частоты встречаемости некоторых комбинаций символов в n -сериях $x = (x_1, \dots, x_n) \in \mathcal{A}^n$. Статистика $t = t(x)$, на которой основан обобщенный покер-тест, определяется как количество различных символов в n -серии x ($n \geq 2$). При этом

$$L = \min(n, N),$$

$$p_{0j} = \frac{N(N-1) \cdots (N-j+1)}{N^n} \begin{Bmatrix} n \\ j \end{Bmatrix}, \quad j \in U = \{1, 2, \dots, L\},$$

где $\begin{Bmatrix} n \\ j \end{Bmatrix}$ – число Стирлинга второго рода [35]. Комбинаторные числа Стирлинга определяются равенствами

$$x^n = \sum_{j=0}^n \begin{Bmatrix} n \\ j \end{Bmatrix} x(x-1) \cdots (x-j+1), \quad \begin{Bmatrix} n \\ j \end{Bmatrix} = \frac{1}{j!} \sum_{l=0}^j (-1)^{j-l} \binom{j}{l} l^n$$

и могут быть вычислены при помощи рекуррентных соотношений

$$\begin{Bmatrix} n \\ j \end{Bmatrix} = \begin{Bmatrix} n-1 \\ j-1 \end{Bmatrix} + j \begin{Bmatrix} n-1 \\ j \end{Bmatrix}, \quad n \geq 1,$$

с начальными условиями $\begin{Bmatrix} 0 \\ 0 \end{Bmatrix} = 1$, $\begin{Bmatrix} 0 \\ j \end{Bmatrix} = 0$, $j \neq 0$.

В классическом покер-тесте рассматриваются серии длиной $n = 5$, $n \leq N$, а статистика $t = t(x)$ определяет количество различных элементов, содержащихся в серии x :

$$t(x) = \begin{cases} 1, & \text{если } x \text{ состоит из 5 одинаковых элементов,} \\ 2, & \text{если } x \text{ содержит 2 различных элемента,} \\ 3, & \text{если } x \text{ содержит 3 различных элемента,} \\ 4, & \text{если } x \text{ содержит 4 различных элемента,} \\ 5, & \text{если все элементы в } x \text{ различны.} \end{cases}$$

Распределение вероятностей этой статистики при гипотезе H_{0n} легко находится с помощью комбинаторного анализа и приведено в работе [42].

6.6. ТЕСТ «СОБИРАТЕЛЯ КУПОНОВ»

Частным случаем теста (6.4)–(6.6) также является *тест «собирателя купонов»*, который относится к классическим тестам Кнута [23] и основывается на исследовании случайной длины фрагмента последовательности $\{x_t\}$, обладающего заданным характерным свойством. Этот тест обычно применяется при $N > 2$. Если исходная тестируемая последовательность – двоичная, то можно увеличить мощность алфавита до значения $N' = 2^r$, рассматривая «укрупненную последовательность» фрагментов размера $r > 1$:

$$x'_1 = \sum_{i=1}^r x_i 2^{i-1}, \quad x'_2 = \sum_{i=1}^r x_{i+r} 2^{i-1}, \dots$$

Для проверки гипотезы n -мерной равномерности целесообразно брать $r = n$, тогда $\mathcal{A}' = \{0, 1, \dots, 2^n - 1\}$, $N' = 2^n$.

В teste «собирателя купонов» находится такой фрагмент $\{x'_1, x'_2, \dots, x'_t\}$ минимальной длины t в «укрупненной последовательности», который «впервые соберет полный комплект N' купонов», т. е. содержит все N' символов из \mathcal{A}' .

Тест «собирателя купонов» имеет вид (6.4)–(6.6) при следующих значениях определяющих его характеристик:

$$\begin{aligned} m &= 1; \quad U = \{N', N' + 1, \dots\}; \quad L = \infty; \\ t(x'_1, x'_2, \dots) &= \min \{\tau \in \mathbb{N}_0 : \mathcal{A}' \subseteq \{x'_1, x'_2, \dots, x'_\tau\}\}; \\ p_{0j} &= \frac{N'!}{(N')^j} \left\{ \begin{array}{l} j-1 \\ N'-1 \end{array} \right\}, \quad j \in U. \end{aligned} \tag{6.10}$$

6.7. ТЕСТ ПЕРЕСТАНОВОК

Еще один из классических тестов Кнута [23] – *тест перестановок*, являющийся частным случаем теста общего вида (6.4)–(6.6).

Тест перестановок можно рассматривать как развитие покер-теста, основанного на частоте встречаемости различных перестановок символов «укрупненной последовательности»

$$x'_t \in \mathcal{A}' = \{0, 1, \dots, N' - 1\}, \quad N' = 2^r,$$

формируемой как и в п. 6.6.

Для построения теста выберем произвольное натуральное число (параметр теста) $2 \leq n' \leq N'$ и рассмотрим вариационный ряд

$$0 \leq x'_{(1)} \leq x'_{(2)} \leq \dots \leq x'_{(n')} \leq N' - 1$$

«укрупненной последовательности» $x'_1, x'_2, \dots, x'_{n'}$. Статистику $t(\cdot)$ теста определим следующим комбинаторным соотношением:

$$t = t(x'_1, \dots, x'_{n'}) = \begin{cases} j, & \text{если } x'_1, \dots, x'_{n'} \\ & \text{попарно различны,} \\ n'! + 1 & \text{в противном случае,} \end{cases} \quad (6.11)$$

где $j = j(x'_1, \dots, x'_{n'})$ – номер перестановки, переводящей упорядоченный набор $(x'_{(1)}, \dots, x'_{(n')})$ в наблюдаемый набор $(x'_1, \dots, x'_{n'})$.

При этом в обозначениях п. 6.2

$$U = \{1, 2, \dots, n'!, n'! + 1\}, \quad L = n'! + 1;$$

$$p_{0j} = \begin{cases} \frac{1}{n'!} \prod_{i=0}^{n'-1} \left(1 - \frac{i}{N'}\right), & j \in \{1, 2, \dots, n'!\}, \\ 1 - \prod_{i=0}^{n'-1} \left(1 - \frac{i}{N'}\right), & j = n'! + 1. \end{cases}$$

6.8. ТЕСТ ПЕРЕСЕКАЮЩИХСЯ n -ГРАММ

Этот тест входит в батарею статистических тестов Дж. Марсальи, предложенную им в 1985 г. и предъявляющую «более жесткие требования» к исследуемой последовательности $\{x_t\}$, чем батарея тестов Кнута. Данный тест похож на использованный в заданиях 1, 5. Отличие состоит в том, что для построения теста используются частоты всевозможных комбинаций, вычисленные по пересекающимся n -граммам. Это усложняет нахождение распределения тестовой статистики при гипотезе H_{0n} , но существенно уменьшает длину n_+ требуемой для тестирования случайной последовательности $\{x_t\}$.

Тест пересекающихся n -грамм состоит в следующем. Подвергаемая тестируанию последовательность $(x_1, x_2, \dots, x_{n_+}) \in \mathcal{A}^{n_+}$ длиной n_+ ($n_+ > n$) «зацикливается» добавлением в конце $n - 1$ ее начальных символов:

$$X' = (x_1, x_2, \dots, x_{n-1}, x_n, \dots, x_{n_+}, x_1, x_2, \dots, x_{n-1}).$$

Затем анализируются n_+ пересекающихся n -грамм (со сдвигом на одну позицию):

$$(x_1, x_2, \dots, x_n), (x_2, x_3, \dots, x_{n+1}), \dots, (x_{n_+}, x_1, \dots, x_{n-1}) \in \mathcal{A}^n$$

и для $i_1, i_2, \dots, i_n \in \mathcal{A} = \{0, 1, \dots, N - 1\}$ вычисляются частоты встречаемости N^n всевозможных n -грамм:

$$\nu_{i_1, \dots, i_n} = \sum_{t=1}^{n+} \mathbf{1}\{\{x_t = i_1, x_{t+1} = i_2, \dots, x_{t+n-1} = i_n\}\}.$$

Аналогично вычисляются частоты встречаемости N^{n-1} всевозможных $(n - 1)$ -грамм $\{\nu_{i_1, \dots, i_{n-1}}\}$.

С использованием вычисленных частот строится статистика

$$Q = \sum_{i_1, \dots, i_n \in \mathcal{A}} \frac{(\nu_{i_1, \dots, i_n} - n_+ N^{-n})^2}{n_+ N^{-n}} - \sum_{i_1, \dots, i_{n-1} \in \mathcal{A}} \frac{(\nu_{i_1, \dots, i_{n-1}} - n_+ N^{-n+1})^2}{n_+ N^{-n+1}} \geq 0. \quad (6.12)$$

Доказано [124], что при $n_+ \rightarrow \infty$ для истинной гипотезы H_{0n} распределение статистики Q сходится к χ^2 -распределению с $f = N^n - N^{n-1} = N^{n-1}(N - 1)$ степенями свободы:

$$\mathcal{L}\{Q|H_{0n}\} \rightarrow \chi_f^2.$$

Данный факт позволяет по аналогии с п. 6.2 построить решающее правило, основанное на P -значении $P = 1 - G_f(Q)$:

$$\text{принимается гипотеза } \begin{cases} H_{0n}, & \text{если } P > \varepsilon, \\ H_{1n}, & \text{если } P \leq \varepsilon, \end{cases}$$

где $\varepsilon \in (0, 1)$ – заданный уровень значимости теста; $G_f(\cdot)$ – стандартная функция χ^2 -распределения с f степенями свободы.

6.9. ТЕСТ, ОСНОВАННЫЙ НА РАНГАХ ДВОИЧНЫХ МАТРИЦ

Данный тест, входящий в батарею статистических тестов Дж. Марсальи, является частным случаем теста (6.4)–(6.6) и основан на вероятностных свойствах двоичной матрицы, которая строится по исследуемому фрагменту двоичной последовательности $x_1, x_2, \dots, x_n \in \mathcal{A}$.

Пусть длина n исследуемого фрагмента последовательности представима в виде произведения двух чисел:

$$n = k \times l, \quad (6.13)$$

где k и l – некоторые натуральные числа, $2 \leq k \leq l$. Представим исследуемый фрагмент $x_1, x_2, \dots, x_{kl} \in \mathcal{A}$ в виде двоичной $(k \times l)$ -матрицы

$$X = (x_{ij}) := \begin{pmatrix} x_1 & x_2 & \dots & x_l \\ x_{l+1} & x_{l+2} & \dots & x_{2l} \\ \dots & & & \\ x_{(k-1)l+1} & x_{(k-1)l+2} & \dots & x_{kl} \end{pmatrix}. \quad (6.14)$$

Для теста (6.4)–(6.6) в качестве тестовой статистики $t(\cdot)$ используется ранг двоичной матрицы (6.14):

$$t = t(X) = \text{rank}(X) \in U = \{0, 1, \dots, k\}. \quad (6.15)$$

Вероятностные свойства статистики (6.15), необходимые для построения теста (6.4)–(6.6), выражаются следующим утверждением [124, 151].

Теорема 6.1. *Если верна гипотеза n -мерной равномерности H_{0n} , то распределение вероятностей статистики (6.15) имеет вид*

$$\begin{aligned} \mathbf{P}_{H_{0n}} \{\text{rank}(X) = j\} &= p_{0j} = \\ &= 2^{j(k+l-j)-kl} \prod_{i=0}^{j-1} \frac{(1 - 2^{i-k})(1 - 2^{i-l})}{1 - 2^{i-j}}, \quad j \in U. \end{aligned} \quad (6.16)$$

Отметим, что случай $j = 0$ соответствует нулевой матрице X ; при этом

$$p_{00} = 2^{-kl}. \quad (6.17)$$

Таким образом, *тест, основанный на рангах двоичных матриц*, имеет вид (6.4)–(6.6) с учетом обозначений (6.13)–(6.17).

В заключение отметим, что вычисление ранга (6.15) осуществляется методом Гаусса: приведением матрицы X к матрице верхнего треугольного вида с помощью элементарных преобразований ее строк.

6.10. СПЕКТРАЛЬНЫЕ ТЕСТЫ

Тесты псевдослучайных (случайных) последовательностей, основанные на *преобразовании Фурье* (спектре Фурье) исследуемых последовательностей, называются *спектральными тестами*.

Рассмотрим спектральный тест для выявления скрытой периодичности в исследуемой последовательности $x_1, x_2, \dots, x_n \in \mathbb{R}$. Определим гипотезу H_0 : $x_t = \xi_t + f(t)$ – РПСП с нулевым средним значением и дисперсией $\sigma^2 = \mathbf{D}\{x_t\}$ и альтернативу H_1 о наличии скрытой периодичности с некоторым неизвестным периодом $1 \leq T_0 < n/2$ и соответствующей частотой $\omega_0 = 2\pi/T_0$:

$$x_t = \xi_t + f(t), \quad f(t) = \sum_{k=1}^K (a_k \cos(k\omega_0 t) + b_k \sin(k\omega_0 t)), \quad (6.18)$$

где $f(t)$ – периодический тренд с некоторыми неизвестными коэффициентами $\{a_k\}, \{b_k\}$; $K \geq 1$ – параметр тренда, определяющий количество гармоник с частотами, кратными основной исследуемой частоте ω_0 .

Если исследуется «чисто гармонический» тренд с единственной частотой ω_0 , то полагаем $K = 1$. Будем предполагать, что n кратно T_0 . Это предположение не является критическим, если n достаточно велико.

Определим конечное множество $L = [n/(2K)]$ частот

$$\Omega = \{\omega_j : \omega_j = 2\pi j/n, j = \overline{1, L}\}$$

и дискретное преобразование Фурье временного ряда $\{x_t\}$ на частоте $\omega \in [0, \pi]$ (i – мнимая единица):

$$X(\omega) = \frac{2}{n} \left(\sum_{t=1}^n x_t \cos(\omega t) + i \sum_{t=1}^n x_t \sin(\omega t) \right). \quad (6.19)$$

Для обнаружения скрытой периодичности строится специальная статистика – гармограмма:

$$H(\omega_j) = \frac{n}{2\sigma^2} \sum_{k=1}^K |X(k\omega_j)|^2 \geq 0, \quad j = \overline{1, L}. \quad (6.20)$$

Оказывается, если верна гипотеза H_0 , то при $n \rightarrow \infty$ и известной дисперсии σ^2 статистика (6.20) имеет асимптотическое χ^2 -распределение с $2K$ степенями свободы ($j = \overline{1, L}$) [3]:

$$\mathcal{L}\{H(\omega_j)\} \rightarrow \chi^2_{2K}. \quad (6.21)$$

Соотношения (6.18)–(6.21) позволяют построить следующий алгоритм статистического обнаружения периодичности и оценивания периода T_0 .

1. Вычисление максимума гармограммы:

$$M_H = \max_{1 \leq j \leq L} H(\omega_j), \quad j_0 = \arg \max_{1 \leq j \leq L} H(\omega_j).$$

2. Статистическая проверка гипотез H_0, H_1 с помощью решающего правила:

$$\text{принимается } \begin{cases} H_0, & \text{если } P_1 > \varepsilon, \\ H_1, & \text{если } P_1 \leq \varepsilon, \end{cases}$$

где $P_1 = 1 - (G_{2K}(M_H))^L$ – P -значение; $G_f(z)$ – функция χ^2 -распределения с f степенями свободы; ε – заданный уровень значимости.

3. Если принятая гипотеза H_1 , то вычисляются статистические оценки частоты и периода:

$$\hat{\omega}_0 = \omega_{j_0}, \quad \hat{T}_0 = \frac{n}{j_0}.$$

Построим семейство спектральных тестов, основанных на асимптотическом свойстве (6.21) при $K = 1$. Согласно (6.19) и (6.20) построим последовательность значений выборочной спектральной плотности (периодограммы):

$$h_j = \frac{n}{2\sigma^2} \left| X \left(\frac{2\pi j}{n} \right) \right|^2 \geq 0, \quad j = 1, 2, \dots, n/2; \quad (6.22)$$

здесь предполагается, что n – четное число.

Известно [3], что если верна гипотеза H_0 , то определяемые (6.22) случайные величины $\{h_j\}$ при $n \rightarrow \infty$ асимптотически независимы и одинаково распределены по закону χ^2 с двумя степенями свободы:

$$\mathcal{L}\{h_j\} \rightarrow \chi^2_2, \quad F_{h_j}(z) \rightarrow G_2(z), \quad z \geq 0. \quad (6.23)$$

Зададимся некоторым положительным числом $c > 0$ и с помощью (6.22) построим последовательность двоичных случайных величин ($j = 1, 2, \dots, n/2$):

$$\xi_j = \begin{cases} 1, & \text{если } h_j < c, \\ 0, & \text{если } h_j \geq c. \end{cases} \quad (6.24)$$

С учетом (6.22)–(6.24) гипотеза H_0 трансформируется в гипотезу H'_0 : $\{\xi_j\}$ – случайная выборка объема $n/2$ из распределения Бернулли: $Bi(1, p)$, $p = G_2(c)$.

Для проверки гипотезы H'_0 против альтернативы общего вида $H'_1 = \overline{H'_0}$ построим тест, основанный на статистике

$$m_n = \sum_{j=1}^{n/2} \xi_j \in \{0, 1, \dots, n/2\}, \quad (6.25)$$

представляющей собой частоту появления «1» в выборке $\{\xi_j\}$. В силу теоремы Муавра – Лапласа при $n \rightarrow \infty$ и истинной гипотезе H_0 нормированная статистика

$$\eta_n = \frac{m_n - np/2}{\sqrt{np(1-p)/2}} \in \mathbb{R} \quad (6.26)$$

распределена асимптотически нормально:

$$\mathcal{L}\{\eta_n\} \rightarrow \mathcal{N}_1(0, 1).$$

На основании этого факта можно построить тест, асимптотический размер которого равен заданному значению $\varepsilon \in [0, 1]$:

$$\text{принимается } \begin{cases} H_0, & \text{если } P_2 > \varepsilon, \\ H_1, & \text{если } P_2 \leq \varepsilon, \end{cases} \quad (6.27)$$

где $P_2 = 2(1 - \Phi(\eta_n))$; $\Phi(\cdot)$ – функция распределения стандартного нормального закона.

Отметим, что $c > 0$ – параметр построенного семейства решающих правил (6.22)–(6.27). В работе [151] используется частный случай этого решающего правила, когда параметр c выбран из условия

$$p = G_2(c) = 0,95, \quad c = G_2^{-1}(0,95) \approx 6,$$

а тестируемая последовательность $\{x_t\}$ является двоичной и для достижения нулевого значения математического ожидания преобразуется следующим образом:

$$x_t := (-1)^{x_t} \in \{-1, 1\}.$$

При этом для гипотезы H_0 дисперсия $\sigma^2 = 1$, а формула (6.24) принимает упрощенный вид:

$$\xi_j = \begin{cases} 1, & \text{если } |X(2\pi j/n)| < \sqrt{12/n}, \\ 0, & \text{если } |X(2\pi j/n)| \geq \sqrt{12/n}. \end{cases}$$

Построим еще один спектральный тест, также основанный на асимптотическом (при $n \rightarrow \infty$) поведении статистик $\{h_j\}$, определяемых (6.22). Для этого по выборке $\{h_j\}$ построим *выборочную (эмпирическую) функцию распределения*

$$F_n(z) = \frac{2}{n} \sum_{j=1}^{n/2} \mathbf{1}(z - h_j), \quad z \geq 0, \quad (6.28)$$

затем вычислим *расстояние Колмогорова* между выборочной функцией распределения (6.28) и гипотетической функцией распределения $G_2(z)$:

$$D_n = \sqrt{\frac{n}{2}} \max_{z \geq 0} |F_n(z) - G_2(z)|. \quad (6.29)$$

Тест Колмогорова, основанный на статистике (6.29), имеет следующий вид [55]:

$$\text{принимается } \begin{cases} H_0, & \text{если } D_n < \Delta_\varepsilon, \\ H_1, & \text{если } D_n \geq \Delta_\varepsilon, \end{cases} \quad (6.30)$$

где $\Delta_\varepsilon = \mathcal{K}^{-1}(1 - \varepsilon)$ – квантиль уровня $1 - \varepsilon$ распределения Колмогорова. Приведем в таблице некоторые значения функции Δ_ε :

Уровень значимости ε	0,01	0,05	0,10	0,20
Квантиль Δ_ε	1,63	1,36	1,22	1,07

6.11. ТЕСТЫ СЛУЧАЙНОГО БЛУЖДАНИЯ

Предполагается, что исследуемая последовательность $\{x_t\}$ двоичная: $x_t \in \mathcal{A}$. С помощью преобразования $y_t = 2x_t - 1$, $t = \overline{1, T}$ перейдем к последовательности $\{y_t\}$, $y_t \in \{-1, 1\}$. Обозначим через S_n частичную сумму n элементов последовательности $\{y_t\}$: $S_n = \sum_{i=1}^n y_i$, $n > 0$, $S_0 = 0$. Тесты случайного блуждания основаны на вероятностных свойствах траектории S_n , $n = 1, 2, \dots$.

Если $\{y_t\}$ – РРСП (гипотеза H_0), то для распределения частичных сумм $\{S_0, S_1, \dots, S_n\}$ известны следующие результаты [42]:

$$p_{n,r} = \mathbf{P}\{S_n = r\} = \left(\frac{n}{n+r}\right)2^{-n}, \quad p_{0,r} = \delta_{0,r}, \quad r = \overline{0, n},$$

$$u_{2n} = \mathbf{P}\{S_{2n} = 0\} = \binom{2n}{n}2^{-2n}.$$

Определим статистики [42], на основании которых построим тесты случайного блуждания.

Количество возвратов $\text{RN}_{2L} = |\{2k: S_{2k} = 0, 0 < k \leq L\}|$. Распределение вероятностей статистики RN_{2L} при условии, что верна H_0 , имеет вид

$$\mathbf{P}\{\text{RN}_{2L} = r\} = p_{2L-r,r}, \quad 0 \leq r \leq L. \quad (6.31)$$

Количество изменений знаков $\text{CH}_{2L+1} = |\{2k: S_{2k-1}S_{2k+1} < 0, 0 < k \leq L\}|$. Распределение вероятностей CH_{2L+1} при истинности H_0 :

$$\mathbf{P}\{\text{CH}_{2L+1} = r\} = 2p_{2L+1,2r+1}, \quad 0 \leq r \leq L. \quad (6.32)$$

Вес Хэмминга $\text{HW}_{2L} = \sum_{k=1}^{2L} x_t = S_{2L}/2 + L$. Распределение вероятностей HW_{2L} при истинности H_0 :

$$\mathbf{P}\{\text{HW}_{2L} = r\} = \binom{2L}{r}2^{-2L}, \quad 0 \leq r \leq 2L. \quad (6.33)$$

Исходная последовательность $\{x_t\}$ длиной n разбивается на фрагменты длиной $2L$ (для статистики CH_{2L+1} – длиной $2L + 1$): $\{X_1, \dots, X_M\}$. На каждом из них вычисляются ряды частичных сумм $\{S_{(1)}, \dots, S_{(M)}\}$, $S_{(j)} = (S_j^0, \dots, S_j^M)$, $j = \overline{1, M}$.

Для каждого ряда $S_{(j)}$, $j = \overline{1, M}$, вычисляются значения используемой статистики (одной из трех описанных ранее); условимся статистику обозначать $\gamma_{2L}^j = \gamma_{2L}(S_{(j)})$. В результате получаем последовательность значений статистики $\{\gamma_{2L}^j\}$, $j = \overline{1, M}$.

Данную последовательность M значений статистики разбивают на J групп $\{\Gamma_{(1)}, \dots, \Gamma_{(J)}\}$ одинаковой длины $M_J = M/J$, $\Gamma_{(i)} = \{\gamma_{2L}^{(i-1)M_J+1}, \dots, \gamma_{2L}^{iM_J}\}$. По каждой группе $\Gamma_{(i)}$, $i = \overline{1, J}$, находится эмпирическое распределение вероятностей статистики γ_{2L} :

$$p_m^j = \left| \{l : \gamma_{2L}^l = m, l = \overline{(j-1)M_J + 1, jM_J}\} \right|, \quad m = \overline{0, 2L}, \quad j = \overline{1, J}.$$

На основании j -го эмпирического распределения вероятностей вычисляем χ^2 -статистику:

$$q_j = \sum_{m=0}^K \frac{(\tilde{p}_m^j - M_J \mu_{2L, m})^2}{M_J \mu_{2L, m}},$$

где $K = \max\{n : M_J \mathbf{P}\{\gamma_{2L} = n\} > 10\}$; $\mu_{2L, k} = \mathbf{P}\{\gamma_{2L} = k\}$; $\tilde{p}_k^j = p_k^j$, $k < K$;
 $\mu_{2L, K} = \sum_{k=K}^{2L} \mathbf{P}\{\gamma_{2L} = k\}$; $\tilde{p}_K^j = \sum_{k=K}^{2L} \tilde{p}_k^j$.

В результате получаем выборку $\{q_j\}$, $j = \overline{1, J}$, значений χ^2 -статистики для групп $\{\Gamma_{(1)}, \dots, \Gamma_{(J)}\}$. Далее находим выборочную функцию распределения $F_{J(z)}$ по выборке $\{q_j\}$, $j = \overline{1, J}$:

$$F_J(z) = \frac{1}{J} \sum_{j=1}^J \mathbf{1}\{z - q_j\}, \quad z \geq 0.$$

С помощью этой выборочной функции распределения вычисляется расстояние Колмогорова до функции χ^2 -распределения $G_{J-1}(z)$ с $J - 1$ степенями свободы:

$$D_J = \sqrt{J} \max_{z \geq 0} |F_J(z) - G_{J-1}(z)| \geq 0.$$

На основе этой статистики строится тест Колмогорова (аналогично (6.30)).

6.12. УНИВЕРСАЛЬНЫЙ СТАТИСТИЧЕСКИЙ ТЕСТ МАУРЕРА

Данный тест основан на следующей идее: последовательность случайна, когда ее невозможно значительно упаковать или сжать. Для этого вводится мера, которая оценивает степень сжимаемости. Впервые этот метод был предложен Я. Зивом [164]; в данном пункте используется мера Элиса и Вильямса [92, 161] для *универсального кодирующего алгоритма*.

Пусть x_1, \dots, x_{n_+} – наблюдаемая последовательность над алфавитом $\mathcal{A} = \{0, 1\}$, $n_+ = (K + Q)L$, где K, Q и L – фиксированные натуральные числа. Параметр L – длина фрагмента, Q – количество начальных данных теста, K – количество основных шагов теста. Пусть $Y_i = (x_{L(i-1)+1}, \dots, x_{Li})$ – фрагмент (слово) длиной L и $i = \overline{1, Q+K}$.

Для $Q + 1 \leq i \leq Q + K$ определим величину

$$A_i = \begin{cases} i, & \text{если не существует } 1 \leq j < i, \text{ что } Y_i = Y_{i-j}, \\ \min\{j: j \geq 1, Y_i = Y_{i-j}\} & \text{в противном случае.} \end{cases} \quad (6.34)$$

Универсальный статистический тест Маурера основан на следующей статистике (мере сжимаемости):

$$F_{\text{uni}} = \frac{1}{K} \sum_{i=Q+1}^{Q+K} \log_2 A_i. \quad (6.35)$$

Теорема 6.2 (теорема Маурера [131]). Если выполняется гипотеза H_0 и $Q \rightarrow \infty$, то для статистик (6.34) и (6.35) моменты имеют вид

$$\mu = \mathbf{E}\{F_{\text{uni}}\} = \mathbf{E}\{\log_2 A_i\} = 2^{-L} \sum_{i=1}^{\infty} (1 - 2^{-L})^{i-1} \log_2 i, \quad (6.36)$$

$$\mathbf{D}\{\log_2 A_i\} = 2^{-L} \sum_{i=1}^{\infty} (1 - 2^{-L})^{i-1} (\log_2 i)^2 - (\mathbf{E}\{F_{\text{uni}}\})^2. \quad (6.37)$$

Дисперсия статистики F_{uni} вычисляется по формуле

$$\sigma^2 = \mathbf{D}\{F_{\text{uni}}\} = c(L, K)^2 \frac{\mathbf{D}\{\log_2 A_i\}}{K}, \quad (6.38)$$

где для $L \geq 3$ $c(L, 2^L)$ близко к 0,8, и если $K \gg 2^L$, то $c(L, K)$ близко к 0,5, 0,6 и 0,65 соответственно для $L = 4$, $L = 8$ и $L = 12$. Для $K \geq 2^L$ достаточна точна аппроксимация [131]:

$$c(L, K) \approx 0,7 - \frac{0,8}{L} + \frac{(4 + 32/L)K^{-3/L}}{15}. \quad (6.39)$$

На практике L рекомендуется выбирать от 6 до 16, $Q \geq 10 \times 2^L$ и K «как можно больше» (например, $K = 1000 \times 2^L$).

Универсальный статистический тест Маурера, таким образом, имеет следующий вид.

1. Выбираются параметры Q , K , L и регистрируется последовательность $x_1, x_2, \dots, x_{(Q+K)L} \in \mathcal{A}$.
2. На основании последовательности $\{x_i\}$ строится статистика F_{uni} .
3. Принимается гипотеза

$$\begin{cases} H_0, & \text{если } t_1 < F_{\text{uni}} < t_2, \\ H_1 & \text{в противном случае,} \end{cases} \quad (6.40)$$

где $t_1 = \mu + \sigma\Phi^{-1}(\varepsilon/2)$ и $t_2 = \mu - \sigma\Phi^{-1}(\varepsilon/2)$; ε – уровень значимости критерия; $\Phi^{-1}(\varepsilon/2)$ – квантиль уровня $\varepsilon/2$ стандартного нормального распределения.

6.13. ТЕСТЫ НА ОСНОВЕ ПРИРАЩЕНИЙ ЭНТРОПИИ

С. Пинкус и Б. Сингер [139] для оценки «степени» случайности последовательности ввели специальную характеристику, основанную на *приращениях энтропии*. Эта характеристика вводится в предположении, что «похожие» фрагменты остаются такими при последовательном увеличении количества членов во фрагментах при некоторых нарушениях гипотезы о РРСП (о равномерном распределении случайной последовательности).

Пусть $x_1, \dots, x_n \in \mathcal{A} = \{0, 1, \dots, N - 1\}$ – наблюдаемая последовательность, $N \geq 2$. Определена следующая гипотеза H_0 : $\{x_t \text{ есть РРСП}\}$ и альтернатива $H_1 = \overline{H_0}$. Обозначим через

$$Y_i(m) = (x_i, x_{i+1}, \dots, x_{i+m-1}) \in \mathcal{A}^m, \quad i = \overline{1, n-m+1},$$

слово из m элементов, начинающееся i -м элементом последовательности. Вычислим относительную частоту встречаемости фрагмента $Y_i(m)$ в последовательности

$$A_m^i = \frac{1}{n-m+1} |\{j: 1 \leq j \leq n-m+1, Y_j(m) = Y_i(m)\}|. \quad (6.41)$$

Пусть

$$F^{(m)} = \frac{1}{n-m+1} \sum_{i=1}^{n-m+1} \log A_m^i \quad (6.42)$$

есть *выборочная энтропия* фрагментов длиной m . Определим *статистику приращения энтропии* для фрагментов длиной m и $m+1$:

$$g(0) = -F^{(1)}, \quad g(m) = F^{(m)} - F^{(m+1)}, \quad m = 1, 2, \dots, n-1. \quad (6.43)$$

Заметим, что малым значениям $g(m)$ соответствует последовательность с «сильной регулярностью». Напротив, большие значения $g(m)$ соответствуют чисто случайным последовательностям.

Асимптотическое поведение статистики $g(m)$ при увеличении объема регистрируемых данных n определяется следующей теоремой [151].

Теорема 6.3. Для фиксированного m при $n \rightarrow \infty$ и выполнении гипотезы H_0 имеет место сходимость к χ^2 -распределению:

$$\mathcal{L} \{2n(\log_2 N - g(m))\} \rightarrow \chi_f^2, \quad f = N^{m+1} - N^m.$$

С помощью этой теоремы построим тест, основанный на приращении энтропии при фиксированном m .

1. Регистрируется последовательность $x_1, x_2, \dots, x_n \in \mathcal{A}$ и задается параметр m .
2. На основании последовательности $\{x_i\}$ вычисляется статистика приращения энтропии $g(m)$ согласно (6.41)–(6.43).
3. Вычисляется статистика $\chi^2 = 2n|\log N - g(m)|$.

4. Выносится решение с помощью решающего правила:

$$\text{принимается } \begin{cases} H_0, & \text{если } P > \varepsilon, \\ H_1, & \text{если } P \leq \varepsilon; \end{cases} \quad (6.44)$$

$$P = 1 - G_f(\chi^2), \quad f = N^{m+1} - N^m,$$

где ε – уровень значимости критерия; $G_f(z)$ – функция χ^2 -распределения с f степенями свободы.

Отметим, что тест (6.44) применим лишь тогда, когда m достаточно мало, так что $n/N^m \rightarrow \infty$. С целью снизить требуемую длину последовательности n рассмотрим «менее жесткую» асимптотику: $n/N^m \rightarrow \lambda$, $0 < \lambda < \infty$.

Для этого построим следующую функцию N аргументов:

$$f(u_1, \dots, u_N) = -u_1 \log \frac{u_1}{u_1 + \dots + u_N} - \dots - u_N \log \frac{u_N}{u_1 + \dots + u_N}.$$

Пусть Π_1, \dots, Π_N – N независимых случайных величин, распределенных по закону Пуассона с параметром λ . Вычислим моменты функции $f(\cdot)$:

$$\mu_n = \frac{1}{N\lambda} \mathbf{E} \{f(\Pi_1, \dots, \Pi_N)\};$$

$$\nu = \frac{1}{N\lambda} \mathbf{Cov} \{f(\Pi_1, \dots, \Pi_N), \Pi_1 + \dots + \Pi_N\};$$

$$\sigma_n^2 = N^m (\mathbf{D} \{f(\Pi_1, \dots, \Pi_N) - N\lambda\nu^2\}).$$

Асимптотическое поведение статистики $g(m)$ при совместном увеличении параметра m и объема регистрируемых данных n определяется следующей теоремой [151].

Теорема 6.4. *Если $n, m \rightarrow \infty$, $n/N^m \rightarrow \lambda > 0$ и верна гипотеза H_0 , то*

$$\mathbf{P} \left\{ n \frac{g(m) - \mu_n}{\sigma_n} < x \right\} \rightarrow \Phi(x),$$

где $\Phi(x)$ – функция распределения стандартного нормального закона.

На основе этой теоремы построен следующий тест.

1. Выбирается параметр m , и регистрируется последовательность $x_1, x_2, \dots, x_n \in \mathcal{A}$.

2. На основе последовательности $\{x_i\}$ строится статистика $g(m)$ согласно (6.43).

3. Выносится решение с помощью решающего правила:

$$\begin{cases} H_0, & \text{если } t_1 < g(m) < t_2, \\ H_1 & \text{в противном случае,} \end{cases} \quad (6.45)$$

где $t_1 = \mu_n + \sigma_n \Phi^{-1}(\varepsilon/2)/n$ и $t_2 = \mu_n - \sigma_n \Phi^{-1}(\varepsilon/2)/n$; ε – уровень значимости критерия; $\Phi^{-1}(\varepsilon/2)$ – квантиль уровня $\varepsilon/2$ стандартного нормального распределения.

6.14. ТЕСТ, ОСНОВАННЫЙ НА АЛГОРИТМЕ СЖАТИЯ ЛЕМПЕЛЯ – ЗИВА

Данный тест основан на следующей идее. Подлежащая тестированию двоичная ($N = 2$) последовательность $x_1, x_2, \dots, x_n \in \mathcal{A}$ подвергается сжатию с помощью алгоритма Лемпеля – Зива [164]. Если результаты сжатия статистически значимо отличаются от теоретических результатов для РРСП, то тестируемая последовательность признается отличной от РРСП.

Алгоритм Лемпеля – Зива состоит из трех шагов.

Шаг 1. Разбиваем исходную последовательность $x_1, \dots, x_n \in \{0, 1\}$ на двоичные слова (цепочки) так, чтобы каждое последующее слово было кратчайшим словом, не встречавшимся ранее.

Шаг 2. Нумеруем все полученные слова последовательно в двоичном алфавите.

Шаг 3. Каждое слово новой («сжатой») двоичной последовательности y_1, y_2, \dots, y_m состоит из префикса и суффикса: префикс – номер предыдущего слова, которое отличается от данного слова лишь последним символом (окончанием); суффикс – окончание.

В данном тесте в качестве статистики используется статистика W_n – число слов, построенных на шагах 1, 2 алгоритма сжатия. Например, если задана исходная двоичная последовательность $\{x_t\}$ длиной $n = 11 : 01100111010$, то после шага 1 получаем $W_n = 6$ двоичных слов: 0; 1; 10; 01; 11; 010. При этом «сжатая» последовательность будет иметь длину $m = 21$:

$$0; \quad 001 \quad 1; \quad 010 \quad 0; \quad 001 \quad 1; \quad 010 \quad 1; \quad 100 \quad 0.$$

Отметим, что при малых значениях n (как в данном числовом примере) длина m сжатой последовательности может оказаться больше n .

Тест основан на следующем асимптотическом свойстве статистики W_n . Если верна гипотеза H_0 и $n \rightarrow \infty$, то W_n распределена асимптотически нормально:

$$\mathcal{L} \left\{ \frac{W_n - \mu_n}{\sigma_n} \right\} \rightarrow \mathcal{N}_1(0, 1), \quad (6.46)$$

причем для математического ожидания μ_n и среднеквадратического отклонения σ_n справедливы асимптотические приближения:

$$\mu_n \approx \frac{n}{\log_2 n}, \quad \sigma_n \approx \sqrt{\frac{0,266n}{(\log_2 n)^3}}. \quad (6.47)$$

Для фиксированных значений n величины μ_n и σ_n можно вычислить экспериментально методом Монте-Карло с помощью некоторого «истинно случайного» генератора. Например, генератора BBS (Blum – Blum – Shub) (см. п. 10.7). Таким способом для $n = 10^6$ было найдено

$$\mu_n \approx 69586,25, \quad \sigma_n \approx 70,448718. \quad (6.48)$$

Тест, основанный на статистике

$$z_n = \frac{W_n - \mu_n}{\sigma_n}, \quad (6.49)$$

имеет вид аналогичный (6.27):

$$\text{принимается } \begin{cases} H_0, & \text{если } P > \varepsilon, \\ H_1, & \text{если } P \leq \varepsilon; \end{cases} \quad (6.50)$$

$$P = 2(1 - \Phi(z_n)),$$

где $\Phi(\cdot)$ – функция распределения стандартного нормального закона; $\varepsilon \in (0, 1)$ – заданный асимптотический уровень значимости.

6.15. ТЕСТ, ОСНОВАННЫЙ НА ЛИНЕЙНОЙ СЛОЖНОСТИ

Линейной сложностью наблюдаемой двоичной последовательности $X_n = (x_1, x_2, \dots, x_n)$ называется минимальный порядок

$$L_n = L(X_n) \in \{1, 2, \dots, n\}$$

линейной рекурренты над \mathbb{F}_2 :

$$x_t = (\theta_1 x_{t-1} + \theta_2 x_{t-2} + \dots + \theta_{L_n} x_{t-L_n}) \bmod 2, \quad t = L_n + 1, L_n + 2, \dots,$$

где $\theta_1, \dots, \theta_{L_n} \in \mathbb{F}_2$ – коэффициенты линейной рекуррентной последовательности, порождающей наблюдаемую последовательность X_n длиной n . Для вычисления линейной сложности L_n может быть использован алгоритм Берлекэмпа – Месси [133]. Оказывается, если верна гипотеза H_0 , то математическое ожидание статистики L_n равно [151]:

$$\mu_n = \mathbf{E}_{H_0} \{L_n\} = \frac{n}{2} + \frac{9 + (-1)^{n+1}}{36} - \frac{1}{2^n} \left(\frac{n}{3} + \frac{2}{9} \right). \quad (6.51)$$

При этом для $n \rightarrow \infty$ дискретное распределение вероятностей статистики $T_n = (-1)^n (L_n - \mu_n) + \frac{2}{9}$ сходится к следующему предельному распределению:

$$\mathbf{P}_{H_0} \{T_n = l\} \rightarrow \begin{cases} 2^{-(2|l|+1)}, & \text{если } l = 0, -1, -2, \dots, \\ 2^{-2l}, & \text{если } l = 1, 2, \dots. \end{cases} \quad (6.52)$$

Из-за быстрого убывания этих вероятностей при $|l| \rightarrow \infty$ удобно разбить область значений T_n на $K = 7$ ячеек:

$$\begin{aligned} \Gamma_1 &= \{T_n \leq -2,5\}, \quad \Gamma_2 = \{-2,5 < T_n \leq -1,5\}, \\ \Gamma_3 &= \{-1,5 < T_n \leq -0,5\}, \quad \Gamma_4 = \{-0,5 < T_n \leq 0,5\}, \\ \Gamma_5 &= \{0,5 < T_n \leq 1,5\}, \quad \Gamma_6 = \{1,5 < T_n \leq 2,5\}, \\ \Gamma_7 &= \{T_n > 2,5\}. \end{aligned} \quad (6.53)$$

Согласно (6.52) и (6.53) находятся теоретические вероятности попадания T_n в эти ячейки:

$$\begin{aligned}\pi_1 &= 0,01047, \quad \pi_2 = 0,03125, \quad \pi_3 = 0,12500, \quad \pi_4 = 0,50000, \\ \pi_5 &= 0,2500, \quad \pi_6 = 0,06250, \quad \pi_7 = 0,02078.\end{aligned}$$

Алгоритм тестирования при этом имеет следующий вид.

1. Исходная n -битовая последовательность X_n разбивается на M блоков по N бит ($500 < N < 5000$):

$$X_n = X_N^{(1)} \| X_N^{(2)} \| \dots \| X_N^{(M)}.$$

2. С помощью алгоритма Берлекэмпа – Месси для i -го блока ($i = \overline{1, M}$) вычисляется линейная сложность $L_N^{(i)} = L(X_N^{(i)})$.

3. Согласно (6.51) находится теоретическое математическое ожидание $\mu_N = E_{H_0} \{ L_N^{(i)} \}$.

4. Для каждого $i = \overline{1, M}$ рассчитывается статистика

$$T_i = (-1)^N (L_N^{(i)} - \mu_N) + \frac{2}{9}.$$

5. Вычисляются частоты попадания $\{T_i\}$ в ячейки $\{\Gamma_k\}$:

$$\nu_k = \sum_{i=1}^M \mathbf{1}_{\Gamma_k}(T_i), \quad k = \overline{1, 7},$$

где $\mathbf{1}_\Gamma(\cdot)$ – индикаторная функция множества Γ ($\nu_1 + \dots + \nu_7 \equiv M$).

6. Рассчитывается χ^2 -статистика:

$$\chi^2 = \sum_{k=1}^7 \frac{(\nu_k - M\pi_k)^2}{M\pi_k} \geq 0.$$

7. Выносится решение по правилу:

принимается $\begin{cases} H_0, & \text{если } P > \varepsilon, \\ H_1, & \text{если } P \leq \varepsilon; \end{cases}$

$$P = 1 - G_f(\chi^2),$$

где $G_f(\cdot)$ – функция χ^2 -распределения с f степенями свободы.

6.16. ТЕСТ НА ОСНОВЕ ЭКСТРЕМАЛЬНОЙ СТАТИСТИКИ СКАЛЯРНОГО ПРОИЗВЕДЕНИЯ

Этот тест предложен в работе [53]. Пусть $x_1, x_2, \dots, x_n \in \mathcal{A} = \{0, 1\}$ – наблюдаемая двоичная последовательность. Определены гипотеза H_0 : $\{x_i\}$ – независимые в совокупности одинаково распределенные случайные величины Бернулли с равномерным распределением $\mathbf{P}\{x_i = 0\} = \mathbf{P}\{x_i = 1\} = \frac{1}{2}$ и альтернатива $H_1 = \overline{H}_0$.

Для заданного $N \in \mathbb{N}$ разобьем последовательность x_1, x_2, \dots, x_n длиной $n = mN$ на m последовательных непересекающихся фрагментов длиной N : $X_1, X_2, \dots, X_m \in \mathcal{A}^N$, где

$$X_i = \begin{pmatrix} x_{(i-1)N+1} \\ x_{(i-1)N+2} \\ \vdots \\ x_{iN} \end{pmatrix} \in \mathcal{A}^N, \quad i \in \{1, \dots, m\}. \quad (6.54)$$

Определим на X_1, X_2, \dots, X_m статистики

$$Y_j = X_1^T X_j = \sum_{k=1}^N x_k x_{(j-1)N+k}, \quad j \in \{2, \dots, m\}. \quad (6.55)$$

Статистика Y_j характеризует «степень похожести» первого и j -го фрагментов. Статистики Y_2, \dots, Y_m при выполнении гипотезы H_0 имеют следующее распределение вероятностей.

Лемма 6.1. *Если верна гипотеза H_0 , то распределение статистики Y_j имеет вид*

$$\mathbf{P}_{H_0}\{Y_j = y\} = 2^{-2N} 3^{N-y} \binom{N}{y},$$

где $y \in \{0, 1, \dots, N\}$, $j \in \{2, \dots, m\}$.

Лемма 6.2. *Если верна гипотеза H_0 , то для $i, j \in \{2, \dots, m\}$, $i \neq j$*

$$\mathbf{E}_{H_0}\{Y_j\} = \frac{N}{4}, \quad \mathbf{D}_{H_0}\{Y_j\} = \frac{3N}{16}, \quad \mathbf{Corr}_{H_0}\{Y_i, Y_j\} = \frac{1}{3}.$$

Лемма 6.3. *Совместное распределение статистик Y_2, \dots, Y_m при H_0 имеет вид*

$$p_m(y_2, \dots, y_m; N) = 2^{-N} \sum_{\substack{l=\max\{y_j\} \\ 2 \leq j \leq m}}^N 2^{-(m-1)l} \binom{N}{l} \prod_{j=2}^m \binom{l}{y_j},$$

$$0 \leq y_2, y_3, \dots, y_m \leq N.$$

На статистиках Y_2, \dots, Y_m построим статистику максимального скалярного произведения:

$$Y_{\max} = \max \{Y_2, \dots, Y_m\}. \quad (6.56)$$

Теорема 6.5. Если верна гипотеза H_0 , то статистика Y_{\max} имеет следующее распределение:

$$\begin{aligned} q_0 &= \mathbf{P}\{Y_{\max} = 0\} = 2^{-N}(1 + 2^{1-m})^N, \\ q_k &= \mathbf{P}\{Y_{\max} = k\} = 2^{-N} \sum_{l=0}^N \binom{N}{l} \left(\sum_{y=0}^k 2^{-l} \binom{l}{y} \right)^{m-1} - \\ &\quad - 2^{-N} \sum_{l=0}^N \binom{N}{l} \left(\sum_{y=0}^{k-1} 2^{-l} \binom{l}{y} \right)^{m-1}, \quad k > 0. \end{aligned}$$

Следствие 6.1. При $m \rightarrow \infty$, если верна гипотеза H_0 , то асимптотическое распределение статистики Y_{\max} имеет вид

$$\lim_{m \rightarrow \infty} q_k = 2^{-N} \binom{N}{k}, \quad k \in \{0, \dots, N\}.$$

Замечание 6.4. Вместо (6.55) для расчета «степени похожести» можно применять следующие формулы:

$$\begin{aligned} Y_j &= \sum_{k=1}^N (1 - x_k) x_{(j-1)N+k}, \quad j \in \{2, \dots, m\}; \\ Y_j &= \sum_{k=1}^N (1 - x_k)(1 - x_{(j-1)N+k}), \quad j \in \{2, \dots, m\}; \\ Y_j &= \sum_{k=1}^N x_k (1 - x_{(j-1)N+k}), \quad j \in \{2, \dots, m\}. \end{aligned}$$

При этом распределение статистик $\{Y_j\}$ при выполнении H_0 не изменится.

При помощи теоремы 6.5 построим тест максимального скалярного произведения на основе χ^2 -статистики.

1. Регистрируется последовательность $x_1, x_2, \dots, x_T \in \mathcal{A}$ длиной $T = LmN = Ln$, где $m, N \in \mathbb{N}$, $m \geq 2$, задаются пользователем, а $L \in \mathbb{N}$ выбирается так, чтобы обеспечить условие применимости χ^2 -критерия.

2. Последовательность длиной T разбивается на L подпоследовательностей длиной $n = mN$, а каждая из этих подпоследовательностей разбивается на m фрагментов длиной N согласно (6.54):

$$X_i^{(l)} = (x_{(l-1)n+(i-1)N+1}, \dots, x_{(l-1)n+iN})^T \in \mathcal{A}, \quad i \in \{1, \dots, m\}, \quad l \in \{1, \dots, L\}.$$

По l -й ($l \in \{1, \dots, L\}$) подпоследовательности $X^{(l)}$ согласно (6.54) вычисляется $(m - 1)$ статистик:

$$Y_j^{(l)} = X_1^{(l)T} X_j^{(l)} = \sum_{k=1}^N x_{(l-1)n+k} x_{(l-1)n+(j-1)N+k}, \quad j \in \{2, \dots, m\}.$$

Для l -й подпоследовательности по статистикам $Y_2^{(l)}, \dots, Y_m^{(l)}$ вычисляется экстремальная статистика $\bar{Y}_{\max}^{(l)}$.

3. По выборке экстремальных статистик $\left\{ Y_{\max}^{(l)} : l \in \{1, \dots, L\} \right\}$ вычисляется χ^2 -статистика:

$$\chi^2 = \sum_{k=0}^N \frac{(f_k - Lq_k)^2}{Lq_k}, \quad f_k = \sum_{l=1}^L \mathbf{1}\{Y_{\max}^{(l)} = k\}.$$

4. Выносим решение с помощью статистического правила:

$$\text{принимается гипотеза } \begin{cases} H_0, & \text{если } P > \varepsilon, \\ H_1, & \text{если } P \leq \varepsilon, \end{cases} \quad (6.57)$$

где $P = 1 - G_N(\chi^2)$ – P -значение; ε – уровень значимости.

Тест максимального скалярного произведения на основе распределения Колмогорова получается из предыдущего теста повторением шагов 1–3 с последующим применением теста Колмогорова к полученной выборке статистики χ^2 . Приведем описание этого теста.

1. Регистрируется последовательность $x_1, x_2, \dots, x_T \in \mathcal{A}$ длиной $T = LSmN = LSn$, где $m, N \in \mathbb{N}$, $m \geq 2$, задаются пользователем, а $L \in \mathbb{N}$ и $S \in \mathbb{N}$ выбираются так, чтобы обеспечить выполнение соответствующих критериев.

2. Последовательность длиной T разбивается на S подпоследовательностей длиной Ln , которые, в свою очередь, разбиваются на L подпоследовательностей длиной $n = mN$, а каждая из этих подпоследовательностей – на m фрагментов длиной N согласно (6.54):

$$X_i^{(l)(s)} = (x_{(s-1)Ln+(l-1)n+(i-1)N+1}, \dots, x_{(s-1)Ln+(l-1)n+iN})^T,$$

где $i \in \{1, \dots, m\}$, $l \in \{1, \dots, L\}$, $s \in \{1, \dots, S\}$.

По l -й ($l \in \{1, \dots, L\}$) подпоследовательности $X^{(l)(s)}$ согласно (6.54) вычисляется $(m - 1)$ статистик:

$$\begin{aligned} Y_j^{(l)(s)} &= X_1^{(l)(s)T} X_j^{(l)(s)} = \\ &= \sum_{k=1}^N x_{(s-1)Ln+(l-1)n+k} x_{(s-1)Ln+(l-1)n+(j-1)N+k}, \quad j \in \{2, \dots, m\}. \end{aligned}$$

Для l -й подпоследовательности по статистикам $Y_2^{(l)(s)}, \dots, Y_m^{(l)(s)}$ вычисляется экстремальная статистика $Y_{\max}^{(l)(s)}$.

3. По выборке экстремальной статистики $\{Y_{\max}^{(l)(s)} : l \in \{1, \dots, L\}\}$ вычисляется χ^2 -статистика:

$$\chi_s^2 = \sum_{k=0}^N \frac{(f_k^{(s)} - Lq_k)^2}{Lq_k}, \quad f_k^{(s)} = \sum_{l=1}^L \delta_{Y_{\max}^{(l)(s)}, k}, \quad s \in \{1, \dots, S\}.$$

4. Над выборкой статистики $\{\chi_s^2 : s \in \{1, \dots, S\}\}$ проведем следующее преобразование: $t_s = G_N(\chi_s^2)$, где $G_N(x)$ – функция χ^2 -распределения с N степенями свободы. Если выполняется гипотеза H_0 , то статистика t_s асимптотически равномерно распределена на отрезке $[0, 1]$. Пусть $t_{(1)}, \dots, t_{(S)}$ – значения t_1, \dots, t_S , упорядоченные по возрастанию.

Статистика

$$D_S = \sqrt{S} \max_{1 \leq j \leq S} \left| \frac{j}{S} - t_{(j)} \right|$$

имеет распределение Колмогорова.

5. Выносим решение с помощью статистического правила:

$$\text{принимается гипотеза } \begin{cases} H_0, & \text{если } D_S < D(1 - \varepsilon), \\ H_1, & \text{если } D_S \geq D(1 - \varepsilon), \end{cases} \quad (6.58)$$

где ε – уровень значимости критерия; $D(1 - \varepsilon)$ – квантиль уровня $(1 - \varepsilon)$ распределения Колмогорова.

6.17. ТЕСТ НА ОСНОВЕ ЭКСТРЕМАЛЬНОЙ СТАТИСТИКИ ДЕЛЬТА-ПРОИЗВЕДЕНИЯ

Этот тест описан в работе [53]. Пусть $x_1, x_2, \dots, x_n \in \mathcal{A} = \{0, 1\}$ – наблюдаемая двоичная последовательность. Определены гипотеза $H_0 = \{\{x_i\} \text{ – независимые в совокупности одинаково распределенные случайные величины Бернулли с равномерным распределением } \mathbf{P}\{x_i = 0\} = \mathbf{P}\{x_i = 1\} = 1/2\}$ и альтернатива $H_1 = \overline{H_0}$.

Для заданного $N \in \mathbb{N}$ разобьем последовательность x_1, x_2, \dots, x_n длиной $n = mN$ на m последовательных непересекающихся фрагментов длиной N : $X_1, X_2, \dots, X_m \in \mathcal{A}^N$, где

$$X_i = \begin{pmatrix} x_{(i-1)N+1} \\ x_{(i-1)N+2} \\ \vdots \\ x_{iN} \end{pmatrix} \in \mathcal{A}^N, \quad i \in \{1, \dots, m\}. \quad (6.59)$$

Определим на X_1, X_2, \dots, X_m статистики

$$Y_j^* = \sum_{k=1}^N \mathbf{1}\{x_k = x_{(j-1)N+k}\}, \quad j \in \{2, \dots, m\}. \quad (6.60)$$

Статистика Y_j^* характеризует «степень похожести» первого и j -го фрагментов. Статистики Y_2^*, \dots, Y_m^* при выполнении гипотезы H_0 имеют следующее распределение.

Лемма 6.4. *Если верна гипотеза H_0 , то статистики $\{Y_j^*\}$ распределены по биномиальному закону распределения вероятностей:*

$$\mathbf{P}_{H_0}\{Y_j^* = y\} = 2^{-N} \binom{N}{y}, \quad 0 \leq y \leq N, \quad j \in \{2, \dots, m\}.$$

Теорема 6.6. *Совместное распределение Y_2^*, \dots, Y_m^* при H_0 имеет вид*

$$p_m^*(y_2, \dots, y_m; N) = \prod_{j=2}^m 2^{-N} \binom{N}{y_j}, \quad 0 \leq y_2, \dots, y_m \leq N.$$

Следствие 6.2. *Если верна гипотеза H_0 , то статистики Y_2^*, \dots, Y_m^* независимы в совокупности.*

На статистиках Y_2^*, \dots, Y_m^* построим статистику максимального дельта-произведения:

$$Y_{\max}^* = \max\{Y_2^*, \dots, Y_m^*\}.$$

Следствие 6.3. *Функция распределения статистики Y_{\max}^* при выполнении гипотезы H_0 имеет вид*

$$F_{Y_{\max}^*}(k) = \left(\sum_{y=0}^k 2^{-N} \binom{N}{y} \right)^{m-1}, \quad k \in \{0, \dots, N\}.$$

Следствие 6.4. *При $m \rightarrow \infty$, если верна гипотеза H_0 , то асимптотическое распределение статистики Y_{\max}^* имеет вид*

$$\lim_{m \rightarrow \infty} q_k^* = \begin{cases} 0, & \text{если } k < N, \\ 1, & \text{если } k = N. \end{cases}$$

Замечание 6.5. Вместо (6.60) можно применять следующую формулу:

$$Y_j^* = \sum_{k=1}^N \mathbf{1}\{x_k \neq x_{(j-1)N+k}\}, \quad j \in \{2, \dots, m\}.$$

При этом распределение статистик $\{Y_j^*\}$ не изменится.

Пользуясь теоремой 6.6, построим *тест максимального дельта-произведения* на основе χ^2 -статистики.

1. Регистрируется последовательность $x_1, x_2, \dots, x_T \in \mathcal{A}^T$, $\mathcal{A} = \{0, 1\}$, длиной $T = LmN = Ln$, где $m, N \in \mathbb{N}$, $m \geq 2$, задаются пользователем, а $L \in \mathbb{N}$ выбирается так, чтобы обеспечить выполнение условий применимости χ^2 -критерия.

2. Последовательность длиной T разбивается на L подпоследовательностей длиной $n = m \times N$, а каждая из этих подпоследовательностей – на m фрагментов длиной N согласно (6.54):

$$X_i^{(l)} = (x_{(l-1)n+(i-1)N+2}, \dots, x_{(l-1)n+iN})^T \in \mathcal{A}^N, \quad i \in \{1, \dots, m\}, \quad l \in \{1, \dots, L\}.$$

По l -й ($l \in \{1, \dots, L\}$) подпоследовательности $X^{(l)}$ согласно (6.54) вычисляется $(m - 1)$ статистик:

$$Y_j^{*(l)} = \sum_{k=1}^N \mathbf{1}\{x_{(l-1)n+k} = x_{(l-1)n+(j-1)N+k}\}, \quad j \in \{2, \dots, m\}.$$

Для l -й подпоследовательности по статистикам $Y_2^{*(l)}, \dots, Y_m^{*(l)}$ вычисляется экстремальная статистика $Y_{\max}^{*(l)}$.

3. По выборке экстремальной статистики $\{Y_{\max}^{*(l)}: l \in \{1, \dots, L\}\}$ вычисляется χ^2 -статистика:

$$\chi^2 = \sum_{k=0}^N \frac{(f_k - Lq_k)^2}{Lq_k}, \quad f_k = \sum_{l=1}^L \mathbf{1}\{Y_{\max}^{*(l)} = k\}.$$

4. Выносим решение с помощью статистического правила (6.57).

Тест максимального дельта-произведения на основе распределения Колмогорова получается из предыдущего теста повторением шагов 1–3 с последующим применением теста Колмогорова к полученной выборке статистики χ^2 аналогично тому, как это сделано в п. 6.16.

6.18. ОБ АЛГОРИТМИЧЕСКОМ ОПРЕДЕЛЕНИИ СЛУЧАЙНОСТИ

Все статистические тесты РРСП, представленные в предыдущих пунктах, базируются на вероятностной модели случайности [25]. Однако в последние годы интенсивно развивается также алгоритмическая концепция случайности, берущая начало от работ А. Н. Колмогорова и В. А. Успенского, П. Мартин-Лёфа, Г. Дж. Чайтина, К. Р. Шнорра. Как отмечается в [25, с. 18],

«...хотя алгоритмы и случайность могут показаться диаметрально противоположными, но это только на первый взгляд. Рассматривая наши представления о случайности и об алгоритмах более пристально, можно найти связи между ними... Алгоритмы могут быть использованы для определения случайности, а случайность – для построения нового класса алгоритмов...». Имея в виду потенциальную возможность построения нового семейства тестов, базирующихся на алгоритмическом определении случайности, изложим основные понятия, связанные с этим определением.

Примем обозначения: $V = \{0, 1\}$ – двоичный алфавит; Ξ – множество всех двоичных цепочек (strings) $x = (x_1, \dots, x_n)$ произвольной конечной длины $n = |x| \in \mathbb{N}$; $x \in V^{|x|}$; $\Omega = V^\infty$ – множество всех бесконечных двоичных последовательностей; $\mu(\omega)$, $\omega \in \Omega$ – равномерная бернуlliева мера на Ω , $\mu(\Omega) = 1$; $R \in \Omega$ – подлежащее построению подмножество «случайных в некотором разумном неформальном смысле» последовательностей.

Случайные последовательности обладают тремя фундаментальными свойствами: типичность, хаотичность, устойчивость частот, каждое из них может быть взято за основу некоторого определения случайности.

Класс $T \in \Omega$ «типических последовательностей» введен П. Мартин-Лёфом, так что $T \in R$. Свойство типичности означает принадлежность последовательности к каждому разумному их большинству. На первый взгляд кажется, что в качестве T можно взять пересечение всех подмножеств множества Ω , имеющих меру 1. Однако, как показал Мартин-Лёф, такое пересечение есть пустое множество \emptyset .

Чтобы построить класс T , понадобится ряд вспомогательных понятий. Каждой конечной цепочке $X \in \Xi$ длиной $n = |x| < +\infty$ поставим в соответствие шар

$$\Gamma_x = \{(x \parallel x_{n+1}, x_{n+2}, \dots) \in \Omega : x_{n+1}, x_{n+2}, \dots \in V\} \in \Omega,$$

состоящий из всевозможных бесконечных продолжений цепочки x , причем

$$\mu(\Gamma_x) = \frac{1}{2^n}.$$

Известно, что $\mu(M) = 0$, т. е. подмножество $M \subset \Omega$ является *пренебрежимым* тогда и только тогда, когда для любого рационального $\varepsilon > 0$ существует такая последовательность конечных цепочек $X_k \in \Xi$, $k \in \mathbb{N}$, что соответствующая последовательность шаров удовлетворяет двум условиям:

- 1) $M \subset \bigcup_k \Gamma_{X_k}$;
- 2) $\sum_k \mu(\Gamma_{X_k}) < \varepsilon$

Если к тому же последовательность цепочек X_k , $k \in \mathbb{N}$, является вычисляемой, т. е. существует алгоритм для вычисления X_k по заданному k ,

реализуемый конечной программой $\langle X_k \rangle$, то принято говорить, что множество имеет эффективно меру 0:

$$\mu(L)_{\text{eff}} = 0,$$

или является эффективно пренебрежимым. При этом множество $L \subset \Omega$ имеет эффективно меру единица:

$$\mu(L)_{\text{eff}} = 1,$$

если для его дополнения имеем $\mu(\Omega \setminus L)_{\text{eff}} = 0$.

Теорема 6.7 (теорема Мартин-Лёфа). Пересечение всех множеств эффективной меры 1 не только не пусто, но и само имеет эффективную меру 1.

Указанное в этой теореме пересечение T является наименьшим множеством эффективной меры 1 и называется *конструктивным носителем* (constructive support) меры. Это множество и принимается в качестве класса всех типических последовательностей, $\mu(L)_{\text{eff}} = 1$.

Последовательность $X \in T$, принадлежащая конструктивному носителю меры, называется *случайной по Мартин-Лёфу*.

Заметим, что многие предельные теоремы теории вероятностей (например, закон повторного логарифма), в которых указывается сходимость «почти наверное», на самом деле верны для класса T , т. е. по эффективной мере 1.

Класс $C \subset \Omega$ хаотических последовательностей введен А. Н. Колмогоровым ($R \subset C$): всякая случайная последовательность – хаотическая в том смысле, что для нее отсутствует какой-либо простой закон, управляющий чередованием ее членов. Таким образом, хаотичность означает сложность строения последовательности.

Пусть $k = K(x) : \Xi \rightarrow \mathbb{N}$ – мера сложности конечной двоичной цепочки x . В качестве меры сложности $K(\cdot)$ целесообразно, оказывается, использовать монотонную энтропию $L(\cdot)$, предложенную А. К. Звонкиным и К. Шнорром [25]. Алгоритм вычисления монотонной энтропии – довольно сложный и использует язык программирования для одноголовочной машины Тьюринга.

Последовательность $(x_1, x_2, \dots, x_n) \in \Omega$ принадлежит классу $C \subset \Omega$ хаотических последовательностей тогда и только тогда, когда для любого $n \in \mathbb{N}$

$$K(x_1, x_2, \dots, x_n) \geq n - c,$$

где $c \geq 0$ – некоторая константа, не зависящая от n , т. е. когда энтропия растет достаточно быстро при $n \rightarrow \infty$. Последовательность $X \in C$ при этом называется *случайной по Колмогорову*.

Теорема 6.8 (теорема Левина – Шнорра). Классы C и T совпадают: $T = C$.

Классом случайных последовательностей в связи с этим называют $R = T = C$.

Класс $S \subset \Omega$ стохастических последовательностей содержит такие последовательности $X \in \Omega$, для которых выполняется *свойство устойчивости частот*:

$$\frac{1}{s} \sum_{t=1}^s x_t \xrightarrow{s \rightarrow \infty} \frac{1}{2};$$

причем это свойство сохраняется и для подпоследовательности $(x_{\gamma_1}, x_{\gamma_2}, \dots) \in \Omega$ при «любом допустимом» выборе индексов $\gamma_1, \gamma_2, \dots$. Существует два понятия допустимости индексных последовательностей $\gamma_1, \gamma_2, \dots$

Класс стохастических по Чёрчу последовательностей $CS \subset \Omega$ допускает следующий способ построения индексных последовательностей:

$$\gamma_i = f(\gamma_1, \dots, \gamma_{i-1}, x_1, \dots, x_i), \quad \gamma_1 < \gamma_2 < \gamma_3 < \dots$$

Класс стохастических по Колмогорову последовательностей $KS \subset \Omega$ отличается тем, что условие упорядоченности индексов $\{\gamma_i\}$ снимается. При этом

$$R \subset KS \subset CS, \quad KS \neq CS.$$

Открытой пока остается проблема: $KS \stackrel{?}{=} R$.

В заключение кратко остановимся на алгоритмическом определении случайности для конечных цепочек. Как отмечает А. Н. Колмогоров, «кажется естественным назвать цепочку $x \in \Xi$ случайной, если она не может быть записана в более сжатом виде, т. е. если кратчайшая программа ее порождения $\langle x \rangle$ столь же длинна, как она сама. Правильный вопрос состоит не в том, является ли данная цепочка случайной, а в том, насколько она случайна» [25, с. 18]. Для этого вводится понятие «дефект случайности».

Пусть $M \subset \Xi$ – произвольное фиксированное множество конечных двоичных цепочек и $x \in M$. Тогда дефект случайности элемента x относительно M есть

$$d = d(x | M) = \log_2 |M| - H\{x | M\},$$

где $H\{x | M\} \geq 0$ – условная сложность описания элемента x . Иначе говоря, d – разность в длине между двумя описаниями элемента x средствами множества M : между стандартным (указать номер элемента в алфавите M) и кратчайшим описаниями. При этом элемент $x \in M$ называется δ -случайным (δ -стохастическим) для некоторого фиксированного $\delta > 0$, если $d(x | M) \leq \delta$.

6.19. ТЕСТ ВЫЯВЛЕНИЯ МАРКОВСКОЙ ЗАВИСИМОСТИ

Как отмечалось в п. 6.1, представим ряд тестов, ориентированных на специальный вид альтернативы H_1 . В этом пункте мы рассмотрим случай, когда альтернатива H_1 соответствует марковская модель – цепь Маркова первого порядка, – рассмотренная в п. 5.3.

Итак, имеется нулевая гипотеза $H_0 = \{x_t \text{ есть РРСП}\}$ и альтернатива $H_1 = \{x_t \text{ есть однородная ЦМ с некоторой невырожденной матрицей вероятностей одношаговых переходов } P = (p_{ij})\}$. Переформулируем эти гипотезы в параметрическом виде:

$$H_0 : P = P_0 = \frac{1}{N} \mathbb{I}_{N \times N};$$

$$H_1 = \bar{H}_0 : P \neq P_0,$$

где $\mathbb{I}_{N \times N} - N \times N$ -матрица, все элементы которой равны единице.

Тест отношения правдоподобия для проверки гипотез имеет вид [16, 69]:

$$\text{принимается } \begin{cases} H_0, & \text{если } \chi_n^2 < \Delta, \\ H_1, & \text{если } \chi_n^2 \geq \Delta, \end{cases}$$

где статистика

$$\chi_n^2 = N \sum_{i, j \in A} \frac{(\nu_{ij} - \nu_{i \cdot} N^{-1})^2}{\nu_{i \cdot}};$$

$$\nu_{i \cdot} = \sum_{j \in A} \nu_{ij}, \quad \nu_{ij} = \sum_{t=1}^{n-1} \delta_{x_t, i} \delta_{x_{t+1}, j} -$$

частота встречаемости биграммы (i, j) в наблюдаемой реализации $X_n = (x_1, \dots, x_n)' \in A^n$, $\Delta = G_{N(N-1)}^{-1}(1 - \varepsilon)$ – квантиль уровня $1 - \varepsilon$ стандартного хи-квадрат распределения вероятностей с $N(N - 1)$ степенями свободы. Асимптотический размер этого теста совпадает с заданным уровнем значимости $\varepsilon \in (0, 1)$;

$$\lim_{n \rightarrow \infty} P_{H_0} \{\chi_n^2 \geq \Delta\} = \varepsilon;$$

асимптотические оценки мощности этого теста имеются в [16, 69].

6.20. ТЕСТ НА ОСНОВЕ МОДЕЛИ ДЖЕКОБСА – ЛЬЮИСА

Рассмотрим случай, когда альтернатива H_1 соответствует малопараметрическая марковская модель Джекобса – Льюиса, представленная в п. 5.5.

Теорема 6.9. [45] Пусть наблюдается реализация $X_n = (x_1, \dots, x_n)'$ длительности n дискретного временного ряда, соответствующего модели Джекобса – Льюиса (5.23)–(5.25). Тогда логарифмическая функция правдоподобия имеет вид

$$l(\pi, \lambda, \rho) = \sum_{t=1}^s \pi_{x_t} + \sum_{t=s+1}^n \ln \left((1 - \rho)\pi_{x_t} + \rho \sum_{j=1}^s \lambda_j \mathbf{1}\{i_{x_{t-j}} = i_{x_t}\} \right).$$

Примем обозначения: $P(\pi, \lambda, \rho)$ – матрица, элементы которой вычислены по формуле (5.25); \hat{P} – эмпирическая матрица вероятностей переходов, вычисленная по наблюдаемой реализации $X_n = (x_1, \dots, x_n)'$ длительности n ; $S(P)$ – сумма квадратов элементов матрицы P ; $F(P)$ – сумма квадратов элементов матрицы P , для которых выполнено

$$\{i_1 = \dots = i_s = i_{s+1}\} \text{ или } \{i_0 \neq i_{s+1}, \dots, i_s \neq i_{s+1}\}.$$

Теорема 6.10. [45] Если имеет место модель Джекобса – Льюиса (5.23–5.25) и $\rho \neq 1$, то при увеличении длительности наблюдаемого временного ряда $n \rightarrow \infty$ состоятельными оценками для параметров модели являются статистики $\tilde{\pi}, \tilde{\lambda}, \tilde{\rho}$:

$$\tilde{\pi}_i = \sum_{t=1}^n \mathbf{1}\{x_t = i\} / n, \quad i \in A; \quad \tilde{\rho} = \arg \min_{\rho \in [0, 1]} F(\hat{P} - P(\tilde{\pi}, \lambda, \rho));$$

$$\tilde{\lambda} = \arg \min S(\hat{P} - P(\tilde{\pi}, \lambda, \tilde{\rho})),$$

причем получены явные формулы для вычисления $\tilde{\lambda}, \tilde{\rho}$.

Эти оценки используются в качестве начального приближения при итерационном вычислении оценки максимального правдоподобия (ОМП) $(\hat{\pi}, \hat{\lambda}, \hat{\rho})$.

Согласно (5.23–5.25) гипотеза $H_0 = \{\{x_t\} \text{ есть РРСП}\}$ допускает эквивалентное представление: $H_0 = \{\rho = 0, \pi_i = N^{-1}, i \in A\}$. Тест проверки гипотез $H_0, H_1 = \bar{H}_0$, основанный на статистике обобщенного отношения правдоподобия $\lambda_n = 2(l(\hat{\pi}, \hat{\lambda}, \hat{\rho}) + n \ln N)$ и имеющий асимптотический (при $n \rightarrow \infty$) размер $\varepsilon \in (0, 1)$, имеет вид

$$d = d(X_n) = \{0, \lambda_n < \Delta_\varepsilon; 1, \lambda_n \geq \Delta_\varepsilon; \},$$

где Δ_ε – квантиль уровня ε хи-квадрат распределения с N степенями свободы.

6.21. ТЕСТ НА ОСНОВЕ МТД-МОДЕЛИ

Рассмотрим случай, когда альтернативе H_1 соответствует малопараметрическая марковская МТД-модель Рафтери, представленная в п. 5.6.

Теорема 6.11. [45] Пусть наблюдается реализация $X_n = (x_1, \dots, x_n)'$ длительности n дискретного временного ряда, соответствующего МТД-модели (5.26). Тогда логарифмическая функция правдоподобия параметров λ, Q имеет вид

$$l(\lambda, Q) = \sum_{t=s+1}^n \ln \left(\sum_{j=1}^s \lambda_j q_{x_{t-s+j-1}, x_t} \right).$$

Аналогично предыдущему пункту для проверки гипотез $H_0 = \{\{x_t\}$ есть РРСП}, $H_1 = \bar{H}_0$ будем использовать тест, основанный на статистике обобщенного отношения правдоподобия

$$\lambda_n = 2 \left(l(\hat{Q}, \hat{\lambda}) + (n - s) \ln N \right),$$

где $\hat{Q}, \hat{\lambda}$ – ОМП параметров модели. Заметим, что согласно (5.26) гипотеза H_0 допускает эквивалентное представление

$$H_0 = \{q_{ki} = N^{-1}, k, i \in A\}.$$

Тест проверки гипотез $H_0, H_1 = \bar{H}_0$

$$d = d(X_T) = \{0, \lambda_T < \Delta_\varepsilon; 1, \lambda_T \geq \Delta_\varepsilon\},$$

где Δ_ε – квантиль уровня ε хи-квадрат распределения с $N(N - 1)$ степенями свободы, имеет асимптотический (при $n \rightarrow \infty$) размер $\varepsilon \in (0, 1)$.

Основная сложность этого теста состоит в вычислении ОМП. Для этого в [76] предложен итерационный алгоритм вычисления ОМП. Однако предложенные там же [76] начальные значения ухудшают работу алгоритма, так как с увеличением n число итераций алгоритма, необходимых для достижения результата, не уменьшается. Поэтому в качестве начальных значений алгоритма предлагается [56] использовать статистики $\tilde{Q} = (\tilde{q}_{ki}), k, i \in A, \tilde{\lambda} = (\tilde{\lambda}_1, \dots, \tilde{\lambda}_s)'$:

$$\begin{aligned} \tilde{q}_{ki} &= \begin{cases} \sum_{j=1}^s \hat{\pi}_{ki}(j) / \hat{\pi}_k - (s - 1) \hat{\pi}_i, & \text{если } \hat{\pi}_k > 0, \\ 1/N & \text{в противном случае,} \end{cases} \\ \tilde{\lambda}_j &= \sum_{i, k \in A} z_{ki}(s - j) d_{ki} / \sum_{i, k \in A} d_{ki}^2, \quad j = 1, \dots, s, \end{aligned}$$

где

$$\hat{\pi}_i = \sum_{t=s+1}^{n-s+1} \mathbf{1}\{x_t = i\} / (n - 2s + 1),$$

$$\hat{\pi}_{ki}(j) = \sum_{t=s+j+1}^{n-s+j+1} \mathbf{1}\{x_{t-j} = k\} \mathbf{1}\{x_t = i\} / (n - 2s + 1),$$

$z_{ki}(j) = \hat{\pi}_{ki}(j) / \hat{\pi}_k - \hat{\pi}_i$, $d_{ki} = \hat{q}_{ki} - \hat{\pi}_i$, $i, k \in A$, $j = 1, \dots, s$, причем матрица \tilde{Q} – стохастическая. Отметим, что вычислительная сложность этих начальных приближений совпадает с вычислительной сложностью начальных значений из [76], однако предложенные статистики являются более точными начальными значениями, что подтверждается многочисленными компьютерными экспериментами.

6.22. ТЕСТ НА ОСНОВЕ ЦЕПЕЙ МАРКОВА С ЧАСТИЧНЫМИ СВЯЗЯМИ

Рассмотрим случай, когда альтернативе H_1 соответствует малопараметрическая модель ЦМ (s, r) – цепь Маркова порядка s с r частичными связями, представленная в п. 5.7.

Примем обозначения: $J_s = (j_1, \dots, j_s) = (J_{s-1}, j_s) \in A^s$ – мультииндекс s -го порядка; $S_t(X_n; M_r) = (x_{t+m_1-1}, \dots, x_{t+m_r-1}) \in A^r$ – функция $A^n \times M \rightarrow A^r$, которую условимся называть селектором r -го порядка с параметрами $M_r \in M$ и $t \in \{1, \dots, n - s + 1\}$; $\Pi_{K_s} = \mathbf{P}\{X_s = K_s\}$ – начальное s -мерное распределение вероятностей ЦМ (s, r) ; $\nu_{r+1}(J_{r+1}; M_r) = \sum_{t=1}^{n-s} \mathbf{1}\{S_t(X_n; M_{r+1}) = J_{r+1}\}$ – частота $(r+1)$ -грамммы $J_{r+1} \in A^{r+1}$ для шаблона $M_{r+1} = (M_r, s+1)$, удовлетворяющая условию нормировки:

$$\sum_{J_{r+1} \in A^{r+1}} \nu_{r+1}(J_{r+1}; M_r) \equiv n-s.$$

Положим, что если вместо какого-то индекса стоит точка, то это означает суммирование по всем возможным значениям этого индекса:

$$\nu_{r+1}(J_r; M_r) = \sum_{j_{r+1} \in A} \nu_{r+1}(J_{r+1}; M_r), \quad \nu_{r+1}(\cdot; j_{r+1}) = \sum_{J_r \in A^r} \nu_{r+1}(J_{r+1}; M_r) -$$

«накопленные» статистики, отличающиеся не более чем на s от частоты r -грамммы $J_r \in A^r$ и частоты символа $j_{r+1} \in A$ соответственно.

По наблюдаемой реализации X_n на основе подстановочного принципа («plug-in») построим информационный функционал $\hat{I}_{r+1}(M_r)$, т. е. выборочную оценку количества информации по Шеннону, содержащейся в r -грамме $S_t(X_n, M_r)$ о будущем символе $x_{t+s} \in A$.

Теорема 6.12. [54] ОМП \hat{M}_r , $\hat{Q} = (\hat{q}_{J_{r+1}})$, $J_{r+1} \in A^{r+1}$, параметров M_r^0 , Q^0 модели (5.28) определяются следующими соотношениями:

$$\hat{M}_r = \arg \max_{M_r \in M} \hat{I}_{r+1}(M_r),$$

$$\hat{q}_{J_{r+1}}(M_r) = \begin{cases} \nu_{r+1}(J_{r+1}; \hat{M}_r) / \nu_{r+1}(J_r; \hat{M}_r), & \text{если } \nu_{r+1}(J_r; \hat{M}_r) > 0, \\ 1/N, & \text{если } \nu_{r+1}(J_r; \hat{M}_r) = 0. \end{cases}$$

Обозначим: $\Pi_{K_s}^*$, $K_s \in A^s$, – стационарное распределение вероятностей ЦМ(s, r);

$$\mu_{r+1}(J_{r+1}; M_r, M_r^0) = \sum_{K_{s+1} \in A^{s+1}} \mathbf{1}\{S_1(K_{s+1}; M_{r+1}) = J_{r+1}\} \Pi_{K_s}^* p_{K_{s+1}},$$

$$J_{r+1} \in A^{r+1}.$$

Теорема 6.13. [54] Если ЦМ (s, r), определяемая (5.28), стационарна и шаблон $M_r^0 \in M$ удовлетворяет условию идентифицируемости, то при $n \rightarrow \infty$ ОМП \hat{M}_r , \hat{Q} состоятельны:

$$\hat{M}_r \xrightarrow{P} M_r^0, \quad \hat{Q} \xrightarrow{L_2} Q^0,$$

причем справедливо асимптотическое разложение для вариации оценки \hat{Q} :

$$\Delta_n^2 = \mathbf{E} \left\{ \left\| \hat{Q} - Q^0 \right\|^2 \right\} = \frac{1}{n-s} \sum_{J_{r+1} \in A^{r+1}} \frac{\left(1 - q_{J_{r+1}}^0\right) q_{J_{r+1}}^0}{\mu_{r+1}(J_r; M_r^0, M_r^0)} + o\left(\frac{1}{n}\right).$$

С помощью этих оценок построен критерий статистической проверки гипотез $H_0 = \{Q^0 = Q_0\}$ против альтернативы общего вида $H_1 = \tilde{H}_0$, где $Q_0 = (q_{0J_{r+1}})$ – некоторая заданная стохастическая матрица. Решающее правило заданного асимптотического размера $\varepsilon \in (0, 1)$ имеет вид

$$d(X_n) = \{0, \text{ если } \rho \leq \Delta; 1, \text{ если } \rho > \Delta\},$$

где

$$\rho = \sum_{J_{r+1}: q_{0J_{r+1}} > 0} \nu_{r+1}(J_r; \hat{M}_r) \left(\hat{q}_{J_{r+1}} - q_{0J_{r+1}} \right)^2 / q_{0J_{r+1}}, \quad \Delta = G_L^{-1}(1 - \varepsilon).$$

6.23. ЗАДАНИЯ

1. Для $N = 2$; $t_k(x_1, \dots, x_n) = x_k$, $k = \overline{1, n}$; $m = n$; $L = 2^n$; $U = \mathcal{A}^n$; $\varepsilon = 0,01$ реализовать на компьютере тест (6.4)–(6.6) n -мерной дискретной равномерности двоичной последовательности $\{x_t\}$. Методом статистического моделирования на компьютере для заданных значений n оценить вероятности ошибок первого и второго рода. Оценить сложность алгоритма тестирования аналитически и проверить согласованность полученной оценки с реальным быстродействием алгоритма. В качестве тестируемых последовательностей использовать File00 – File06 из Архива дискретных последовательностей (АДП) по указанию преподавателя (прил. 1).

2. Реализовать на компьютере тест (6.4)–(6.6) при $N = 2$; $n = 8, 16, 32, 64$; $m = 8$; $U = \mathcal{A}^8$ (суммирование выполняется по mod 2);

$$t_1(x_1, \dots, x_n) = x_1 \oplus x_2 \oplus \dots \oplus x_n;$$

$$t_2(x_1, \dots, x_n) = \sum_{1 \leq i_1 < i_2 \leq n} x_{i_1} x_{i_2}, \dots;$$

$$t_8(x_1, \dots, x_n) = \sum_{1 \leq i_1 < i_2 < \dots < i_8 \leq n} x_{i_1} x_{i_2} \dots x_{i_8}.$$

Методом статистического моделирования на компьютере оценить вероятности ошибок первого и второго рода. Оценить сложность алгоритма тестирования аналитически и проверить согласованность полученной оценки с реальным быстродействием алгоритма. Для компьютерных экспериментов использовать те же последовательности, что и в задании 1.

3. В задании 2 исследовать влияние выбора m и функций $\{t_k(\cdot)\}$ на мощность теста и быстродействие алгоритма тестирования.

4. Аналогично заданию 2 рассмотреть еще один универсальный способ задания статистик $\{t_k(\cdot)\}$:

$$t_k(x_1, \dots, x_n) = \mathbf{1}_{\Gamma_{k-1}}(x_1, \dots, x_n), \quad k = \overline{1, m},$$

где $\mathbf{1}_\Gamma(\cdot)$ – индикаторная функция множества $\Gamma \subseteq \mathcal{A}^n$; $\Gamma_0, \dots, \Gamma_{m-1} \subset \mathcal{A}^n$ – разбиение пространства \mathcal{A}^n на m «ячеек»:

$$\mathcal{A}^n = \bigcup_{k=0}^{m-1} \Gamma_k, \quad \Gamma_k \cap \Gamma_l = \emptyset, \quad l \neq k.$$

Пример такого разбиения при $m = 8, n = 16$:

$$\Gamma_{i_1+2i_2+4i_3} = \{(i_1, i_2, i_3, j_1, \dots, j_{13}) : j_1, \dots, j_{13} \in \mathcal{A}\}, \quad i_1, i_2, i_3 \in \mathcal{A}.$$

При таком разбиении все ячейки имеют одинаковую мощность:

$$|\Gamma_0| = |\Gamma_1| = \dots = |\Gamma_7| = 2^{13}.$$

5. Выполнить задание 1 для $N > 2$, $m = n$, $t_k(x_1, \dots, x_n) = x_k$, $k = \overline{1, n}$, $U = \mathcal{A}^n$, $L = N^n$. Для компьютерных экспериментов использовать последовательности File07, File08 из АДП при $N = 8$.

6. Разработать модифицированный алгоритм тестирования n -мерной равномерности на основе подхода, указанного в замечании 6.3. Выполнить с модифицированным алгоритмом эксперименты, указанные в задании 1. Провести сравнительный анализ модифицированного и базового алгоритма (6.4)–(6.6) по быстродействию и мощности.

7. Выполнить задания 1, 5 для теста n -серий из п. 6.3.

8. Оптимизировать по быстродействию программу, реализующую тест n -серий.

9. Выполнить задания 1, 5 для теста интервалов при заданных значениях α , β .

10. Модифицировать тест интервалов так, чтобы обеспечить выполнение условия, указанного в замечании 6.2. Для этого вместо статистики (6.8) рассмотреть «усеченную статистику»:

$$\tilde{t}(x_1, x_2, \dots) = \begin{cases} t(x_1, x_2, \dots), & \text{если } t(x_1, x_2, \dots) < t_+, \\ t_+, & \text{если } t(x_1, x_2, \dots) \geq t_+, \end{cases} \quad (6.61)$$

где $t_+ \in \mathbb{N}$ – заданная верхняя граница такая, что $M \times p \times (1 - p)^{t_+-1} > 5$.

11. Исследовать вопрос выбора параметров α , β с целью увеличить мощность теста интервалов в рамках задания 1 для класса альтернатив H_{1n} , определяемых однородной цепью Маркова (прил. 1).

12. Выполнить задание 11 для класса альтернатив H_{1n} , определяемых двоичной авторегрессионной последовательностью BAR(p) (прил. 1).

13. Методом статистического моделирования модифицированного теста, основанного на статистике (6.61), исследовать зависимость мощности теста от «параметра усечения» t_+ на основе данных задания 11.

14. Для обобщенного покер-теста выполнить задания 1, 5 с помощью классического покер-теста. Исследовать влияние n на вычислительную сложность и мощность теста.

15. Выполнить задания 1, 5 с помощью обобщенного покер-теста. Исследовать влияние n на вычислительную сложность и мощность теста.

16. Осуществить модификацию классического покер-теста и обобщенного покер-теста для двоичных последовательностей ($N = 2$), переходя от битов к фрагментам последовательности – полубайтам ($N' = 16$), байтам ($N'' = 256$), n -кам ($N''' = 2^n$). Для проверки n -мерной равномерности третий вариант более предпочтителен.

17. Провести сравнительный анализ классического и обобщенного покер-теста по вычислительной сложности и мощности.

18. Выполнить задания 1, 5 с помощью теста «собирателя купонов». Аналитически и методом компьютерного моделирования исследовать влияние параметра укрупнения r на мощность и вычислительную сложность теста.
19. Аналитически и методом компьютерного моделирования оценить математическое ожидание и дисперсию статистики (6.10) при истинности гипотезы H_{0n} . Использовать эти результаты для уточнения оценки вычислительной сложности теста.
20. Разработать обобщение теста «собирателя купонов» (в целях снижения его вычислительной сложности), в котором осуществляется сбор неполного комплекта из $N'_0 \leq N'$ купонов.
21. Выполнить задания 1, 5 с помощью обобщенного теста «собирателя купонов». Исследовать влияние параметра N'_0 на мощность теста и вычислительную сложность реализующего его алгоритма.
22. Выполнить задание 1 с помощью теста перестановок.
23. В условиях задания 22 аналитически и методом компьютерного моделирования исследовать влияние параметра n' на мощность и вычислительную сложность теста.
24. Выполнить задание 22 для обобщенного теста перестановок, описанного в работе [49] и учитывающего раздельно комбинации с совпадающими компонентами. Сравнить этот тест по мощности и вычислительной сложности с тестом, основанным на статистике (6.11).
25. Выполнить задание 1 с помощью теста пересекающихся n -грамм.
26. Сравнить тест пересекающихся n -грамм и ранее рассмотренный (задание 1) тест непересекающихся n -грамм при одних и тех же значениях n , n_+ , ε по мощности и вычислительной сложности.
27. Выполнить задание 5 с помощью теста пересекающихся n -грамм.
28. Разработать и исследовать (методом компьютерного моделирования) модификацию теста пересекающихся n -грамм, заключающуюся в том, что в (6.12) вместо частот $(n - 1)$ -грамм используются частоты $(n - 2)$ -грамм.
29. Реализовать на компьютере алгоритм вычисления ранга двоичной матрицы.
30. Выполнить задание 1 с помощью теста, основанного на рангах двоичных матриц, для заданных значений k и l в (6.13).
31. Методом компьютерного моделирования для последовательностей File01 – File06 из АДП (прил. 1) исследовать зависимость мощности теста из п. 6.9 и его вычислительной сложности от количества строк k и столбцов l в (6.13) при фиксированном n . Дать рекомендации по использованию произвола при выборе k , l для достижения максимально возможной мощности теста.

32. Модифицировать данный тест на основе использования иных, чем (6.15), функций от двоичной матрицы X .

33. Выполнить задание 1 с помощью алгоритма обнаружения периодичности и оценивания периода (для $K = 1; 2; 5; 10$). Методом компьютерного моделирования для последовательностей File01 – File06 из АДП исследовать зависимость мощности теста и его вычислительной сложности от основных параметров алгоритма.

34. Выполнить задание 33 с помощью семейства спектральных тестов (6.22)–(6.27). Исследовать вопрос оптимального выбора параметра c из условия максимума мощности теста.

35. Выполнить задание 33 с помощью спектрального теста (6.28)–(6.30).

36. Выполнить задание 1 с помощью теста случайного блуждания, основанного на статистике «количество возвратов» (6.31). Методом компьютерного моделирования для последовательностей File01–File06 из АДП исследовать зависимость мощности теста и его вычислительной сложности от основных параметров.

37. Выполнить задание 36 для теста случайного блуждания, основанного на статистике «количество изменений знаков» (6.32).

38. Выполнить задание 36 для теста случайного блуждания, основанного на статистике «вес Хэмминга» (6.33).

39. Сравнить рассмотренные тесты случайного блуждания по мощности и вычислительной сложности.

40. Аналогично (6.31)–(6.33) предложить статистику и построить новый тест случайного блуждания.

41. Выполнить задание 1 с помощью универсального статистического теста Маурера (6.34)–(6.40) для заданных параметров Q, K, L .

42. Методом компьютерного моделирования для последовательностей File01–File06 из АДП исследовать зависимость мощности теста из п. 6.12 и его вычислительной сложности от параметров Q, K, L .

43. Решить задачу оптимального выбора параметров теста Маурера по критерию максимума мощности теста при фиксированной длине n_+ наблюдаемой двоичной последовательности.

44. Указать класс альтернатив H_1 , для которых тест Маурера:

- 1) наиболее чувствителен (имеет наибольшую мощность);
- 2) нечувствителен (не отличает эти альтернативы от H_0).

45. Выполнить задание 1 с помощью теста (6.44) для заданных значений t .

46. Методом компьютерного моделирования для последовательностей File01–File06 из АДП исследовать зависимость мощности теста (6.44) и его вычислительной сложности от параметров n, N, t .

47. Выполнить задание 45 для теста (6.45).
48. Выполнить задание 46 для теста (6.45).
49. Для $N = 2$ разработать программную реализацию тестов (6.44) и (6.45) с максимальным быстродействием.
50. Выполнить задание 1 с помощью теста (6.46), (6.47), (6.49), (6.50). Для $n = 10^6$ выполнить это же задание с помощью теста (6.46), (6.48)–(6.50). Сравнить уровни значимости этих двух тестов.
51. Методом компьютерного моделирования для последовательностей File01–File06 из АДП исследовать мощность и вычислительную сложность тестов из предыдущего задания.
52. Используя «надежный» генератор псевдослучайной двоичной последовательности (предложенный преподавателем), разработать программу оценки параметров μ_n , σ_n в (6.46) методом Монте-Карло.
53. Разработать модификацию данного теста, основанную на обработке M реализаций длиной n каждой и на проверке согласия выборочной функции распределения P -значений (6.50) с функцией распределения стандартного равномерного закона $R[0, 1]$ с помощью критерия Колмогорова.
54. Выполнить задание 1 с помощью алгоритма тестирования, основанного на статистике линейной сложности.
55. Методом компьютерного моделирования для последовательностей File01 – File06 из АДП (прил. 1) исследовать мощность и вычислительную сложность алгоритма тестирования из п. 6.15.
56. Разработать модификацию теста, используя произвол в задании ячеек (6.53) и их количества.
57. Выполнить задание 1 с помощью теста (6.55), (6.56), (6.57). Методом компьютерного моделирования для последовательностей File01 – File06 из АДП исследовать зависимость мощности теста и его вычислительной сложности от основных параметров.
58. Выполнить задание 57 для теста (6.55), (6.56), (6.58).
59. Выполнить задание 57 с учетом замечания 6.4.
60. Выполнить задание 58 с учетом замечания 6.4.
61. Выполнить задания 57–60 с использованием теста на основе экстремальной статистики дельта-произведения.

Г л а в а 7

МЕТОДЫ ТЕОРИИ ИНФОРМАЦИИ В КРИПТОЛОГИИ

7.1. ИСТОЧНИКИ ДИСКРЕТНЫХ СООБЩЕНИЙ И ИХ ВЕРОЯТНОСТНЫЕ МОДЕЛИ

Пусть рассматривается произвольный источник сообщений. Каждое сообщение представляет собой некоторую последовательность символов (например, букв белорусского алфавита, точек и тире в телеграфии, нулей и единиц в компьютерной логике и т. д.). Отдельный символ сообщения будем обозначать ξ , и предполагать числовой величиной, принимающей всевозможные значения из некоторого множества $A \subset \mathbb{R}$, $\xi \in A$.

Множество A значений символа сообщения ξ принято называть *алфавитом сообщений*. Если алфавит A является конечным множеством мощности $2 \leq N < \infty$:

$$A = \{a^{(1)}, a^{(2)}, \dots, a^{(N)}\},$$

то принято говорить, что имеет место *источник дискретных сообщений* (*ИДС*). В противном случае говорят об *источнике непрерывных сообщений* (*ИНС*).

В этом пункте будем рассматривать модели ИДС, наиболее часто используемые в современных криптосистемах, а модели ИНС будут изучены в п. 7.6.

Величины $a^{(1)}, \dots, a^{(N)}$ принято называть *символами алфавита*, а число N – *мощностью алфавита*. Появление в сообщении любого символа алфавита характеризуется высокой степенью неопределенности. Для математического описания этой неопределенности будем использовать дискретную вероятностную модель. Пусть $(\Omega, \mathcal{F}, \mathbf{P})$ – основное вероятностное пространство, описывающее случайный «эксперимент» по появлению (регистрации) символа ξ . Тогда каждому элементарному исходу $\omega \in \Omega$ этого эксперимента ставится в соответствие значение символа $\xi = \xi(\omega) \in A$. Таким образом, символ $\xi = \xi(\omega)$ – дискретная случайная величина, полностью определяемая дискретным распределением вероятностей:

$$\mathbf{P}\{\xi = a\} = p(a), \quad a \in A, \tag{7.1}$$

причем

$$0 < p(a) < 1, \quad \sum_{i=1}^N p(a^{(i)}) = 1. \tag{7.2}$$

В качестве примера в таблице представлено распределение вероятностей символов русского алфавита (упорядоченных в порядке убывания $p(a^{(i)})$):

Символ	пробел	о	е, ё	а	и	т	н	с
Вероятность	0,175	0,090	0,072	0,062	0,062	0,053	0,053	0,045
Символ	р	в	л	к	м	д	п	у
Вероятность	0,040	0,038	0,035	0,028	0,026	0,025	0,023	0,021
Символ	я	ы	з	ь, ъ	б	г	ч	й
Вероятность	0,018	0,016	0,016	0,014	0,014	0,013	0,012	0,010
Символ	х	ж	ю	ш	ц	щ	э	ф
Вероятность	0,009	0,007	0,006	0,006	0,004	0,003	0,003	0,002

Таким образом, ИДС в случае односимвольного сообщения описывается дискретной вероятностной моделью $\langle A, p(a) \rangle$. Эта модель описывает лишь одиночный случайный символ сообщения. Сообщение, порождаемое ИДС, – это в общем случае последовательность $n \geq 1$ случайных символов:

$$\Xi_n = (\xi_1, \dots, \xi_n) \in A^n.$$

При этом полное вероятностное описание ИДС задается вероятностной моделью случайного временного ряда (случайного процесса) Ξ_n с дискретным временем $t \in \mathbb{N}$ и дискретным пространством состояний A ($n = 1, 2, \dots$):

$$\langle A^n, p_n(a_1, \dots, a_n) \rangle, \\ p_n(a_1, \dots, a_n) = \mathbf{P}\{\xi_1 = a_1, \dots, \xi_n = a_n\}, a_1, \dots, a_n \in A, \quad (7.3)$$

где $p_n(a_1, \dots, a_n)$ – n -мерное дискретное распределение вероятностей n -символьного сообщения. Отметим, что n -мерные распределения вероятностей (7.3) удовлетворяют условию самосогласованности ($1 \leq k_1 < \dots < k_m \leq n, 1 \leq m < n$):

$$p_m(a_{k_1}, \dots, a_{k_m}) = \sum_{a_i \in A, i \in \{1, \dots, n\} \setminus \{k_1, \dots, k_m\}} p_n(a_1, \dots, a_n). \quad (7.4)$$

ИДС называется *стационарным*, если случайный процесс Ξ_n является стационарным (в узком смысле), т. е. если конечномерные распределения (7.3) инвариантны относительно сдвига начала отсчета времени.

Стационарный ИДС называется *источником без памяти*, если для любых $a_1, \dots, a_n \in A$ справедлива факторизация n -мерного распределения вероятностей:

$$p_n(a_1, \dots, a_n) = \prod_{i=1}^n p(a_i), \quad (7.5)$$

т. е. порождаемые ИДС случайные символы ξ_1, \dots, ξ_n независимы в совокупности и одинаково распределены.

7.2. ФУНКЦИОНАЛ ЭНТРОПИИ И ЕГО СВОЙСТВА

Пусть ИДС описывается некоторой дискретной вероятностной моделью $\langle A, p(a) \rangle$. Тогда *энтропией* ИДС (или энтропией случайного символа ξ) называется величина, определяемая функционалом [15, 39, 40]:

$$\mathbf{H}\{\xi\} = h(p(a^{(1)}), \dots, p(a^{(N)})) := \mathbf{E}\{-\log_b p(\xi)\} = -\sum_{i=1}^N p(a^{(i)}) \log_b p(a^{(i)}),$$

где $\mathbf{E}\{\cdot\}$ – символ математического ожидания. Если в (7.6) логарифм берется по основанию $b = 2$, то энтропия измеряется в *битах* (*bit* = binary digit), а если используется натуральный логарифм по основанию $b = e$, то энтропия измеряется в *натах* (*nat* = natural digit).

Заметим, что встречающаяся в (7.6) неопределенность $0 \log 0$ разрешается следующим образом: $0 \log 0 := 0$. Мы будем пользоваться логарифмом по основанию $b = 2$ и, следовательно, измерять энтропию в битах.

Сформулируем и докажем основные свойства функционала энтропии.

Свойство 7.1. Функционал энтропии принимает неотрицательные значения: $\mathbf{H}\{\xi\} \geq 0$; он обращается в 0 только для вырожденного распределения:

$$\exists a' \in A, p(a') = 1, p(a) = 0, a \neq a'. \quad (7.7)$$

Доказательство. Поскольку $0 \leq p(\xi) \leq 1$, то $\eta = -\log p(\xi) \geq 0$. Согласно свойству математического ожидания из (7.6) имеем

$$\mathbf{H}\{\xi\} = \mathbf{E}\{\eta\} \geq 0,$$

причем $\mathbf{H}\{\xi\} = 0$ тогда и только тогда, когда $\eta \stackrel{\text{п.н.}}{=} 0$. Последнее соотношение, очевидно, выполняется лишь в случае (7.7). \square

Заметим, что вырожденное распределение (7.7) соответствует случаю, когда символ ξ не является случайным: $\xi = a'$ с вероятностью 1 ($\xi \stackrel{\text{п.н.}}{=} \text{const}$).

Свойство 7.2. Энтропия ИДС с алфавитом мощности $N < \infty$ имеет максимальное значение

$$\max_{p(\cdot)} \mathbf{H}\{\xi\} = \log N, \quad (7.8)$$

которое достигается, если дискретное распределение вероятностей $p(\cdot)$ – равномерное, т. е. все N значений символов равновероятны:

$$p(a) = \frac{1}{N}, \quad a \in A. \quad (7.9)$$

Доказательство. Воспользуемся неравенством Иенсена

$$\mathbf{E}\{f(\zeta)\} \leq f(\mathbf{E}\{\zeta\}), \quad (7.10)$$

которое справедливо для любой случайной величины ζ и произвольной выпуклой функции $y = f(x)$. Положим в (7.10):

$$\zeta = \frac{1}{p(\xi)}, \quad f(x) = \log_2 x, \quad (7.11)$$

□

причем $f(x) = -\frac{1}{(\ln 2)x^2} < 0$, так что $f(\cdot)$ – выпукла вверх. Используя (7.6) и (7.11), имеем

$$\begin{aligned} \mathbf{E}\{\zeta\} &= \sum_{i=1}^N p(a^{(i)}) \frac{1}{p(a^{(i)})} = N, \quad f(\mathbf{E}\{\zeta\}) = \log_2 N; \\ \mathbf{E}\{f(\zeta)\} &= \sum_{i=1}^N p(a^{(i)}) \log_2 \frac{1}{p(a^{(i)})} = \mathbf{H}\{\xi\}. \end{aligned}$$

Подставляя эти выражения в (7.10), получаем (7.8). Подставляя (7.9) в (7.6), находим

$$h\left(\frac{1}{N}, \dots, \frac{1}{N}\right) = -\sum_{i=1}^N \frac{1}{N} \log \frac{1}{N} = \log N.$$

Отметим, что впервые энтропия была введена Д. Хартли в 1928 г. в виде

$$\mathbf{H}\{\xi\} = \log N \quad (7.12)$$

для случайного символа ξ с N равновероятными значениями, и поэтому (7.12) иногда называют *энтропией Хартли*. Энтропия в общем виде (7.6) называется *энтропией Шеннона*.

Следствие 7.1. Чем больше мощность алфавита N , тем больше энтропия Хартли (максимально возможная энтропия).

Свойство 7.3 (свойство аддитивности). Если случайные символы $\xi_1 \in A_1$, $\xi_2 \in A_2$ сообщения независимы, то совместная энтропия равна сумме энтропий:

$$\mathbf{H}\{\xi_1, \xi_2\} = \mathbf{H}\{\xi_1\} + \mathbf{H}\{\xi_2\}. \quad (7.13)$$

Доказательство. Построим на $(\Omega, \mathcal{F}, \mathbf{P})$ случайный вектор $\Xi_2 = (\xi_1, \xi_2) \in A_1 \times A_2$ с дискретным распределением вероятностей:

$$p_2(a_1, a_2) = \mathbf{P}\{\xi_1 = a_1, \xi_2 = a_2\}, \quad \sum_{a_1 \in A_1} \sum_{a_2 \in A_2} p_2(a_1, a_2) = 1.$$

В силу независимости ξ_1 , ξ_2 (ИДС без памяти) имеем

$$p_2(a_1, a_2) = p(a_1)p(a_2), \quad a_1 \in A_1, a_2 \in A_2. \quad (7.14)$$

Тогда из (7.6) и (7.14) следует (с учетом условия нормировки):

$$\begin{aligned}\mathbf{H}\{\xi_1, \xi_2\} &= - \sum_{a_1 \in A_1} \sum_{a_2 \in A_2} p_2(a_1, a_2) \log p_2(a_1, a_2) = \\ &= - \sum_{a_1 \in A_1} p(a_1) \sum_{a_2 \in A_2} p(a_2) (\log p(a_1) + \log p(a_2)) = \\ &= - \sum_{a_1 \in A_1} p(a_1) \log p(a_1) - \sum_{a_2 \in A_2} p(a_2) \log p(a_2) = \\ &= \mathbf{H}\{\xi_1\} + \mathbf{H}\{\xi_2\},\end{aligned}$$

что совпадает с (7.13). \square

Следствие 7.2. Если независимы в совокупности $n \geq 2$ случайных символов $\xi_1 \in A_1, \dots, \xi_n \in A_n$, то их совместная энтропия аддитивна:

$$\mathbf{H}\{\xi_1, \dots, \xi_n\} = \mathbf{H}\{\xi_1\} + \dots + \mathbf{H}\{\xi_n\}. \quad (7.15)$$

Доказательство. Оно состоит в $(n-1)$ -кратном применении свойства 7.3. \square

Свойство 7.4. Добавление к алфавиту символов одного символа с нулевой вероятностью, а следовательно, и любого количества таких символов не изменяет энтропии ИДС.

Доказательство. Пусть алфавит A расширен, как указано в условии:

$$A = \{a^{(1)}, \dots, a^{(N)}\}, \quad A' = A \cup \{a^{(N+1)}\}, \quad \mathbf{P}\{\xi = a^{(N+1)}\} = 0.$$

Тогда согласно (7.6)

$$\mathbf{H}\{\xi'\} = - \sum_{i=1}^{N+1} p(a^{(i)}) \log p(a^{(i)}) = - \sum_{i=1}^N p(a^{(i)}) \log p(a^{(i)}) = \mathbf{H}\{\xi\}.$$

\square

7.3. УСЛОВНАЯ ЭНТРОПИЯ И ЕЕ СВОЙСТВА

Чтобы изучить новые важные свойства энтропии, используемые в криптосистемах, нам понадобится понятие условной энтропии.

Пусть определен случайный n -вектор символов $\xi = (\xi_j) \in A_1 \dots A_n$ с некоторым n -мерным дискретным распределением вероятностей

$$p_n(a_1, \dots, a_n) = \mathbf{P}\{\xi_1 = a_1, \dots, \xi_n = a_n\}, \quad (7.16)$$

где $a_1 \in A_1, \dots, a_n \in A_n$. Пусть задано натуральное число $2 \leq k \leq n$ и определено $(n-k+1)$ -мерное условное распределение вероятностей подвектора $\xi' = (\xi_k, \dots, \xi_n) \in A_k \dots A_n$ при условии, что фиксирован подвектор $\xi = (\xi_1, \dots, \xi_{k-1}) \in A_1 \dots A_{k-1}$:

$$p_{n-k+1}(a_k, \dots, a_n | a_1, \dots, a_{k-1}) = \frac{p_n(a_1, \dots, a_n)}{p_{k-1}(a_1, \dots, a_{k-1})}. \quad (7.17)$$

Здесь использована формула умножения вероятностей.

Условной энтропией подвектора ξ' при условии, что фиксирован подвектор ξ , называется функционал

$$\begin{aligned} \mathbf{H}\{\xi_k, \dots, \xi_n \mid \xi_1 = a_1, \dots, \xi_{k-1} = a_{k-1}\} = \\ = - \sum_{a_k \in A_k} \cdots \sum_{a_n \in A_n} p_{n-k+1}(a_k, \dots, a_n \mid a_1, \dots, a_{k-1}) \times \\ \times \log p_{n-k+1}(a_k, \dots, a_n \mid a_1, \dots, a_{k-1}). \end{aligned} \quad (7.18)$$

Условной энтропией случайного подвектора символов $\xi' = (\xi_k, \dots, \xi_n)$ относительно случайного подвектора символов $\xi = (\xi_1, \dots, \xi_{k-1})$ называется функционал, получающийся усреднением (7.18):

$$\begin{aligned} \mathbf{H}\{\xi' \mid \xi\} = \sum_{a_1 \in A_1} \cdots \sum_{a_{k-1} \in A_{k-1}} p_{k-1}(a_1, \dots, a_{k-1}) \times \\ \times \mathbf{H}\{\xi_k, \dots, \xi_n \mid \xi_1 = a_1, \dots, \xi_{k-1} = a_{k-1}\} = \\ = - \sum_{a_1 \in A_1} \cdots \sum_{a_n \in A_n} p_n(a_1, \dots, a_n) \times \\ \times \log p_{n-k+1}(a_k, \dots, a_n \mid a_1, \dots, a_{k-1}) \geq 0. \end{aligned} \quad (7.19)$$

Продолжим исследование свойств энтропии и условной энтропии с учетом введенных понятий.

Теорема 7.1. *Если подвекторы случайных символов ξ' , ξ независимы, то условная энтропия совпадает с безусловной:*

$$\mathbf{H}\{\xi' \mid \xi\} = \mathbf{H}\{\xi'\}. \quad (7.20)$$

Доказательство. В силу независимости ξ' , ξ условное распределение вероятностей совпадает с безусловным:

$$p_{n-k+1}(a_k, \dots, a_n \mid a_1, \dots, a_{k-1}) = p_{n-k+1}(a_k, \dots, a_n).$$

Подставляя это выражение в (7.19) и используя свойство самосогласованности распределений:

$$\sum_{a_1 \in A_1} \cdots \sum_{a_{k-1} \in A_{k-1}} p_n(a_1, \dots, a_n) = p_{n-k+1}(a_k, \dots, a_n),$$

получаем (7.20). □

Теорема 7.2. *Для любой последовательности случайных символов сообщения ξ_1, \dots, ξ_n энтропия обладает свойством иерархической аддитивности:*

$$\begin{aligned} \mathbf{H}\{\xi_1, \dots, \xi_n\} = \mathbf{H}\{\xi_1\} + \mathbf{H}\{\xi_2 \mid \xi_1\} + \\ + \mathbf{H}\{\xi_3 \mid (\xi_1, \xi_2)\} + \dots + \mathbf{H}\{\xi_n \mid (\xi_1, \dots, \xi_{n-1})\}. \end{aligned} \quad (7.21)$$

Доказательство. Воспользуемся обобщенной формулой умножения вероятностей (свойством иерархической мультиплексивности вероятностей):

$$p_n(a_1, \dots, a_n) = p_1(a_1)p_1(a_2 | a_1) \cdots p_1(a_n | (a_1, \dots, a_{n-1})).$$

Тогда по определению энтропии с учетом свойств вероятности имеем

$$\begin{aligned} H\{\xi_1, \dots, \xi_n\} &= - \sum_{a_1 \in A_1} p_1(a_1) \log_2 p_1(a_1) - \\ &- \sum_{a_1 \in A_1} \sum_{a_2 \in A_2} p_2(a_1, a_2) \log_2 p_1(a_2 | a_1) - \dots - \\ &- \sum_{a_1 \in A_1} \cdots \sum_{a_n \in A_n} p_n(a_1, \dots, a_n) \log_2 p_1(a_n | (a_1, \dots, a_{n-1})). \end{aligned}$$

Учитывая (7.19), приходим к (7.21). \square

Следствие 7.3. *Если случайные символы сообщения ξ_1, \dots, ξ_n независимы в совокупности, то выполняется доказанное в предыдущем пункте свойство аддитивности (7.15):*

$$H\{\xi_1, \dots, \xi_n\} = \sum_{i=1}^n H\{\xi_i\}.$$

Доказательство. Согласно теореме 7.1 условные энтропии совпадают с безусловными:

$$H\{\xi_i | (\xi_1, \dots, \xi_{i-1})\} = H\{\xi_i\}, \quad (i = \overline{1, n}).$$

Используя с учетом этого факта теорему 7.2, приходим к доказываемому. \square

Следует заметить, что свойство иерархической аддитивности (7.21) и его следствие порождаются наличием логарифмической функции в определении энтропии; это свойство является ключевым для функционала энтропии.

Теорема 7.3. *Пусть $\xi = (\xi_i) \in A$ – произвольный случайный вектор символов с дискретным распределением вероятностей $p_n(a)$, $a = (a_i) \in A$, а $q_n(a)$, $a \in A$ – некоторое дискретное распределение вероятностей. Тогда справедливо неравенство*

$$J(p_n : q_n) = \sum_{a \in A} p_n(a) \log_2 \frac{p_n(a)}{q_n(a)} \geq 0. \quad (7.22)$$

Равенство нулю имеет место тогда и только тогда, когда распределение $q_n(\cdot)$ совпадает с $p_n(\cdot)$:

$$q_n(a) = p_n(a), \quad a \in A. \quad (7.23)$$

Доказательство. Воспользуемся известным неравенством Иенсена, справедливым для произвольной случайной величины ζ и выпуклой вверх функции $f(x)$:

$$E\{f(\zeta)\} \leq f(E\{\zeta\}). \quad (7.24)$$

Положим в (7.24)

$$\zeta := \frac{q_n(\xi)}{p_n(\xi)} \geq 0, f(x) := \log_2 x.$$

Тогда, используя условие нормировки, имеем

$$\mathbf{E}\{\zeta\} = \sum_{a \in A} p_n(a) \frac{q_n(a)}{p_n(a)} = \sum_{a \in A} q_n(a) = 1,$$

$$\mathbf{E}\{f(\zeta)\} = \sum_{a \in A} p_n(a) \log_2 \frac{q_n(a)}{p_n(a)} = -J(p_n : q_n).$$

Подставляя эти выражения в (7.24), получаем неравенство

$$J(p_n : q_n) \geq 0.$$

Как известно, равенство в неравенстве Иенсена имеет место тогда и только тогда, когда $\zeta \equiv \text{const} = c$. В силу условия нормировки константа c может быть равна только единице:

$$c = \frac{q_n(a)}{p_n(a)} \equiv 1,$$

что и означает $q_n(\cdot) \equiv p_n(\cdot)$. □

Теорема 7.4. Условная энтропия не может превосходить безусловную:

$$\mathbf{H}\{\{\} \mid \xi'\} \leq \mathbf{H}\{\xi'\}. \quad (7.25)$$

Доказательство. Воспользуемся теоремой 7.3 и положим в (7.22)

$$p(a) := \mathbf{P}\{\xi' = a \mid \xi = b\}, q(a) := \mathbf{P}\{\xi' = a\}, a \in A, b \in B.$$

Тогда получим

$$\sum_{a \in A} \mathbf{P}\{\xi' = a \mid \xi = b\} \log_2 \frac{\mathbf{P}\{\xi' = a \mid \xi = b\}}{\mathbf{P}\{\xi' = a\}} \geq 0$$

или (что эквивалентно)

$$\begin{aligned} & - \sum_{a \in A} \mathbf{P}\{\xi' = a \mid \xi = b\} \log_2 \mathbf{P}\{\xi' = a \mid \xi = b\} \leq \\ & \leq - \sum_{a \in A} \mathbf{P}\{\xi' = a \mid \xi = b\} \log_2 \mathbf{P}\{\xi' = a\}. \end{aligned}$$

Умножим обе части этого неравенства на $P\{\xi = b\} \geq 0$ и просуммируем по всевозможным $b \in B$ (с учетом формулы умножения вероятностей):

$$\begin{aligned} & - \sum_{a \in A} \sum_{b \in B} P\{\xi' = a \mid \xi = b\} \log_2 P\{\xi' = a \mid \xi = b\} \leq \\ & \leq - \sum_{a \in A} \sum_{b \in B} P\{\xi' = a, \xi = b\} \log_2 P\{\xi' = a\} = \\ & = - \sum_{a \in A} P\{\xi' = a\} \log_2 P\{\xi' = a\}, \end{aligned}$$

что совпадает с (7.25). \square

Следствие 7.4. При добавлении условий условная энтропия не увеличивается:

$$H\{\xi \mid (\eta, \zeta)\} \leq H\{\xi \mid \eta\}. \quad (7.26)$$

Доказательство. Неравенство (7.26) доказывается аналогично теореме 7.4. \square

Следствие 7.5. Энтропия последовательности случайных символов сообщения ξ_1, \dots, ξ_n не превосходит суммы энтропий всех этих символов, рассматриваемых по отдельности:

$$H\{\xi_1, \dots, \xi_n\} \leq \sum_{i=1}^n H\{\xi_i\}. \quad (7.27)$$

Доказательство. Согласно теореме 7.4 справедливы неравенства

$$H\{\xi_i \mid (\xi_1, \dots, \xi_{i-1})\} \leq H\{\xi_i\} \quad (i = \overline{2, n}).$$

Подставляя их в (7.21), получаем (7.27). \square

В криптологии дискретные сообщения $\xi \in A$ часто подвергаются некоторому дискретному функциональному преобразованию ($\mathcal{D}\Phi\mathcal{P}$):

$$\eta = f(\xi), \quad \xi \in A, \quad \eta \in B, \quad (7.28)$$

где A, B – некоторые конечные множества. Исследуем, как изменяется энтропия сообщения при таком функциональном преобразовании.

Теорема 7.5. При дискретном функциональном преобразовании вида (7.28) энтропия не возрастает:

$$H\{\eta\} \leq H\{\xi\}, \quad (7.29)$$

причем равенство в (7.29) достигается тогда и только тогда, когда $\mathcal{D}\Phi\mathcal{P}$ (7.28) – биекция.

Доказательство. Рассмотрим «составное» сообщение $\binom{\xi}{\eta} \in A \times B$. По теореме 7.2

$$H\{\xi, \eta\} = H\{\xi\} + H\{\eta | \xi\} = H\{\eta\} + H\{\xi | \eta\}. \quad (7.30)$$

В силу функциональной зависимости (7.28) условное распределение η , при условии, что $\xi = a$ фиксировано, является вырожденным:

$$P\{\eta = b | \xi = a\} = P\{f(a) = b | \xi = a\} = \delta_{f(a), b}, \quad b \in B,$$

где $\delta_{i,j}$ – символ Кронеккера. Согласно свойству 7.1 энтропия для вырожденного распределения обращается в 0:

$$H\{\eta | \xi\} = 0.$$

Тогда из последнего равенства (7.30) имеем

$$H\{\eta\} = H\{\xi\} - H\{\xi | \eta\}.$$

Поскольку $H\{\xi | \eta\} \geq 0$ по свойству энтропии, то отсюда следует (7.28). Равенство в (7.29) будет тогда и только тогда, когда $H\{\xi | \eta\} = 0$. Это возможно лишь в случае функциональной зависимости ξ от η : $\xi = f^{-1}(\eta)$, т. е. когда (7.28) – биекция. \square

Введенный и исследованный в пп. 7.2 и 7.3 функционал энтропии Шеннона

$$H\{\xi\} = h_N(p_1, \dots, p_N) = - \sum_{i=1}^N p_i \log p_i \quad (7.31)$$

для ИДС $\langle A, \{p_i\} \rangle$, $p_i = P\{\xi = a^{(i)}\}$, $i = \overline{1, N}$ обладает всеми свойствами, которые необходимо требовать от количественной меры неопределенности. Поэтому энтропию Шеннона $H\{\xi\}$ и используют в криптологии как количественную меру неопределенности сообщения $\xi \in A$. Однако является ли (7.31) единственным функционалом, обладающим изученными свойствами? Положительный ответ на этот вопрос дает теорема единственности, приведенная без доказательства [39].

Теорема 7.6. Пусть для любого натурального числа N функция N переменных

$$h = h_N(p_1, \dots, p_N), \quad p_1, \dots, p_N \geq 0, \quad \sum_{i=1}^N p_i = 1,$$

непрерывна по совокупности аргументов и обладает следующими тремя свойствами:

а) максимальное значение функции достигается при равномерном распределении:

$$h_N(p_1, \dots, p_N) \leq h_N\left(\frac{1}{N}, \dots, \frac{1}{N}\right);$$

б) иерархическая аддитивность:

$$\mathbf{H}\{\xi, \eta\} = \mathbf{H}\{\xi\} + \mathbf{H}\{\eta | \xi\};$$

в) добавление к алфавиту, состоящему из N символов, одного символа с нулевой вероятностью ($p_{N+1} = 0$) не изменяет значения энтропии:

$$h_{N+1}(p_1, \dots, p_N, 0) = h_N(p_1, \dots, p_N).$$

Тогда эта функция $h_N(\cdot)$ необходимо имеет шенноновский вид:

$$h_N(p_1, \dots, p_N) = -\lambda \sum_{i=1}^N p_i \log p_i,$$

где λ – произвольная положительная константа.

7.4. УДЕЛЬНАЯ ЭНТРОПИЯ СТАЦИОНАРНОЙ СИМВОЛЬНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ

Пусть рассматривается стационарный ИДС с некоторым алфавитом $A = \{a^{(1)}, \dots, a^{(N)}\}$, порождающий (генерирующий) n -символьные случайные сообщения $\Xi_n = (\xi_1, \dots, \xi_n) \in A^n$. Исследуем асимптотическое поведение энтропии $\mathbf{H}\{\Xi_n\}$ при $n \rightarrow \infty$.

В силу свойства стационарности (в узком смысле) n -мерное дискретное распределение вероятностей

$$p_n(a_1, \dots, a_N) = \mathbf{P}\{\xi_1 = a_1, \dots, \xi_N = a_N\}, \quad a_1, \dots, a_N \in A, \quad (7.32)$$

не меняется при сдвиге $\tau \geq 1$ начала отсчета времени:

$$\mathbf{P}\{\xi_{\tau+1} = a_1, \dots, \xi_{\tau+n} = a_n\} = p_n(a_1, \dots, a_n) = \text{invar}_{\tau}. \quad (7.33)$$

В силу (7.32) и (7.33) энтропия n -символьного «сдвинутого» сообщения $\tilde{\Xi}_{n,\tau} = (\xi_{\tau+1}, \dots, \xi_{\tau+n}) \in A^n$ не зависит от τ :

$$\mathbf{H}\{\tilde{\Xi}_{n,\tau}\} = \mathbf{H}\{\Xi_n\} = \text{invar}_{\tau}. \quad (7.34)$$

Поэтому в дальнейшем при исследовании энтропии мы можем не делать различий между случайными сообщениями Ξ_n и $\tilde{\Xi}_{n,\tau}$.

Удельной энтропией (плотностью энтропии) стационарного ИДС называется предел

$$h = \lim_{n \rightarrow \infty} \frac{H\{\Xi_n\}}{n}, \quad (7.35)$$

если он существует.

Согласно определению (7.35) h представляет собой энтропию, приходящуюся на один символ и вычисленную по бесконечно длинному случайному сообщению. Если $h > 0$, то с увеличением длины сообщения $n \rightarrow \infty$ энтропия растет линейно:

$$H\{\Xi_n\} \sim hn. \quad (7.36)$$

Знак \sim означает, что $H\{\Xi_n\}/(hn) \rightarrow 1$. Выясним условия существования предела в (7.35), т. е. асимптотического поведения энтропии (7.36).

Пусть $\Xi_{n-1} \in A^{n-1}$ и $\Xi_n = (\Xi_{n-1} || \xi_n) \in A^n$ – случайные сообщения длиной $n - 1$ и n соответственно (здесь $||$ – символ конкатенации (присоединения)). Обозначим условную энтропию символа ξ_n относительно случайного сообщения Ξ_{n-1} , состоящего из $n - 1$ предыдущих символов ($n = 1, 2, \dots$):

$$H\{\xi_n | \Xi_{n-1}\} = - \sum_{a_1, \dots, a_n \in A} p_n(a_1, \dots, a_n) \log p_1(a_n | a_1, \dots, a_{n-1}). \quad (7.37)$$

При этом в случае $n = 1$ полагается $H\{\xi_1 | \Xi_0\} := H\{\xi_1\}$.

Теорема 7.7. Для произвольного стационарного ИДС числовая последовательность условных энтропий $H\{\xi_n | \Xi_{n-1}\}$, $n = 1, 2, \dots$, определяемых (7.37), имеет конечный предел:

$$H\{\xi | \Xi_\infty\} := \lim_{n \rightarrow \infty} H\{\xi_n | \Xi_{n-1}\}. \quad (7.38)$$

Доказательство. Докажем, что исследуемая последовательность не возрастает и ограничена снизу. Действительно, в силу свойства стационарности (7.34) $\forall i, j \in \mathbb{N}$:

$$H\{\xi_j | \xi_{j-i}, \dots, \xi_{j-1}\} = H\{\xi_n | \xi_{n-i}, \dots, \xi_{n-1}\}.$$

Поэтому рассматриваемая последовательность совпадает с последовательностью

$$H\{\xi_n\}, H\{\xi_n | \xi_{n-1}\}, \dots, H\{\xi_n | \xi_1, \dots, \xi_{n-1}\}.$$

В силу свойств условной энтропии

$$H\{\xi_n\} \geq H\{\xi_n | \xi_{n-1}\} \geq \dots \geq H\{\xi_n | \xi_1, \dots, \xi_{n-1}\} \geq 0.$$

Как известно, любая невозрастающая ограниченная снизу числовая последовательность имеет предел, который обозначается согласно (7.38). \square

Теорема 7.8. Для произвольного стационарного ИДС удельная энтропия (7.35) существует и совпадает с предельным значением (7.38):

$$h = H\{\xi | \Xi_\infty\}. \quad (7.39)$$

Доказательство. Во-первых, докажем, что числовая последовательность

$$h_n := \frac{H\{\Xi_n\}}{n}, \quad n = 1, 2, \dots \quad (7.40)$$

из определения удельной энтропии (7.35) ($h := \lim_{n \rightarrow \infty} h_n$) является невозрастающей и ограниченной снизу. Для $(n+1)$ -символьного сообщения $\Xi_{n+1} = (\Xi_n || \xi_{n+1}) \in A^{n+1}$ в силу свойства иерархической аддитивности энтропии имеем

$$H\{\Xi_{n+1}\} = H\{\Xi_n\} + H\{\xi_{n+1} | \Xi_n\}. \quad (7.41)$$

В силу свойства условной энтропии (следствие 7.4) и свойства стационарности (7.34) имеем оценку для второго слагаемого в (7.41):

$$\begin{aligned} H\{\xi_{n+1} | \Xi_n\} &= H\{\xi_{n+1} | \xi_1, \dots, \xi_n\} \leq H\{\xi_{n+1} | \xi_2, \dots, \xi_n\} = \\ &= H\{\xi_n | \xi_1, \dots, \xi_{n-1}\} = H\{\xi_n | \Xi_{n-1}\}. \end{aligned}$$

Подставляя это в (7.41), получаем неравенство

$$H\{\Xi_{n+1}\} \leq H\{\Xi_n\} + H\{\xi_n | \Xi_{n-1}\}. \quad (7.42)$$

В силу свойства иерархической аддитивности энтропии (теорема 7.8) и свойств условной энтропии имеем

$$H\{\Xi_n\} = \sum_{i=1}^n H\{\xi_i | \Xi_{i-1}\} \geq nH\{\xi_n | \Xi_{n-1}\},$$

так что

$$H\{\xi_n | \Xi_{n-1}\} \leq \frac{1}{n} H\{\Xi_n\} = h_n. \quad (7.43)$$

Подставляя (7.43) в (7.42), получаем

$$0 \leq H\{\Xi_{n+1}\} \leq \left(1 + \frac{1}{n}\right) H\{\Xi_n\} = \frac{n+1}{n} H\{\Xi_n\}.$$

Разделив обе части этого неравенства на $n+1$ и использовав обозначение (7.40), находим:

$$0 \leq h_{n+1} \leq h_n,$$

т. е. $\{h_n\}$ – невозрастающая числовая последовательность, ограниченная снизу. Следовательно, ее предел (7.35) – удельная энтропия h – существует.

Во-вторых, покажем справедливость равенства (7.39) для этого предела. Для произвольного $1 \leq k < n$ по свойствам энтропии с учетом (7.40) имеем

$$\begin{aligned} h_n &= \frac{1}{n} \sum_{i=1}^n H\{\xi_i | \Xi_{i-1}\} \equiv \frac{1}{n} \sum_{i=1}^k H\{\xi_i | \Xi_{i-1}\} + \\ &+ \frac{1}{n} \sum_{i=k+1}^n H\{\xi_i | \Xi_{i-1}\} \leq \frac{k}{n} H\{\xi_1\} + \frac{n-k}{n} H\{\xi_{k+1} | \Xi_k\}. \end{aligned} \quad (7.44)$$

Воспользуемся произволом k и выберем $k = k(\varepsilon)$ таким, чтобы для любого наперед заданного $\varepsilon > 0$ выполнялось неравенство

$$H\{\xi_{k+1} | \Xi_k\} - H\{\xi | \Xi_\infty\} \leq \frac{\varepsilon}{2}. \quad (7.45)$$

Это всегда можно сделать, так как по теореме 7.7

$$H\{\xi_{k+1} | \Xi_k\} \rightarrow H\{\xi | \Xi_\infty\} + 0.$$

По выбранному таким образом k определим $\bar{n} = \bar{n}(k, \varepsilon)$ так, чтобы при любом $n > \bar{n}$ выполнялось неравенство

$$\frac{k}{n} H\{\xi_1\} \leq \frac{\varepsilon}{2}. \quad (7.46)$$

Тогда из (7.44), (7.45), (7.46) следует, что $\forall \varepsilon > 0 \exists \bar{n}_1 = \bar{n}_1(\varepsilon)$ такое, что для всех $n > \bar{n}_1$ справедлива оценка сверху:

$$h_n \leq H\{\xi | \Xi_\infty\} + \varepsilon. \quad (7.47)$$

С другой стороны, имеем оценку снизу:

$$h_n = \frac{1}{n} \sum_{i=1}^n H\{\xi_i | \Xi_{i-1}\} \geq H\{\xi_n | \Xi_{n-1}\} \geq H\{\xi | \Xi_\infty\}. \quad (7.48)$$

Объединяя (7.47) и (7.48), имеем

$$H\{\xi | \Xi_\infty\} \leq h_n \leq H\{\xi | \Xi_\infty\} + \varepsilon.$$

Поскольку ε – произвольное положительное число, то из этих неравенств следует, что

$$h = \lim_{n \rightarrow \infty} h_n = H\{\xi | \Xi_\infty\},$$

т. е. выполняется (7.39). \square

Из теорем 7.7 и 7.8 следует *асимптотическое разложение энтропии* n -символьного сообщения:

$$H\{\Xi_n\} = nh + o(n). \quad (7.49)$$

Уточним остаточный член в (7.48).

Теорема 7.9. Для энтропии произвольной дискретной стационарной случайной последовательности $\Xi_n \in A^n$ при увеличении количества символов $n \rightarrow \infty$ справедлива асимптотика

$$\mathbf{H}\{\Xi_n\} = nh + 2b + o(1), \quad (7.50)$$

где

$$b = \frac{1}{2} \lim_{m, n \rightarrow \infty} (\mathbf{H}\{\Xi_m\} + \mathbf{H}\{\Xi_n\} - \mathbf{H}\{\Xi_{m+n}\}) \geq 0. \quad (7.51)$$

Доказательство. Обозначим

$$B_{mn} = \mathbf{H}\{\Xi_m\} + \mathbf{H}\{\Xi_n\} - \mathbf{H}\{\Xi_{m+n}\}, \quad m, n \in \mathbb{N}. \quad (7.52)$$

Во-первых, заметим, что B_{mn} – симметричная функция относительно m, n .

Во-вторых, по свойству иерархической аддитивности и свойству стационарности из (7.38) имеем

$$\begin{aligned} B_{mn} &= \mathbf{H}\{\Xi_m\} + \mathbf{H}\{\xi_{m+1}, \dots, \xi_{m+n}\} - \\ &\quad - (\mathbf{H}\{\Xi_m\} + \mathbf{H}\{\xi_{m+1}, \dots, \xi_{m+n}\} \mid \Xi_m) = \\ &= \mathbf{H}\{\xi_{m+1}, \dots, \xi_{m+n}\} - \mathbf{H}\{\xi_{m+1}, \dots, \xi_{m+n} \mid \Xi_m\}. \end{aligned} \quad (7.53)$$

В силу известного свойства энтропии для $\eta = (\xi_{m+1}, \dots, \xi_{m+n})$

$$\mathbf{H}\{\eta\} \geq \mathbf{H}\{\eta \mid \xi_1\} \geq \dots \geq \mathbf{H}\{\eta \mid \xi_1, \dots, \xi_m\}$$

при фиксированном n и растущем m энтропия

$$\mathbf{H}\{\xi_{m+1}, \dots, \xi_{m+n} \mid \Xi_m\}$$

не возрастает. Следовательно, согласно (7.53) последовательность при фиксированном n B_{mn} – неубывающая по $m = 1, 2, \dots$. Аналогичное свойство монотонности выполняется для B_{mn} по n , так как B_{mn} симметрична относительно m, n .

В силу установленного свойства монотонности существует предел (7.51) – конечный или бесконечный.

Чтобы доказать (7.50), воспользуемся (7.51), (7.53) и свойством стационарности:

$$2b = \lim_{m, n \rightarrow \infty} (\mathbf{H}\{\Xi_n\} - \mathbf{H}\{\xi_{m+1}, \dots, \xi_{m+n} \mid \xi_1, \dots, \xi_m\}). \quad (7.54)$$

Применим еще раз свойство иерархической аддитивности энтропии:

$$\mathbf{H}\{\xi_{m+1}, \dots, \xi_{m+n} \mid \xi_1, \dots, \xi_m\} = \sum_{i=1}^n \mathbf{H}\{\xi_{m+i} \mid \xi_1, \dots, \xi_{m+i-1}\}. \quad (7.55)$$

Устремим $m \rightarrow \infty$ в выражении (7.55) и воспользуемся теоремой 7.8 ($i = \overline{1, n}$):

$$\mathbf{H}\{\xi_{m+i} \mid \xi_1, \dots, \xi_{m+i-1}\} \rightarrow h, m \rightarrow \infty.$$

В результате из (7.55) получаем

$$\lim_{m \rightarrow \infty} \mathbf{H}\{\xi_{m+1}, \dots, \xi_{m+n} \mid \xi_1, \dots, \xi_m\} = nh. \quad (7.56)$$

Переходя в (7.54) к пределу при $m \rightarrow \infty$ и используя (7.56), имеем

$$\lim_{n \rightarrow \infty} (\mathbf{H}\{\Xi_n\} - nh) = 2b,$$

что эквивалентно (7.50). \square

Асимптотическое разложение энтропии (7.50) имеет следующую содержательную интерпретацию. Главный член разложения nh – это n -кратная удельная энтропия. Второй член разложения $2b$ – энтропия, обусловленная краевыми (границными) эффектами. Третий член $o(1)$ – остаточный член разложения.

Следствие 7.6. Для стационарного ИДС без памяти соотношение (7.50) обращается в точное равенство

$$\mathbf{H}\{\Xi_n\} = nh, \quad (7.57)$$

где

$$h = \mathbf{H}\{\xi_1\} = - \sum_{a_1 \in A} p_1(a_1) \log p_1(a_1)$$

есть энтропия единичного случайного символа.

Доказательство. Соотношение (7.57) следует из свойств условной энтропии. Кроме того, $\mathbf{H}\{\Xi_{m+n}\} = \mathbf{H}\{\Xi_m\} + \mathbf{H}\{\Xi_n\}$, поэтому согласно (7.51) граничные эффекты отсутствуют: $b = 0$. \square

7.5. ЭНТРОПИЙНЫЕ ХАРАКТЕРИСТИКИ МАРКОВСКИХ СИМВОЛЬНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

В системах обработки и защиты информации символы, образующие сообщение $\Xi_n = (\xi_1, \dots, \xi_n) \in A^n$, обычно стохастически зависимы. Одной из распространенных моделей стохастической зависимости символов печатных текстов, речевых сообщений и изображений является *однородная цепь Маркова* (ОЦМ) с дискретным временем и конечным пространством состояний $A = \{a^{(1)}, \dots, a^{(N)}\}$, $N < \infty$ (см. п. 5.3).

Пусть случайная символьная последовательность, порожденная источником дискретных сообщений $\xi_1, \dots, \xi_n, \dots \in A$, является однородной цепью Маркова. Это означает, что распределение вероятностей будущих значений (состояний) при фиксированных настоящих и прошлых значениях не зависит от прошлых значений:

$$\mathbf{P} \{ \xi_{t+1} = a_{t+1} \mid \xi_t = a_t, \dots, \xi_1 = a_1 \} = p_{a_t, a_{t+1}}, \quad (7.58)$$

$a_1, \dots, a_{t+1} \in A$, $t = 1, 2, \dots$. Известно также, что все конечномерные распределения и все вероятностные характеристики ОЦМ полностью выражаются лишь через вектор-столбец начальных вероятностей

$$\pi = \begin{pmatrix} \pi_1 \\ \vdots \\ \pi_N \end{pmatrix}, \quad \pi_i = \mathbf{P} \{ \xi_1 = a^{(i)} \}, \quad i = \overline{1, N}, \quad (7.59)$$

и через $(N \times N)$ -матрицу вероятностей одношаговых переходов:

$$P = (p_{ij}), \quad p_{ij} = \mathbf{P} \{ \xi_{t+1} = a^{(j)} \mid \xi_t = a^{(i)} \}, \quad i, j = \overline{1, N} \quad (t = 1, 2, \dots). \quad (7.60)$$

Вероятности (7.59) и (7.60) удовлетворяют условиям нормировки:

$$\sum_{i=1}^N \pi_i = 1, \quad \sum_{j=1}^N p_{ij} = 1 \quad (i = \overline{1, N}). \quad (7.61)$$

Для получения дальнейших результатов введем N -вектор-столбец *стационарных вероятностей* $\pi^* = (\pi_i^*)$, $i = \overline{1, N}$, являющийся решением системы линейных алгебраических уравнений:

$$\begin{cases} \sum_{i=1}^n \pi_i p_{ij} = \pi_j, & (i = \overline{1, N}), \\ \pi_1 + \dots + \pi_N = 1. \end{cases} \quad (7.62)$$

Обозначим:

$$H^* \{ \xi_1 \} = - \sum_{i=1}^N \pi_i^* \log \pi_i^* -$$

энтропия стационарного распределения вероятностей;

$$h = - \sum_{i=1}^N \pi_i^* \sum_{j=1}^N p_{ij} \log p_{ij}. \quad (7.63)$$

Теорема 7.10. Если случайная символьная последовательность является ОЦМ со стационарным начальным распределением π^* и матрицей вероятностей одношаговых переходов P , то энтропия n -символьного сообщения Ξ_n равна

$$H\{\Xi_n\} = H^*\{\xi_1\} + (n-1)h, \quad (7.64)$$

при этом величина h , определяемая (7.63), является удельной энтропией.

Доказательство. В силу марковского свойства (7.58) имеем следующее представление для n -мерного дискретного распределения вероятностей:

$$\begin{aligned} p_n(a^{(i_1)}, \dots, a^{(i_n)}) &= P\{\xi_1 = a^{(i_1)}\} \times \\ &\times \prod_{t=1}^{n-1} P\{\xi_{t+1} = a^{(i_{t+1})} \mid \xi_t = a^{(i_t)}, \dots, \xi_1 = a^{(i_1)}\} = \\ &= \pi_{i_1} \prod_{t=1}^{n-1} p_{i_t, i_{t+1}} (i_1, \dots, i_n \in \{1, \dots, N\}). \end{aligned} \quad (7.65)$$

Тогда с учетом (7.65) энтропия n -символьного сообщения $\Xi_n = (\xi_1, \dots, \xi_n)$ будет иметь вид

$$H\{\Xi_n\} = - \sum_{i_1, \dots, i_n=1}^N p_n(a^{(i_1)}, \dots, a^{(i_n)}) \left(\log \pi_{i_1} + \sum_{t=1}^{n-1} \log p_{i_t, i_{t+1}} \right).$$

В силу свойства согласованности многомерных вероятностных распределений имеем

$$\begin{aligned} H\{\Xi_n\} &= - \sum_{i_1=1}^N \pi_{i_1} \log \pi_{i_1} - \\ &- \sum_{t=1}^{n-1} \sum_{i_t, i_{t+1}=1}^N P\{\xi_t = a^{(i_t)}, \xi_{t+1} = a^{(i_{t+1})}\} \log p_{i_t, i_{t+1}} = \\ &= - \sum_{i=1}^N \pi_i \log \pi_i - \sum_{t=1}^{n-1} \sum_{i_t, i_{t+1}=1}^N P\{\xi_t = a^{(i_t)}\} p_{i_t, i_{t+1}} \log p_{i_t, i_{t+1}}. \end{aligned} \quad (7.66)$$

По условию теоремы начальное распределение ОЦМ совпадает с ее стационарным распределением: $\pi_i = \pi_i^* (i = \overline{1, N})$. Тогда из (7.62) легко показать, что одномерное распределение ОЦМ не изменяется с течением времени:

$$P\{\xi_t = a^{(i)}\} = \pi_i^*, \quad t = 1, 2, \dots \quad (i = \overline{1, N}).$$

Учитывая это в (7.66) и используя обозначения (7.63), получаем

$$H\{\Xi_n\} = H^*\{\xi_1\} + \sum_{t=1}^{n-1} \sum_{i=1}^N \left(-\pi_i^* \sum_{j=1}^N p_{ij} \log p_{ij} \right) = H^*\{\xi_1\} + (n-1)h,$$

что совпадает с (7.64).

Найдем удельную энтропию, используя определение и (7.64):

$$\lim_{n \rightarrow \infty} \frac{\mathbf{H}\{\Xi_n\}}{n} = \lim_{n \rightarrow \infty} \frac{\mathbf{H}^*\{\xi_1\}}{n} + h \lim_{n \rightarrow \infty} \left(1 - \frac{1}{n}\right) = h,$$

что и требовалось доказать. \square

Отметим, что если нарушено условие теоремы о стационарности ОЦМ $\pi_i = \pi_i^* \ (i = \overline{1, N})$, то формула (7.66) примет вид

$$\mathbf{H}\{\Xi_n\} = \mathbf{H}\{\xi_1\} + \sum_{t=1}^{n-1} \sum_{i,j=1}^N \left(-\pi_i^{(t)} p_{ij} \log p_{ij}\right), \quad (7.67)$$

где

$$\pi_i^{(t)} = \mathbf{P}\{\xi_t = a^{(i)}\}.$$

Из теории цепей Маркова известно, что при $t \rightarrow \infty$ и выполнении некоторых ограничений на P согласно эргодической теореме имеет место экспоненциальная сходимость распределения вероятностей $\pi^{(t)} = (\pi_i^{(t)})$ к стационарному распределению $\pi^* = (\pi_i^*)$:

$$\pi_i^{(t)} = \pi_i^* + O(\rho^t), \quad i = \overline{1, N},$$

где $0 < \rho < 1$. Поэтому из (7.67) имеем

$$\frac{\mathbf{H}\{\Xi_n\}}{n} = \frac{\mathbf{H}\{\Xi_1\}}{n} + \frac{1}{n} \left((n-1)h + \sum_{t=1}^{n-1} O(\rho^t) \right) \rightarrow h, \quad n \rightarrow \infty.$$

Таким образом, удельная энтропия ОЦМ даже при нарушении условия стационарности определяется (7.63), хотя соотношение (7.64) при этом не выполняется.

Следствие 7.7. В условиях теоремы 7.10 соотношение (7.50) обращается в точное равенство:

$$\mathbf{H}\{\Xi_n\} = nh + 2b, \quad (7.68)$$

где

$$2b = \sum_{i,j=1}^N \pi_i^* p_{ij} \log \frac{p_{ij}}{\pi_j^*}. \quad (7.69)$$

Доказательство. Представляя (7.64) в виде (7.68), имеем

$$2b = \mathbf{H}^*\{\xi_1\} - h.$$

Тогда из (7.63) следует:

$$2b = - \sum_{i=1}^N \pi_i^* \log \pi_i^* + \sum_{i=1}^N \pi_i^* \sum_{j=1}^N p_{ij} \log p_{ij} = \sum_{i=1}^N \pi_i^* \sum_{j=1}^N p_{ij} \log \frac{p_{ij}}{\pi_j^*}.$$

Используя (7.62), можно показать, что это выражение совпадает с правой частью (7.69). \square

Пример 7.1. Рассмотрим двоичную (бинарную) ОЦМ $\xi_1, \xi_2, \dots \in A = \{0, 1\}$, т. е. $N = 2$, $a^{(1)} = 0$, $a^{(2)} = 1$. Вероятностные характеристики ОЦМ:

$$\pi = \begin{pmatrix} \pi_1 \\ \pi_2 \end{pmatrix}, P = \begin{pmatrix} 1 - \alpha & \alpha \\ \beta & 1 - \beta \end{pmatrix},$$

где $\alpha, \beta \in [0, 1]$. Заметим, что при $\alpha + \beta = 1$, $\pi_1 = 1 - \alpha$, $\pi_2 = \alpha$ имеем схему независимых испытаний. Для нахождения стационарного распределения π^* согласно (7.62) имеем систему

$$\begin{cases} \pi_1^*(1 - \alpha) + \pi_2^*\beta = \pi_1^*, \\ \pi_1^* + \pi_2^* = 1. \end{cases}$$

Решение ее единственное:

$$\pi^* = \begin{pmatrix} \frac{\beta}{\alpha + \beta} \\ \frac{\alpha}{\alpha + \beta} \end{pmatrix}.$$

По формулам (7.63) находим

$$\begin{aligned} H^* \{ \xi_1 \} &= -\frac{\beta}{\alpha + \beta} \log \frac{\beta}{\alpha + \beta} - \frac{\alpha}{\alpha + \beta} \log \frac{\alpha}{\alpha + \beta}, \\ h &= -\frac{\beta}{\alpha + \beta} (\alpha \log \alpha + (1 - \alpha) \log (1 - \alpha)) - \\ &\quad -\frac{\alpha}{\alpha + \beta} (\beta \log \beta + (1 - \beta) \log (1 - \beta)). \end{aligned}$$

По формуле (7.69) получаем

$$\begin{aligned} 2b &= 2 \frac{\alpha \beta}{\alpha + \beta} \log \beta + \log \left(1 + \frac{\alpha}{\beta} \right) + \frac{\beta}{\alpha + \beta} (1 - \alpha) \log (1 - \alpha) + \\ &\quad + \frac{\alpha}{\alpha + \beta} (1 - \beta) \log (1 - \beta). \end{aligned}$$

При $\beta = \alpha$ (симметричная ОЦМ) находим $\pi_i^* = 1/2$,

$$H^* \{ \xi_1 \} = \log 2 = 1,$$

$$h = -((1 - \alpha) \log (1 - \alpha) + \alpha \log \alpha),$$

$$2b = \log 2 + (1 - \alpha) \log (1 - \alpha) + \alpha \log \alpha = 1 - h.$$

При $\alpha = 1/2$ имеем схему независимых испытаний (удельная энтропия максимальна). При $\alpha \rightarrow 0$ или $\alpha \rightarrow 1$ удельная энтропия $h \rightarrow 0$, что согласуется с фактом уменьшения неопределенности двоичных сообщений.

7.6. ИСТОЧНИКИ НЕПРЕРЫВНЫХ СООБЩЕНИЙ И ИХ ЭНТРОПИЙНЫЕ СВОЙСТВА

До сих пор мы предполагали, что источник сообщений порождает дискретные символные последовательности со значениями в дискретном алфавите. В приложениях, однако, часто встречаются источники непрерывных сообщений $\xi \in \mathcal{A}$, где $\mathcal{A} \subseteq \mathbb{R}$ – некоторое подмножество мощности континуум. Например, речевой сигнал в каждый момент времени можно рассматривать как величину звукового давления ξ в заданной точке пространства (в данном случае $\mathcal{A} = [0, b]$, где b – некоторая максимально допустимая величина давления). Другой пример – оптические сигналы. Для того чтобы рассмотреть эти более сложные случаи, построим обобщенную математическую модель и обобщим понятие энтропии.

Пусть (для фиксированного момента времени) сообщение принимает значение, описываемое случайной величиной $\xi \in \mathcal{A}$, заданной на вероятностном пространстве $(\Omega, \mathcal{F}, \mathbf{P})$ и имеющей распределение вероятностей (индивидуированную вероятностную меру):

$$\mathbf{P}_\xi(B) := \mathbf{P}\{\xi(\omega) \in B\}, B \in \mathcal{B}, \quad (7.70)$$

где \mathcal{B} – борелевская σ -алгебра на числовой прямой; при этом $\mathbf{P}_\xi(\mathcal{A}) = 1$. Для обобщения понятия энтропии потребуем, чтобы на измеримом пространстве $(\mathbb{R}, \mathcal{B})$ кроме $\mathbf{P}_\xi(\cdot)$ была определена еще одна вспомогательная мера $\nu = \nu(B)$, $B \in \mathcal{B}$, такая, что мера $\mathbf{P}_\xi(\cdot)$ абсолютно непрерывна относительно $\nu(\cdot)$. Напомним, что мера $\mathbf{P}_\xi(\cdot)$ называется абсолютно непрерывной относительно $\nu(\cdot)$, если $\forall B \in \mathcal{B} \nu(B) = 0 \Rightarrow \mathbf{P}_\xi(B) = 0$.

Согласно известной теореме Радона – Никодима из условия абсолютной непрерывности меры $\mathbf{P}_\xi(\cdot)$ относительно меры $\nu(\cdot)$ вытекает существование борелевской функции $f(x)$, $x \in \mathcal{A}$, обозначаемой $\frac{d\mathbf{P}_\xi}{d\nu}(x)$ и называемой *производной Радона – Никодима*. Она определена везде в \mathcal{A} , за исключением подмножества \mathcal{A}_0 , где $\nu(\mathcal{A}_0) = 0$, а значит, и $\mathbf{P}_\xi(\mathcal{A}_0) = 0$. При этом справедливо интегральное представление

$$\mathbf{P}_\xi(B) = \int_B f(x)\nu(dx), B \in \mathcal{B}.$$

Энтропией случайного сообщения ξ с распределением $\mathbf{P}_\xi(\cdot)$ называется величина интеграла Лебега

$$H\{\xi\} = - \int_{\mathcal{A}} \log \frac{d\mathbf{P}_\xi}{d\nu}(x)\mathbf{P}_\xi(dx). \quad (7.71)$$

В частности, если \mathcal{A} – дискретное множество, а $\nu(\cdot)$ – «считывающая мера» (например, при $|\mathcal{A}| < \infty$, $\nu(B) = |B|$ – мощность множества B), то (7.71) превращается в определение Шеннона:

$$H\{\xi\} = - \sum_{a \in \mathcal{A}} (\log p_\xi(a)) p_\xi(a), \quad (7.72)$$

где $p_\xi(a) := P\{\xi = a\}$, $a \in \mathcal{A}$ – дискретное распределение вероятностей случайного символа.

Рассмотрим другой частный случай (7.71), когда случайное сообщение ξ имеет абсолютно непрерывное распределение вероятностей (относительно меры Лебега):

$$P_\xi(B) = \int_B p_\xi(x) dx, \quad B \in \mathcal{B}, \quad - \quad (7.73)$$

с плотностью распределения $p_\xi(x) \geq 0$, удовлетворяющей условию нормировки:

$$\int_{\mathcal{A}} p_\xi(x) dx = 1.$$

Учитывая (7.73), в качестве меры $\nu(\cdot)$ в (7.71) примем меру Лебега:

$$\nu(B) = \frac{\text{mes}(B)}{\text{mes}(\mathcal{A})}, \quad B \subseteq \mathcal{A}; \quad (7.74)$$

предполагается, что $\text{mes}(\mathcal{A}) < \infty$.

Тогда из (7.73) и (7.74) следует

$$\frac{dP_\xi}{d\nu}(x) = p_\xi(x) \text{mes}(\mathcal{A}),$$

а интеграл Лебега (7.71) выражается через интеграл Римана:

$$H\{\xi\} = - \log \text{mes}(\mathcal{A}) - \int_{\mathcal{A}} p_\xi(x) \log p_\xi(x) dx. \quad (7.75)$$

Дифференциальной (относительной) энтропией случайного сообщения $\xi \in \mathcal{A}$ с плотностью распределения $p_\xi(x)$ называется значение функционала

$$H_d\{\xi\} = - \int_{\mathcal{A}} p_\xi(x) \log p_\xi(x) dx. \quad (7.76)$$

Замечание 7.1. Термин «относительная» показывает, что она вычислена *относительно* меры Лебега (7.74).

Замечание 7.2. Если «доопределить» функцию плотности на всей числовой прямой:

$$p_\xi(x) = 0, \quad x \in \mathbb{R} \setminus \mathcal{A},$$

то в определении (7.76) всегда будем полагать интегрирование на $\mathbb{R} = (-\infty, \infty)$.

Замечание 7.3. Аналогично (7.76) определяется энтропия и в ситуации, когда случайный символ $\xi \in \mathbb{R}^N$ – N -мерный и описывается N -мерной плотностью распределения $p_\xi(x)$, $x \in \mathbb{R}^N$.

Отметим, что функционалы (7.75) и (7.76) отличаются на константу:

$$\mathbf{H}\{\xi\} = \mathbf{H}_d\{\xi\} - \log \text{mes}(\mathcal{A}).$$

Пользоваться функционалом (7.75) менее удобно, чем (7.76), поскольку при $\mathcal{A} = \mathbb{R}$ получаем $\text{mes}(\mathcal{A}) = +\infty$ и $\mathbf{H}\{\xi\} = -\infty$ для любой плотности $p_\xi(\cdot)$, в то время как дифференциальная энтропия (7.76) конечна. Поэтому свойства обобщенной энтропии будем выражать через свойства дифференциальной.

Свойство 7.5. Дифференциальная энтропия не изменяется при сдвиге распределения вероятностей, «зеркальных отражениях» и «перестановках фрагментов».

Доказательство. Проиллюстрируем схему доказательства для преобразований типа «сдвиг»:

$$\tilde{p}_\xi(x) = p_\xi(x - c),$$

где $c \in \mathbb{R}$ – некоторая константа, определяющая величину сдвига. Тогда из (7.76), делая замену переменных $y = x - c$, имеем

$$\begin{aligned} \mathbf{H}_d\{\tilde{\xi}\} &= - \int_{-\infty}^{\infty} p_\xi(x - c) \log p_\xi(x - c) dx = \\ &= - \int_{-\infty}^{\infty} p_\xi(y) \log p_\xi(y) dy = \mathbf{H}_d\{\xi\}. \end{aligned}$$

□

Для других преобразований доказательство предлагается провести самостоятельно.

Условной дифференциальной энтропией случайного символа $\xi \in \mathbb{R}$ относительно случайного символа $\eta \in \mathbb{R}$ называется величина двойного интеграла:

$$\mathbf{H}_d\{\xi | \eta\} = - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_{\xi, \eta}(x, y) \log p_{\xi|\eta}(x | y) dx dy, \quad (7.77)$$

где $p_{\xi, \eta}(x, y)$ – совместная плотность распределения вероятностей случайных символов (ξ, η) ; $p_{\xi|\eta}(x | y) = p_{\xi, \eta}(x, y) / p_{\eta}(y)$ – условная плотность распределения ξ при условии $\eta = y$. Это определение похоже на (7.19), применимое для ИДС.

Свойство 7.6. Справедливо свойство *иерархической аддитивности* дифференциальной энтропии для произвольной системы случайных символов $(\xi, \eta) \in \mathbb{R}^2$:

$$\mathbf{H}_d\{\xi, \eta\} = \mathbf{H}_d\{\eta\} + \mathbf{H}_d\{\xi | \eta\} = \mathbf{H}_d\{\xi\} + \mathbf{H}_d\{\eta | \xi\}. \quad (7.78)$$

Доказательство. Оно проводится аналогично доказательству подобного свойства в дискретном случае с использованием формулы умножения плотностей и условия нормировки. \square

Лемма 7.1. Для любых плотностей распределения вероятностей $p(x)$, $q(x)$, $x \in \mathbb{R}^N$ выполняется неравенство

$$J(p(\cdot) : q(\cdot)) := \int_{\mathbb{R}^N} p(x) \log \frac{p(x)}{q(x)} dx \geq 0. \quad (7.79)$$

Доказательство. Оно проводится с использованием неравенства Иенсена по аналогии с дискретным случаем (см. теорему 7.3). \square

Пусть $p_X(x)$, $x \in \mathbb{R}$ – произвольная плотность распределения вероятностей, а $a(x, y)$, $x, y \in \mathbb{R}$ – произвольная весовая функция, удовлетворяющая следующим условиям:

$$a(x, y) \geq 0, \quad \int_{-\infty}^{\infty} a(x, y) dx = \int_{-\infty}^{\infty} a(x, y) dy = 1. \quad (7.80)$$

Тогда интегральное преобразование $p_X(\cdot) \rightarrow q(\cdot)$

$$q(y) = \int_{-\infty}^{\infty} a(x, y) p_X(x) dx, \quad y \in \mathbb{R}, \quad (7.81)$$

называется *преобразованием линейного сглаживания (усреднения)*, или *линейной фильтрации*.

Замечание 7.4. Если $a(x, y) = \delta(y - x)$ – обобщенная δ -функция Дирака, то преобразование (7.81) становится тождественным: $q(\cdot) \equiv p_X(\cdot)$.

Замечание 7.5. Легко убедиться, что функция $q(y)$, определяемая (7.81), с учетом (7.80) удовлетворяет свойствам плотности распределения вероятностей.

Свойство 7.7. При линейном сглаживании плотности распределения вероятностей дифференциальная энтропия не убывает.

Доказательство. Пусть случайный символ $\xi \in \mathbb{R}$ имеет исходную плотность $p_X(\cdot)$, а $\eta \in \mathbb{R}$ – сглаженная плотность $q(\cdot)$. Вычислим разность их дифференциальных энтропий с учетом (7.81) и (7.80):

$$\begin{aligned} \Delta H := \mathbf{H}_d\{\eta\} - \mathbf{H}_d\{\xi\} &= - \int_{-\infty}^{\infty} q(y) \log q(y) dy + \int_{-\infty}^{\infty} p_X(x) \log p_X(x) dx = \\ &= \int_{-\infty}^{\infty} p_X(x) \log p_X(x) dx \times \int_{-\infty}^{\infty} a(x, y) dy - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} a(x, y) p_X(x) \log q(y) dxdy = \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} a(x, y) p_X(x) \log \frac{a(x, y) p_X(x)}{a(x, y) q(y)} dxdy. \end{aligned} \quad \square$$

В силу (7.80) $p_1(x, y) = a(x, y)p_X(x) \geq 0$, $q_1(x, y) = a(x, y)q(y) \geq 0$ и удовлетворяют условиям нормировки:

$$\begin{aligned} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p_1(x, y) dxdy &= \int_{-\infty}^{\infty} p_X(x) \int_{-\infty}^{\infty} a(x, y) dydx = 1, \\ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} q_1(x, y) dxdy &= \int_{-\infty}^{\infty} q(y) \int_{-\infty}^{\infty} a(x, y) dxdy = 1 - \end{aligned}$$

и, следовательно, являются двумерными плотностями некоторых распределений вероятностей. Тогда в силу предыдущей леммы имеем неравенство:

$$\Delta H = \int_{\mathbb{R}^2} p_1(x, y) \log \frac{p_1(x, y)}{q_1(x, y)} dxdy \geq 0.$$

Свойство 7.8. Пусть $\xi \in \mathbb{R}^N$ – случайный символ с плотностью распределения $p_\xi(x)$, $x \in \mathbb{R}^N$, имеющий дифференциальную энтропию $\mathbf{H}_d\{\xi\}$, а $y = f(x) : \mathbb{R}^N \rightarrow \mathbb{R}^N$ – взаимно однозначное непрерывно дифференцируемое функциональное преобразование. Тогда случайный символ $\eta = f(\xi)$ имеет дифференциальную энтропию

$$\mathbf{H}_d\{\eta\} = \mathbf{H}_d\{\xi\} + \mathbf{E}\{\log |J_f(\xi)|\}, \quad (7.82)$$

где $J_f(x) = \left| \frac{Df(x)}{Dx} \right|$ – якобиан преобразования $y = f(x)$.

Доказательство. Обозначим $x = f^{-1}(y)$ – обратное функциональное преобразование, а $J_{f^{-1}}(y) = \left| \frac{Df^{-1}(y)}{Dy} \right|$ – его якобиан. По правилам функционального преобразования многомерных случайных величин имеем

$$p_\eta(y) = p_\xi(f^{-1}(y)) |J_{f^{-1}}(y)|. \quad (7.83)$$

Тогда из (7.83) по формуле (7.76)

$$\mathbf{H}_d\{\eta\} = \mathbf{E}\{-\log p_\eta(\eta)\} = \mathbf{E}\{-\log p_\xi(f^{-1}(\eta))\} - \mathbf{E}\{\log |J_{f^{-1}}(\eta)|\}.$$

Воспользовавшись свойством математического ожидания функции от случайных величин и свойством якобиана прямого $y = f(x)$ и обратного $x = f^{-1}(y)$ преобразований, имеем

$$|J_{f^{-1}}(y)|_{y=f(x)} = |J_f(x)|^{-1};$$

$$\mathbf{H}_d\{\eta\} = \mathbf{E}\{-\log p_\xi(\xi)\} - \mathbf{E}\{\log |J_f(\xi)|^{-1}\} = \mathbf{H}_d\{\xi\} + \mathbf{E}\{\log |J_f(\xi)|\},$$

что совпадает с (7.82). \square

Следствие 7.8. При взаимно однозначном функциональном преобразовании $y = f(x) : \mathbb{R}^N \rightarrow \mathbb{R}^N$ дифференциальная энтропия может возрастать, убывать и оставаться неизменной. Она неизменна тогда и только тогда, когда плотность распределения $p_\xi(\cdot)$ и якобиан преобразования $J_f(x)$ обладают специальным свойством:

$$\mathbf{E}\{\log |J_f(\xi)|\} = \int_{\mathbb{R}^N} p_\xi(x) \log |J_f(x)| dx = 0. \quad (7.84)$$

Следствие 7.9. Если функциональное преобразование $f(\cdot)$ линейное:

$$y = f(x) = Ax + b,$$

где $b \in \mathbb{R}^N$ – произвольный вектор; $A = (a_{ij})$ – произвольная невырожденная $(N \times N)$ -матрица, то

$$\mathbf{H}_d\{\eta\} = \mathbf{H}_d\{\xi\} + \log |A|. \quad (7.85)$$

Доказательство. Согласно (7.82) имеем

$$J_f(x) = \left| \frac{Dy}{Dx} \right| = A;$$

$$\mathbf{E}\{\log |J_f(\xi)|\} = \mathbf{E}\{\log |A|\} = \log |A|.$$

Замечание 7.6. Свойство 7.8 определяет существенное различие между энтропией ИДС и дифференциальной энтропией. Как известно, при взаимно однозначных функциональных преобразованиях энтропия ИДС неизменна. Это различие – результат определения дифференциальной энтропии относительно меры Лебега.

Следствие 7.10. В условиях следствия 7.9 дифференциальная энтропия неизменна, если преобразование имеет единичный якобиан: $|A| = 1$.

Следствие 7.11. При ортогональном преобразовании случайного сообщения $\xi \in \mathbb{R}^N$:

$$\eta = A\xi, AA^T = I_N,$$

дифференциальная энтропия не изменяется:

$$H_d\{\eta\} = H_d\{\xi\}. \quad (7.86)$$

Доказательство. Для ортогонального преобразования $|A| = 1$. Поэтому согласно следствию 7.10 из (7.85) получаем (7.86). \square

Пример 7.2. Если случайный символ ξ на $[a, b]$ имеет равномерное распределение $\mathcal{L}\{\xi\} = R[a, b]$, то дифференциальная энтропия равна

$$H_d\{\xi\} = \log(b - a).$$

Пример 7.3. Если случайный символ ξ имеет гауссовское распределение $\mathcal{L}\{\xi\} = \mathcal{N}_1(\mu, \sigma^2)$ со средним μ и дисперсией $\sigma^2 > 0$, то

$$H_d\{\xi\} = \log \sqrt{2\pi e \sigma^2}.$$

Пример 7.4. Если случайное N -символьное ($N \geq 1$) сообщение имеет N -мерное гауссовское распределение $\mathcal{L}\{\xi\} = \mathcal{N}_N(\mu, \Sigma)$ с вектором математического ожидания $\mu = (\mu_i)$ и ковариационной $(N \times N)$ -матрицей $\Sigma = (\sigma_{ij})$, то

$$H_d\{\xi\} = \log \sqrt{(2\pi e)^N |\Sigma|}.$$

В заключение рассмотрим ситуацию, когда имеется стационарный источник непрерывных сообщений, порождающий случайный процесс с дискретным временем:

$$\xi_1, \xi_2, \dots \in \mathbb{R}.$$

Исследуем дифференциальную энтропию отрезка этого процесса длительностью n ($n = 1, 2, \dots$):

$$\Xi_n = (\xi_1, \dots, \xi_n) \in \mathbb{R}^n.$$

Будем предполагать, что исследуемый случайный процесс является *стационарным гауссовским процессом*. Условие гауссности означает, что для любого n распределение вероятностей случайного n -вектора Ξ_n является n -мерным гауссовским с плотностью

$$p_n(x) = (2\pi)^{-n/2} |\Sigma_n|^{-1/2} \exp\left(-\frac{1}{2}(x-a)^T \Sigma_n^{-1} (x-a)\right), \quad x = (x_i) \in \mathbb{R}^n, \quad (7.87)$$

где $a = (a_i) \in \mathbb{R}^n$ – n -вектор-столбец математических ожиданий $E\{\xi_i\} = a_i$; $\Sigma_n = (\sigma_{ij})$ – ковариационная $(n \times n)$ -матрица; $\sigma_{ij} = \text{Cov}\{\xi_i, \xi_j\} = E\{(\xi_i - a_i)(\xi_j - a_j)\}$ – ковариация случайных величин ξ_i, ξ_j ($i, j = 1, n$).

Согласно примеру 7.4 в случае (7.87) имеем

$$H_d \{ \Xi_n \} = \log \sqrt{(2\pi e)^n |\Sigma_n|},$$

поэтому

$$\frac{H_d \{ \Xi_n \}}{n} = \frac{1}{2n} (n \log (2\pi e) + \log |\Sigma_n|) = \log \sqrt{2\pi e} + \frac{1}{2n} \log |\Sigma_n|. \quad (7.88)$$

Свойство стационарности гауссовского случайного процесса проявляется в специальных свойствах ковариационной матрицы Σ_n :

а) дисперсия случайного процесса ξ_t не зависит от времени t , т. е.

$$D \{ \xi_t \} = \text{Cov} \{ \xi_t, \xi_t \} = \sigma_{tt} = \sigma_0 = \text{invar}_t;$$

б) ковариация значений случайного процесса $\xi_t, \xi_{t'}$ зависит лишь от разности моментов времени:

$$\sigma_{t,t'} = \text{Cov} \{ \xi_t, \xi_{t'} \} = \sigma_{|t-t'|}.$$

В силу указанных следствий стационарности исследуемого случайного процесса имеем

$$D_n = |\Sigma_n| = \begin{vmatrix} \sigma_0 & \sigma_1 & \sigma_2 & \dots & \sigma_{n-1} \\ \sigma_1 & \sigma_0 & \sigma_1 & \dots & \sigma_{n-2} \\ \sigma_2 & \sigma_1 & \sigma_0 & \dots & \sigma_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{n-1} & \sigma_{n-2} & \sigma_{n-3} & \dots & \sigma_0 \end{vmatrix}, \quad (7.89)$$

где Σ_n – так называемая *симметричная теплицева матрица*. Математиками Д. Пойа и Г. Сеге было установлено следующее асимптотическое поведение обобщенной дисперсии (7.89) при $n \rightarrow \infty$:

$$\lim_{n \rightarrow \infty} \log(D_n)^{1/n} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log S_\xi(\lambda) d\lambda; \quad (7.90)$$

где

$$S_\xi(\lambda) = \sum_{\tau=-\infty}^{\infty} \sigma_\tau \cos(\lambda\tau), \lambda \in [-\pi, \pi], \quad (7.91)$$

есть *спектральная плотность* случайного процесса ξ_t , определяемая как косинус-преобразование Фурье *ковариационной функции*

$$\sigma_\tau = \sigma_{-\tau} = \text{Cov} \{ \xi_t, \xi_{t+\tau} \}, \tau \in \mathbb{Z}.$$

Введем *нормированную корреляционную функцию*

$$\rho_\tau = \frac{\sigma_\tau}{\sigma_0} = \text{Corr} \{ \xi_t, \xi_{t+\tau} \}, \tau \in \mathbb{Z}, \quad (7.92)$$

и нормированную спектральную плотность

$$s_\xi(\lambda) = \frac{S_\xi(\lambda)}{\sigma_0} = \sum_{\tau=-\infty}^{\infty} \rho_\tau \cos(\lambda\tau), \lambda \in [-\pi, \pi]. \quad (7.93)$$

Используя (7.90)–(7.93), найдем удельную энтропию гауссовского стационарного случайного процесса. Из (7.88) с помощью эквивалентных преобразований имеем

$$\frac{\mathbf{H}_d\{\Xi_n\}}{n} = \log \sqrt{2\pi e \sigma_0} + \frac{1}{2n} \log \left| \frac{1}{\sigma_0} \Sigma_n \right|.$$

Согласно (7.89) и (7.92) элементами матрицы $\frac{1}{\sigma_0} \Sigma_n$ являются значения нормированной корреляционной функции:

$$d_n = \left| \frac{1}{\sigma_0} \Sigma_n \right| = \begin{vmatrix} \rho_0 & \rho_1 & \rho_2 & \dots & \rho_{n-1} \\ \rho_1 & \rho_0 & \rho_1 & \dots & \rho_{n-2} \\ \rho_2 & \rho_1 & \rho_0 & \dots & \rho_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{n-1} & \rho_{n-2} & \rho_{n-3} & \dots & \rho_0 \end{vmatrix}, \quad \rho_0 = 1.$$

Тогда в силу (7.90) и (7.93)

$$\lim_{n \rightarrow \infty} \log(d_n)^{1/n} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log s_\xi(\lambda) d\lambda.$$

Получаем выражение удельной энтропии:

$$\begin{aligned} h &= \lim_{n \rightarrow \infty} \frac{\mathbf{H}_d\{\Xi_n\}}{n} = \log \sqrt{2\pi e \sigma_0} + \frac{1}{4\pi} \int_{-\pi}^{\pi} \log s_\xi(\lambda) d\lambda = \\ &= \mathbf{H}\{\xi_1\} + \frac{1}{4\pi} \int_{-\pi}^{\pi} \log s_\xi(\lambda) d\lambda, \end{aligned} \quad (7.94)$$

где $\mathbf{H}\{\xi_1\} = \log \sqrt{2\pi e \sigma_0}$ – энтропия единичного гауссовского случайного символа с дисперсией σ_0 .

В (7.94) первое слагаемое характеризует энтропию единичного (отдельного, изолированного) символа, а второе – зависимость символов в последовательности и выражается через нормированную спектральную плотность $s_\xi(\lambda)$ или, что эквивалентно, через нормированную корреляционную функцию.

Пример 7.5. Последовательность независимых гауссовых символов:

$$\mathbf{Cov}\{\xi_t, \xi_{t'}\} = \sigma_0 \delta_{t,t'}.$$

При этом $\rho_\tau = \delta_{\tau,0}$, $s_\xi(\lambda) = 1 \equiv \text{const}$, $\lambda \in [-\pi, \pi]$. Тогда в силу (7.91) второе слагаемое – нулевое, и получаем установленный выше результат:

$$\mathbf{H}_d\{\Xi_n\} = n\mathbf{H}\{\xi_1\}, \quad h = \mathbf{H}\{\xi_1\}.$$

Пример 7.6. Пусть имеет место марковская корреляционная зависимость: $\sigma_\tau = \sigma_0 e^{-\alpha|\tau|}$, $\alpha > 0$. Тогда

$$D_n = \sigma_0^n (1 - \rho^2)^{n-1},$$

где $\rho = e^{-\alpha}$. Поэтому из (7.94) имеем

$$h = \mathbf{H}\{\xi_1\} + \log \sqrt{1 - \rho^2} = \mathbf{H}\{\xi_1\} + \log \sqrt{1 - e^{-2\alpha}}.$$

7.7. ОПТИМИЗАЦИЯ ФУНКЦИОНАЛА ЭНТРОПИИ НА КЛАССЕ ВЕРОЯТНОСТНЫХ РАСПРЕДЕЛЕНИЙ

Для криптологических применений важно исследовать случаи экстремальных значений функционала энтропии. Как видно из примеров 7.2–7.4 п. 7.6, дифференциальная энтропия изменяется от $-\infty$ до $+\infty$. Для ИДС $\xi \in A$, как было установлено в п. 7.2, минимальное значение, равное нулю ($\mathbf{H}_{\min} = 0$), – для вырожденного дискретного распределения вероятностей, а максимальное значение энтропии, равное $\mathbf{H}_{\max} = \log |A|$, достигалось для дискретного равномерного распределения вероятностей. Как видим, имеются существенные различия дискретного и непрерывного случаев. Чтобы максимальные значения дифференциальной энтропии были конечны, будем осуществлять ее максимизацию на заданном ограниченном классе вероятностных распределений \mathcal{P} :

$$\mathbf{H}_d\{\xi\} = - \int_{-\infty}^{\infty} p(x) \log p(x) dx \rightarrow \max_{p(\cdot) \in \mathcal{P}}. \quad (7.95)$$

Исследуем три наиболее часто встречающихся в прикладных задачах класса абсолютно непрерывных вероятностных распределений.

Класс $\mathcal{P}_1(a, b)$:

$$\begin{aligned} \mathcal{P}_1(a, b) = & \left\{ p(x), x \in \mathbb{R}: \right. \\ & p(x) \geq 0, \int_{-\infty}^{\infty} p(x) dx = 1; p(x) = 0, x \notin [a, b] \left. \right\} - \end{aligned} \quad (7.96)$$

есть семейство одномерных плотностей распределения с конечным носителем $[a, b]$, $-\infty < a < b < \infty$.

Класс $\mathcal{P}_2(a, \sigma^2)$:

$$\begin{aligned} \mathcal{P}_2(a, \sigma^2) = & \left\{ p(x), x \in \mathbb{R} : p(x) \geq 0, \int_{-\infty}^{\infty} p(x) dx = 1, \right. \\ & \left. \int_{-\infty}^{\infty} xp(x) dx = a, \int_{-\infty}^{\infty} (x - a)^2 p(x) dx \leq \sigma^2 \right\} - \end{aligned} \quad (7.97)$$

есть семейство одномерных плотностей с конечными моментами первого и второго порядков: заданным математическим ожиданием (средним) $E\{\xi\} = a$ и ограниченной дисперсией $D\{\xi\} \leq \sigma^2$.

Класс $\mathcal{P}_3(n, \mu, \Sigma)$:

$$\begin{aligned} \mathcal{P}_3(n, \mu, \Sigma) = & \left\{ p(x), x \in \mathbb{R}^n : p(x) \geq 0, \int_{\mathbb{R}^n} p(x) dx = 1, \right. \\ & \left. \int_{\mathbb{R}^n} xp(x) dx = \mu, \int_{\mathbb{R}^n} (x - \mu)(x - \mu)^T p(x) dx = \Sigma \right\} - \end{aligned} \quad (7.98)$$

есть семейство n -мерных плотностей распределения с фиксированным n -вектором математического ожидания $\mu = (\mu_i)$ и невырожденной ($n \times n$)-ковариационной матрицей $\Sigma = (\sigma_{ij})$, $|\Sigma| \neq 0$.

Теорема 7.11. Для любого случайного символа $\xi \in \mathbb{R}$ с плотностью распределения $p(\cdot) \in \mathcal{P}_1(a, b)$ дифференциальная энтропия удовлетворяет неравенству

$$H_d\{\xi\} \leq \log(b - a), \quad (7.99)$$

причем верхняя граница, т. е. максимум дифференциальной энтропии по классу $\mathcal{P}_1(a, b)$, достигается в случае равномерного на $[a, b]$ распределения вероятностей $R[a, b]$ с плотностью

$$p^*(x) = \frac{1}{b - a} \mathbf{1}_{[a, b]}(x), \quad x \in \mathbb{R}, \quad (7.100)$$

где $\mathbf{1}_A(x)$ – индикаторная функция множества A .

Доказательство. Будем решать экстремальную задачу (7.95), (7.96) при $\mathcal{P} = \mathcal{P}_1(a, b)$. Без учета условия неотрицательности $p(\cdot) \geq 0$ в (7.96) эта задача эквивалентна следующей задаче вариационного исчисления с ограничением типа равенства:

$$\begin{aligned} \mathbf{H}_d \{\xi\} &= - \int_a^b p(x) \log p(x) dx \rightarrow \max_{p(\cdot)}, \\ \int_a^b p(x) dx &= 1. \end{aligned} \tag{7.101}$$

Для решения задачи (7.101) применим метод неопределенных множителей Лагранжа. Для этого составим функционал Лагранжа

$$L(p(\cdot), \lambda) = \int_a^b (-p(x) \log p(x) + \lambda p(x)) dx,$$

где λ – неопределенный множитель Лагранжа; и запишем необходимое условие максимума:

$$\begin{cases} \delta L = \int_a^b (-1 - \log p(x) + \lambda) \delta p(x) dx = 0, \\ \int_a^b p(x) dx = 1, \end{cases} \tag{7.102}$$

где $\delta p(x)$ – вариация функции $p(\cdot)$; а δL – первая вариация функционала $L(\cdot)$. Поскольку $\delta p(x)$ – произвольная вариация, то система уравнений (7.102) примет эквивалентный вид:

$$\begin{cases} -1 - \log p(x) + \lambda = 0, \\ \int_a^b p(x) dx = 1. \end{cases} \tag{7.103}$$

Решим первое уравнение (7.103):

$$p(x) = 2^{\lambda-1} = \text{const} = \mu, \quad x \in [a, b],$$

и, подставляя это решение во второе уравнение (7.103), получим единственное решение задачи (7.101) в виде (7.100). Заметим, что найденное решение удовлетворяет снятому ограничению $p(\cdot) \geq 0$.

Подставляя (7.100) в целевую функцию (7.101), получим

$$\max_{p(\cdot)} \mathbf{H}_d \{\xi\} = \log(b-a),$$

что и означает (7.99). □

Теорема 7.12. Для любого случайного символа $\xi \in \mathbb{R}$ с плотностью распределения $p(\cdot) \in \mathcal{P}_2(a, \sigma^2)$ дифференциальная энтропия удовлетворяет неравенству

$$H_d\{\xi\} \leq \log \sqrt{2\pi e \sigma^2}, \quad (7.104)$$

причем верхняя граница в (7.104), т. е. максимум дифференциальной энтропии на классе $\mathcal{P}_2(a, \sigma^2)$, достигается в случае гауссовского (нормального) распределения вероятностей $N_1(a, \sigma^2)$ с плотностью

$$p^*(x) = n_1(x | a, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-a)^2}{2\sigma^2}}, \quad x \in \mathbb{R}. \quad (7.105)$$

Доказательство. Имеется два способа доказательства. Первый основан на решении задачи вариационного исчисления (7.95) при $\mathcal{P} = \mathcal{P}_2(a, \sigma^2)$ методом неопределенных множителей Лагранжа подобно доказательству предыдущей теоремы. Второй метод менее громоздкий и излагается ниже.

Получим сначала вспомогательное неравенство – оценку сверху для интеграла. Имеем

$$-\int_{-\infty}^{\infty} p(x) \log \left(\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-a)^2}{2\sigma^2}} \right) dx = \log \sqrt{2\pi\sigma^2} + \frac{\log e}{2\sigma^2} \int_{-\infty}^{\infty} (x-a)^2 p(x) dx.$$

В силу (7.97) последний интеграл равен дисперсии $D\{\xi\} \leq \sigma^2$, поэтому

$$-\int_{-\infty}^{\infty} p(x) \log n_1(x | a, \sigma^2) dx \leq \log \sqrt{2\pi e \sigma^2}, \quad p(\cdot) \in \mathcal{P}_2(a, \sigma^2).$$

Из этого неравенства имеем оценку разности левой и правой частей (7.104):

$$\begin{aligned} H_d\{\xi\} - \log \sqrt{2\pi e \sigma^2} &\leq - \int_{-\infty}^{\infty} p_X(x) \log p_X(x) x dx + \\ &+ \int_{-\infty}^{\infty} p_X(x) \log n_1(x | a, \sigma^2) dx = - \int_{-\infty}^{\infty} p_X(x) \log \frac{p_X(x)}{n_1(x | a, \sigma^2)} dx = -J(p(\cdot) : n_1(\cdot)). \end{aligned}$$

В силу леммы 7.1 $J(p(\cdot) : n_1(\cdot)) \geq 0$, следовательно

$$H_d\{\xi\} - \log \sqrt{2\pi e \sigma^2} \leq 0,$$

что эквивалентно (7.104).

Как установлено в п. 7.6, для гауссовского распределения (7.105) энтропия равна

$$H_d\{\xi^*\} = \log \sqrt{2\pi e \sigma^2},$$

т. е. (7.104) обращается в равенство. \square

Теорема 7.13. Для любой случайной n -символьной последовательности $\Xi_n = (\xi_i) \in \mathbb{R}^n$ с n -мерной плотностью распределения вероятностей $p(\cdot) \in \mathcal{P}_3(n, \mu, \Sigma)$ дифференциальная энтропия удовлетворяет неравенству

$$\mathbf{H}_d\{\Xi_n\} \leq \log \sqrt{(2\pi e)^n |\Sigma|}, \quad (7.106)$$

причем верхняя граница, т. е. максимум дифференциальной энтропии по классу $\mathcal{P}_3(n, \mu, \Sigma)$, достигается в случае n -мерного гауссовского (нормального) распределения вероятностей $N_n(\mu, \Sigma)$ с плотностью

$$p^*(x) = n_n(x | \mu, \Sigma) = (2\pi)^{-n/2} |\Sigma|^{-1/2} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right). \quad (7.107)$$

Доказательство. Воспользуемся тем же методом доказательства, что и в теореме 7.12. Сначала получим вспомогательное равенство для n -кратного интеграла, используя свойства семейства $\mathcal{P}_3(n, \mu, \Sigma)$ и формулу (7.107):

$$\begin{aligned} Q &= - \int_{\mathbb{R}^n} p_X(x) \log p^*(x) dx = \log \sqrt{(2\pi)^n |\Sigma|} + \\ &\quad + \frac{\log e}{2} \int_{\mathbb{R}^n} (x - \mu)^T \Sigma^{-1} (x - \mu) p_X(x) dx = \\ &= \log \sqrt{(2\pi)^n |\Sigma|} + \frac{\log e}{2} \mathbf{E} \left\{ \text{tr} \left(\Sigma^{-1} (x - \mu)(x - \mu)^T \right) \right\} = \\ &= \log \sqrt{(2\pi)^n |\Sigma|} + \frac{\log e}{2} \text{tr} \left(\Sigma^{-1} \mathbf{E} \left\{ (x - \mu)(x - \mu)^T \right\} \right) = \log \sqrt{(2\pi e)^n |\Sigma|}. \end{aligned}$$

Оценим с учетом этого равенства разность левой и правой частей (7.106):

$$\mathbf{H}_d\{\xi\} - \log \sqrt{(2\pi e)^n |\Sigma|} = \mathbf{H}_d\{\xi\} - Q = - \int_{\mathbb{R}^n} p_X(x) \log \frac{p_X(x)}{p^*(x)} dx \leq 0.$$

Здесь использована лемма 7.1. Как установлено в п. 7.6, для многомерного гауссовского распределения (7.107) дифференциальная энтропия равна

$$\mathbf{H}_d\{\Xi_n^*\} = \log \sqrt{(2\pi e)^n |\Sigma|},$$

т. е. (7.106) обращается в равенство. \square

7.8. АСИМПТОТИЧЕСКИЕ СВОЙСТВА СТАЦИОНАРНОГО ИСТОЧНИКА ДИСКРЕТНЫХ СООБЩЕНИЙ

Пусть рассматривается стационарный ИДС без памяти с алфавитом $A = V = \{0, 1\}$, порождающий случайную последовательность независимых в совокупности, одинаково распределенных двоичных символов: $\Xi_n = (\xi_1, \dots, \xi_n)$, $n = 1, 2, \dots$, где $\xi_i \in \{0, 1\}$ – двоичный случайный символ с распределением вероятностей Бернулли ($i = \overline{1, n}$):

$$\mathbf{P}\{\xi_i = 1\} = p, \mathbf{P}\{\xi_i = 0\} = q = 1 - p, \quad (7.108)$$

где $0 < p < 1/2$ – вероятность появления «1» (случай $p > 1/2$ сводится к этому случаю переобозначением символов $0 \leftrightarrow 1$).

Количество реализаций Ξ_n равно 2^n . В силу схемы независимых испытаний порождения символов, распределение вероятностей Ξ_n задается соотношением

$$p_n(a) = p_n(a_1, \dots, a_n) := \mathbf{P}\{\Xi_n = a\} = p^b q^{n-b}, b = \sum_{i=1}^n a_i, \quad (7.109)$$

или

$$\log p_n(a) = n \log q - \sum_{i=1}^n a_i \log \frac{q}{p} = n \log q - b \log \frac{q}{p},$$

где $a = (a_i) \in V_n$, $b = \sum_{i=1}^n a_i$ – суммарное количество «1» в двоичном n -векторе a или значение случайной величины

$$\eta_n = \sum_{i=1}^n \xi_i = |\Xi_n|^2. \quad (7.110)$$

Значения вероятностей (7.109) существенно изменяются при изменении величины b . Отношение наибольшей из этих вероятностей к наименьшей равно

$$\kappa_n = \frac{\max_a p_n(a)}{\min_a p_n(a)} = \frac{q^n}{p^n} = \left(\frac{q}{p}\right)^n > 1, \quad (7.111)$$

и эта величина экспоненциально растет с ростом n .

С учетом (7.108) и (7.110) вычислим моменты первого и второго порядка для случайной величины η_n :

$$\mathbf{E}\{\eta_n\} = n\mathbf{E}\{\xi_1\} = np < n/2, \quad \mathbf{D}\{\eta_n\} = n\mathbf{D}\{\xi_1\} = npq. \quad (7.112)$$

В силу (7.110) уклонение случайного числа единиц в Ξ_n от его среднего значения

$$\zeta_n = \eta_n - np \quad (7.113)$$

имеет нулевые среднее и среднеквадратическое отклонения

$$\sigma_{\zeta_n} = \sqrt{\mathbf{D}\{\zeta_n\}} = \sqrt{npq},$$

а для относительного уклонения

$$\delta_n = \zeta_n/n \quad (7.114)$$

имеем

$$\mathbf{E}\{\delta_n\} = 0, \sigma_{\delta_n} = \sqrt{\mathbf{D}\{\delta_n\}} = \sqrt{\frac{pq}{n}}. \quad (7.115)$$

Как видно из (7.115), среднеквадратичное отклонение для случайной величины δ_n при $n \rightarrow \infty$ убывает как $1/\sqrt{n}$. Если $|b_n - np| = c\sqrt{npq}$, $c > 0$, то различие наибольшей и наименьшей вероятностей на этом подмножестве значений Ξ_n по-прежнему достаточно велико: $\kappa_n = (q/p)^{c\sqrt{npq}}$. Однако эта величина растет при $n \rightarrow \infty$ значительно медленнее, чем $(1/p)^{c\sqrt{npq}}$. Следовательно, справедливо неравенство

$$\left| \log \frac{p_n(a)}{p_n(a')} \right|_{|a'|^2 = |a|^2 - c\sqrt{npq}} \ll \log \frac{1}{p_n(a)}.$$

Сформулируем это свойство в виде теоремы.

Теорема 7.14 (теорема о высоковероятном подмножестве). *Множество всех 2^n реализаций определенного выше двоичного случайного вектора $\Xi_n \in V_n$ можно разбить на два непересекающихся подмножества:*

$$V_2^n = A_n \cup B_n, A_n \cap B_n = \emptyset, \quad (7.116)$$

так что при $n \rightarrow \infty$ выполняются свойства:

а) множество A_n имеет исчезающую малую вероятность

$$\mathbf{P}\{\Xi_n \in A_n\} = \sum_{a \in A_n} p_n(a) \rightarrow 0; \quad (7.117)$$

б) реализации из множества B_n становятся относительно равновероятными:

$$\left| \frac{\log p_n(a) - \log p_n(a')}{\log p_n(a)} \right| \rightarrow 0, \quad a, a' \in B_n. \quad (7.118)$$

Доказательство. Воспользуемся неравенством Чебышева (относительно дисперсий) для случайной величины η_n , определяемой (7.110) с учетом (7.112):

$$\forall \varepsilon > 0 : \mathbf{P}\{|\eta_n - np| \geq \varepsilon\} \leq \frac{\mathbf{D}\{\eta_n\}}{\varepsilon^2} = \frac{npq}{\varepsilon^2}.$$

Полагая $\varepsilon = n^{3/4}$, получим

$$\mathbf{P}\{|\eta_n - np| \geq n^{3/4}\} \leq \frac{pq}{\sqrt{n}}. \quad (7.119)$$

Построим разбиение множества V_n следующим образом:

$$\begin{aligned} A_n &= \left\{ a = (a_i) \in V_n : \left| \sum_{i=1}^n a_i - np \right| \geq n^{3/4} \right\}; \\ B_n &= V_n \setminus A_n = \left\{ a = (a_i) \in V_n : \left| \sum_{i=1}^n a_i - np \right| < n^{3/4} \right\}. \end{aligned} \quad (7.120)$$

Тогда из (7.119) и (7.120) имеем

$$\mathbf{P}(A_n) = \mathbf{P}\{\Xi_n \in A_n\} \leq \frac{p q}{\sqrt{n}} \rightarrow 0, \quad n \rightarrow \infty. \quad (7.121)$$

Следовательно, выполняется (7.117).

В силу (7.120), если $a \in B_n$, то

$$np - n^{3/4} < \sum_{i=1}^n a_i < np + n^{3/4},$$

поэтому согласно (7.109)

$$n \log q - \log \frac{q}{p} (np + n^{3/4}) < \log p_n(a) < n \log q - \log \frac{q}{p} (np - n^{3/4}),$$

или

$$\begin{aligned} nq \log q + np \log p - n^{3/4} \log (q/p) &< \log p_n(a) < \\ &< nq \log q + np \log p + n^{3/4} \log (q/p), \quad a \in B_n; \\ n(q \log q + p \log p) - n^{3/4} \log (q/p) &< \log p_n(a) < \\ &< n(q \log q + p \log p) + n^{3/4} \log (q/p). \end{aligned} \quad (7.122)$$

Отсюда имеем

$$\forall a, a' \in B_n : |\log p_n(a) - \log p_n(a')| < 2n^{3/4} \log \frac{q}{p}; \quad (7.123)$$

$$-\log p_n(a) = |\log p_n(a)| > -n(p \log p + q \log q) - n^{3/4} \log \frac{q}{p}. \quad (7.124)$$

Тогда из неравенств (7.123) и (7.124) следует

$$\begin{aligned} \left| \frac{\log p_n(a) - \log p_n(a')}{\log p_n(a)} \right| &< \frac{2n^{3/4} \log (q/p)}{-n(p \log p + q \log q) - n^{3/4} \log (q/p)} = \\ &= \frac{2n^{-1/4} \log (q/p)}{-p \log p - q \log q - n^{-1/4} \log (q/p)} \rightarrow 0, \quad n \rightarrow \infty, \quad a, a' \in B_n, \end{aligned}$$

что совпадает с (7.118). □

Теорема 7.15. Пусть $B_n \subset V_n$ – «высоковероятное» подмножество реализаций, описанное в теореме 7.14. Тогда мощность этого подмножества $M_n = |B_n|$ при $n \rightarrow \infty$ удовлетворяет асимптотике

$$\frac{\log M_n}{n} \rightarrow H\{\xi_1\} = -(p \log p + q \log q). \quad (7.125)$$

Доказательство. Согласно (7.109) и (7.120)

$$P\{\Xi_n \in B_n\} = 1 - P\{\Xi \in A_n\} = \sum_{a \in B_n} p_n(a). \quad (7.126)$$

В силу (7.121) из (7.126) имеем

$$1 \geq P\{\Xi_n \in B_n\} \geq 1 - \frac{pq}{\sqrt{n}}. \quad (7.127)$$

С другой стороны, с учетом (7.122)

$$p_n(a) < 2^{n(p \log p + q \log q) + n^{3/4} \log(q/p)} = 2^{-nH\{\xi_1\} + n^{3/4} \log(q/p)}.$$

Поэтому, используя второе равенство в (7.126), находим

$$P\{\Xi_n \in B_n\} < M_n 2^{-nH\{\xi_1\} + n^{3/4} \log(q/p)}. \quad (7.128)$$

Из (7.127) и (7.128) имеем

$$M_n 2^{-nH\{\xi_1\} + n^{3/4} \log(q/p)} > 1 - \frac{pq}{\sqrt{n}}.$$

Следовательно,

$$M_n > \left(1 - \frac{pq}{\sqrt{n}}\right) 2^{nH\{\xi_1\} - n^{3/4} \log(q/p)}. \quad (7.129)$$

Аналогично (7.129) получим оценку снизу для M_n . Воспользуемся в (7.122) и (7.127) левыми неравенствами:

$$1 \geq M_n 2^{-nH\{\xi_1\} - n^{3/4} \log(q/p)}.$$

Отсюда имеем

$$M_n \leq 2^{nH\{\xi_1\} + n^{3/4} \log(q/p)}. \quad (7.130)$$

Логарифмируя (7.129) и (7.130), деля на n и объединяя в совместное неравенство, получаем двухстороннюю оценку для $\log M_n/n$:

$$H\{\xi_1\} - n^{-1/4} \log(q/p) < \frac{\log M_n}{n} \leq H\{\xi_1\} + n^{-1/4} \log(q/p).$$

Устремляя $n \rightarrow \infty$, приходим к (7.125). \square

Теоремы 7.14 и 7.15 легко обобщаются для стационарного ИДС без памяти, у которого алфавит A имеет мощность $N = |A| > 2$. При этом $\Xi_n \in A^n$ принимает одно из N^n возможных различных значений. Согласно теоремам 7.14 и 7.15 внимания заслуживают лишь $M_n \approx 2^{nH\{\xi_1\}}$ реализаций, которые можно считать равновероятными. Если распределение вероятностей $p_1(a_1)$, $a_1 \in A$ является равномерным: $p_1(a_1) = P\{\xi_1 = a_1\} = 1/N$, $a_1 \in A$, то $H\{\xi_1\} = \log N$ (формула Хартли) и $2^{nH\{\xi_1\}} = N^n$, т. е. $M_n = |A|^n$. Однако если распределение вероятностей отлично от равномерного, то $H\{\xi_1\} < \log N$ и доля заслуживающих внимания реализаций

$$u_n = \frac{|B_n|}{|A|^n} = \frac{2^{nH\{\xi_1\}}}{N^n} = \left(\frac{2^{H\{\xi_1\}}}{N} \right)^n = \left(2^{H\{\xi_1\} - \log N} \right)^n \quad (7.131)$$

неограниченно уменьшается с ростом n . Следовательно, подавляющее большинство реализаций при этом несущественные, и их можно отбросить. Этот факт лежит в основе теории кодирования сообщений и широко используется в криптологии.

Пример 7.7. Пусть рассматривается стационарный ИДС без памяти с двоичным алфавитом $A = V$ ($N = 2$) и вероятностью появления единичного символа $0 \leq p \leq 1/2$. Тогда согласно теореме 7.14 «высоковероятное множество» двоичных последовательностей имеет вид

$$B_n = \left\{ a = (a_1, \dots, a_n) \in V_n : \left| \frac{1}{n} \sum_{i=1}^n a_i - p \right| < \frac{1}{n^{1/4}} \right\}.$$

Например, при $n = 10^4$ множество B_n состоит из двоичных последовательностей, для которых доля единиц заключена в следующих пределах: $p - 0,1 < \frac{1}{n} \sum_{i=1}^n a_i < p + 0,1$. При $p = 0,1$ количество единиц в таких двоичных

последовательностях $0 \leq \sum_{i=1}^n a_i \leq 2000$. Доля этих «заслуживающих внимания» (по теореме 7.14) последовательностей составляет согласно (7.131) величину

$$u_n = \left(\frac{2^{-(p \log p + (1-p) \log(1-p))}}{2} \right)^n = (F(p))^n,$$

где $F(p) = 2^{-(p \log p + (1-p) \log(1-p)+1)}$. Эта функция протабулирована ниже:

p	0,1	0,2	0,3	0,4	0,5
$F(p)$	0,692	0,825	0,921	0,980	1,000

Например, при $p = 0,1$ и $n = 100$ доля «заслуживающих внимания» реализаций составляет $u_n = 2^{-0,531n} \approx 10^{-16}$.

7.9. ЭНТРОПИЙНАЯ УСТОЙЧИВОСТЬ СЛУЧАЙНЫХ СИМВОЛЬНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Установленные в предыдущем пункте асимптотические свойства разбиения множества реализаций для стационарных ИДС без памяти могут быть обобщены на класс зависимых случайных символьных последовательностей Ξ_n . При этом обобщении нам понадобится понятие *энтропийной устойчивости случайной последовательности*.

Случайная n -символьная последовательность $\Xi_n = (\xi_1, \dots, \xi_n) \in A^n$ с n -мерным дискретным распределением вероятностей

$$p_n(a) = \mathbf{P}\{\Xi_n = a\}, a = (a_i) \in A^n \quad (7.132)$$

и неубывающей по n энтропией $0 < \mathbf{H}\{X\} < \infty$

$$\mathbf{H}\{X\} = \mathbf{E}\{-\log p_n(\Xi_n)\} = -\sum_{a \in A^n} p_n(a) \log p_n(a) \quad (7.133)$$

называется *энтропийно устойчивой*, если при увеличении числа символов ($n \rightarrow \infty$) имеет место сходимость по вероятности

$$\frac{-\log p_n(\Xi_n)}{\mathbf{H}\{X\}} \xrightarrow{\mathbf{P}} 1, \quad (7.134)$$

т. е. $\forall \varepsilon > 0 \exists \bar{n} = \bar{n}(\varepsilon)$ такое, что при любом $n \geq \bar{n}(\varepsilon)$ выполняется неравенство

$$\mathbf{P}\left\{\left|\frac{-\log p_n(\Xi_n)}{\mathbf{H}\{X\}} - 1\right| \geq \varepsilon\right\} < \varepsilon; \quad (7.135)$$

при этом функция $\bar{n}(\varepsilon)$ – монотонно убывающая.

Отметим, что для реальных прикладных задач криптологии энтропия возрастает с ростом n : $\mathbf{H}\{X\} \rightarrow \infty$. Сформулируем важнейший результат, полученный в 1967 г. Р. П. Стратоновичем.

Теорема 7.16 (обобщенная теорема Стратоновича). *Если $\Xi_n \in A^n$ – произвольная случайная n -символьная последовательность, удовлетворяющая свойству энтропийной устойчивости, то ее множество N^n реализаций A^n можно разбить на два непересекающихся подмножества A_n и B_n таким образом, что при $n \rightarrow \infty$ выполняются следующие асимптотические свойства:*

1) суммарная вероятность реализаций подмножества A_n исчезающе мала:

$$\mathbf{P}\{\Xi_n \in A_n\} = \sum_{a \in A_n} p_n(a) \rightarrow 0; \quad (7.136)$$

2) реализации высоковероятного подмножества B_n становятся асимптотически относительно равновероятными в следующем смысле:

$$\left| \frac{\log p_n(a) - \log p_n(a')}{\log p_n(a)} \right| \rightarrow 0, \quad a, a' \in B_n; \quad (7.137)$$

3) количество $M_n = |B_n|$ реализаций (мощность) множества B_n связано с энтропией (последовательности) $\mathbf{H}\{X\}$ асимптотическим соотношением

$$\frac{\log M_n}{\mathbf{H}\{X\}} \rightarrow 1. \quad (7.138)$$

Доказательство. Оно состоит из трех частей.

1. В силу свойства энтропийной устойчивости (7.134) имеем неравенство (7.135), в котором для некоторого натурального m положим $\varepsilon = 1/m$. Тогда для любых $m = 1, 2, \dots, n$, $n \geq \bar{n}(1/m)$ из (7.135) имеем

$$\mathbf{P} \left\{ \left| \frac{-\log p_n(\Xi_n)}{\mathbf{H}\{X\}} - 1 \right| \geq \frac{1}{m} \right\} < \frac{1}{m}. \quad (7.139)$$

В силу указанного выше свойства монотонности $\bar{n}(\varepsilon)$ выберем $m = m_n \in \mathbb{N}$ так, чтобы $\bar{n}(1/m_n, 1/m_n) \leq n \leq \bar{n}(1/(m_n + 1), 1/(m_n + 1))$, и определим подмножества A_n и B_n , используя (7.139):

$$\begin{aligned} A_n &= \{a = (a_i) \in A^n : |-\log p_n(\Xi_n)/\mathbf{H}\{X\} - 1| \geq 1/m_n\}; \\ B_n &= A^n \setminus A_n. \end{aligned} \quad (7.140)$$

С учетом (7.139) и (7.140) получаем (7.136):

$$\mathbf{P}\{\Xi_n \in A_n\} < \frac{1}{m_n} \rightarrow 0, \quad \mathbf{P}\{\Xi_n \in B_n\} > 1 - \frac{1}{m_n} \rightarrow 1,$$

поскольку очевидно, что $m_n \rightarrow \infty$ при $n \rightarrow \infty$.

2. В силу (7.140) $\forall a \in B_n$ имеем

$$\left(1 - \frac{1}{m_n}\right)\mathbf{H}\{X\} < -\log p_n(a) < \left(1 + \frac{1}{m_n}\right)\mathbf{H}\{X\} \quad (7.141)$$

или

$$1 - \frac{1}{m_n} < \frac{-\log p_n(a)}{\mathbf{H}\{X\}} < 1 + \frac{1}{m_n}. \quad (7.142)$$

В силу (7.141) $\forall a, a' \in B_n$ имеем

$$\left| \frac{\log p_n(a) - \log p_n(a')}{-\log p_n(a)} \right| \leq \frac{2\mathbf{H}\{X\}/m_n}{-\log p_n(a)}.$$

Согласно (7.142) $-\mathbf{H}\{X\} / -\log p_n(a) < 1/(1-1/m_n)$, поэтому получаем неравенство

$$\left| \frac{\log p_n(a) - \log p_n(a')}{-\log p_n(a)} \right| < \frac{2/m_n}{1-1/m_n} = \frac{2}{m_n-1} \rightarrow 0,$$

что доказывает (7.136).

3. Для доказательства (7.138) запишем (7.141) $\forall a \in B_n$ в эквивалентном виде:

$$2^{-(1+1/m_n)\mathbf{H}\{X\}} < p_n(a) < 2^{-(1-1/m_n)\mathbf{H}\{X\}}. \quad (7.143)$$

Поскольку

$$1 \geq \mathbf{P}\{\Xi_n \in B_n\} = \sum_{a \in B_n} p_n(a) \geq 1 - \frac{1}{m_n},$$

то с учетом (7.143) (как при доказательстве теоремы 7.15)

$$\left(1 - \frac{1}{m_n}\right)2^{(1-1/m_n)\mathbf{H}\{X\}} < M_n < 2^{(1+1/m_n)\mathbf{H}\{X\}}.$$

Проводя логарифмирование и деление частей этого неравенства на $\mathbf{H}\{X\}$, получаем

$$1 - \frac{1}{m_n} + \frac{\log(1-1/m_n)}{\mathbf{H}\{X\}} < \frac{\log M_n}{\mathbf{H}\{X\}} < 1 + \frac{1}{m_n}.$$

Поскольку $\mathbf{H}\{X\}$ с ростом n не убывает, то из этих неравенств при $n \rightarrow \infty$ заключаем (7.138). \square

Заметим, что проверка основного условия теоремы 7.16 – условия энтропийной устойчивости (7.134) – на практике затруднительна. В связи с этим сформулируем ряд легко проверяемых на практике достаточных условий, влекущих выполнение свойства (7.134).

Теорема 7.17. *Если существует равный нулю предел*

$$\lim_{n \rightarrow \infty} \frac{\mathbf{D}\{\log p_n(\Xi_n)\}}{(\mathbf{H}\{X\})^2} = 0, \quad (7.144)$$

то случайная символьная последовательность Ξ_n является энтропийно устойчивой.

Доказательство. Воспользуемся неравенством Чебышева (относительно дисперсий) для произвольной случайной величины $\xi \in \mathbb{R}^1$ и произвольного $\varepsilon > 0$:

$$\mathbf{P}\{|\xi - \mathbf{E}\{\xi\}| \geq \varepsilon\} \leq \frac{\mathbf{D}\{\xi\}}{\varepsilon^2}.$$

Полагая $\xi = -\log p_n(\Xi_n)/\mathbf{H}\{X\}$, используя (7.144) и учитывая $\mathbf{E}\{\xi\} = 1$, при $n \rightarrow \infty$, получаем

$$\mathbf{P}\left\{\left|\frac{-\log p_n(\Xi_n)}{\mathbf{H}\{X\}} - 1\right| \geq \varepsilon\right\} \rightarrow 0,$$

что по определению сходимости по вероятности и означает (7.134).

Замечание 7.7. Фигурирующая в (7.144) величина

$$\delta_n^2 = \frac{\mathbf{D} \{ \log p_n(\Xi_n) \}}{(\mathbf{H} \{ X \})^2} = \frac{\mathbf{D} \{ -\log p_n(\Xi_n) \}}{(\mathbf{E} - \log p_n(\Xi_n))^2} \geq 0$$

есть квадрат коэффициента вариации случайной величины $-\log p_n(\Xi_n)$.

Таким образом, условие (7.144) означает стремление к нулю коэффициента вариации: $\delta_n \rightarrow 0$.

Теорема 7.18. Если энтропия случайной символьной последовательности при $n \rightarrow \infty$ бесконечно возрастает ($\mathbf{H} \{ X \} \rightarrow \infty$) и существует ограниченный верхний предел

$$\lim_{n \rightarrow \infty} \frac{\mathbf{D} \{ \log p_n(\Xi_n) \}}{\mathbf{H} \{ X \}} \leq C < +\infty, \quad (7.145)$$

то такая символьная последовательность Ξ_n энтропийно устойчива.

Доказательство. Из (7.145) следует, что $\forall \varepsilon > 0$ найдется такой номер $\bar{n} = \bar{n}(\varepsilon)$, что $\forall n \geq \bar{n}$ справедлива оценка сверху:

$$\frac{\mathbf{D} \{ \log p_n(\Xi_n) \}}{(\mathbf{H} \{ X \})^2} \leq \frac{C + \varepsilon}{\mathbf{H} \{ X \}}.$$

Поскольку по условию $\mathbf{H} \{ X \} \rightarrow \infty$, то при $n \rightarrow \infty$ правая часть этого неравенства стремится к нулю, поэтому выполняется (7.144). Применяя теорему 7.16, получаем доказываемое. \square

Теорема 7.19. Если случайная символьная последовательность Ξ_n такая, что для нее существует положительная удельная энтропия

$$h = \lim_{n \rightarrow \infty} \frac{\mathbf{H} \{ X \}}{n} > 0 \quad (7.146)$$

и существует так называемая удельная дисперсия

$$d := \lim_{n \rightarrow \infty} \frac{\mathbf{D} \{ \log p_n(\Xi_n) \}}{n} \geq 0, \quad d < \infty, \quad (7.147)$$

то Ξ_n энтропийно устойчива.

Доказательство. Из (7.146) и (7.147) следуют асимптотические соотношения:

$$\mathbf{H} \{ X \} = hn + o(n), \quad \mathbf{D} \{ \log p_n(\Xi_n) \} = dn + o(n).$$

Поэтому при $n \rightarrow \infty$ существует конечный предел:

$$\frac{\mathbf{D} \{ \log p_n(\Xi_n) \}}{\mathbf{H} \{ X \}} = \frac{d + o(1)}{h + o(1)} \rightarrow \frac{d}{h} \geq 0.$$

В силу (7.145) и теоремы 7.18 получаем доказываемое. \square

Следствие 7.12. Если последовательность Ξ_n состоит из n независимых в совокупности одинаково распределенных невырожденных случайных символов (т. е. порождается стационарным ИДС без памяти), то она является энтропийно устойчивой.

Доказательство. Проверим третье достаточное условие энтропийной устойчивости, выражаемое теоремой 7.19. По условию следствия имеем

$$\log p_n(a) = \sum_{i=1}^n \log p_1(a_i), \text{ поэтому}$$

$$H\{X\} = E - \log p_n(\Xi_n) = nH\{\xi_1\}, 0 < H\{\xi_1\} < \infty;$$

$$D\{\log p_n(\Xi_n)\} = nD\{\log p_1(\xi_1)\}, 0 < D\{\log p_1(\xi_1)\} < \infty.$$

В результате согласно (7.146) и (7.147)

$$h = H\{\xi_1\} > 0, d = D\{\log p_1(\xi_1)\} < \infty.$$

□

Замечание 7.8. Понятие энтропийной устойчивости можно ввести и для отдельной случайной величины ξ , если $\forall \varepsilon > 0$

$$P\left\{\left|\frac{-\log p_1(\xi)}{H\{\xi\}} - 1\right| < \varepsilon\right\} > 1 - \delta.$$

7.10. КОЛИЧЕСТВО ИНФОРМАЦИИ ПО ШЕННОНУ И ЕГО СВОЙСТВА

Рассмотрим простейшую схему передачи дискретной информации (рис. 7.1). Пусть источник дискретных сообщений порождает случайную символьную последовательность – входной сигнал:

$$X = (x_1, \dots, x_n) \in \mathcal{A}^n,$$

где \mathcal{A} – конечный «входной алфавит» символов, $|\mathcal{A}| = N < \infty$; n – количество входных символов.

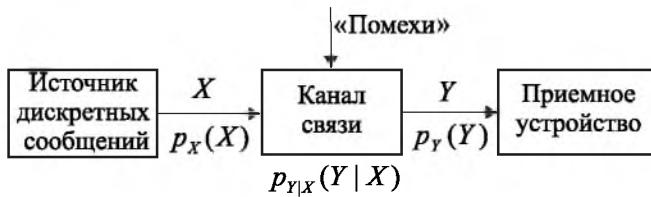


Рис. 7.1. Простейшая схема передачи информации

Канал связи, находящийся под воздействием «помех», преобразует входной сигнал X в выходную последовательность (выходной сигнал)

$$Y = (y_1, \dots, y_{n'}) \in \mathcal{B}^{n'},$$

где \mathcal{B} – конечный «выходной алфавит» символов, $|\mathcal{B}| = N' < \infty$; n' – количество выходных символов. Это преобразование может быть как детерминированным, так и стохастическим.

Обозначим: $p_X(X)$ – дискретное n -мерное распределение вероятностей входного сигнала; $p_Y(Y)$ – дискретное n' -мерное распределение вероятностей выходного сигнала; $p_{Y|X}(Y | X)$ – условное распределение вероятностей выходного сигнала Y при условии, что входной сигнал был X ;

$$p_{X|Y}(X | Y) = \frac{p_{Y|X}(Y | X)p_X(X)}{p_Y(Y)} \quad (7.148)$$

есть условное распределение вероятностей входного сигнала X при условии, что выходной сигнал оказался Y .

Количеством информации по Шеннону, содержащейся в случайной символьной последовательности $Y \in \mathcal{B}^{n'}$ относительно входного сообщения $X \in \mathcal{A}^n$, называется разность безусловной и условной энтропий

$$I\{X, Y\} = H\{X\} - H\{X | Y\}, \quad (7.149)$$

где

$$H\{X\} = - \sum_{X \in \mathcal{A}^n} p_X(X) \log p_X(X) \quad (7.150)$$

есть безусловная энтропия входного сообщения;

$$H\{X | Y\} = - \sum_{\substack{X \in \mathcal{A}^n, \\ Y \in \mathcal{B}^{n'}}} p_{X,Y}(X, Y) \log p_{X|Y}(X | Y) \quad (7.151)$$

есть условная энтропия входного сообщения X относительно выходного сообщения Y .

Данное определение количества информации как приращения энтропии введено К. Шенноном в 1948 г. Исследуем свойства шенноновского количества информации.

Свойство 7.9. Справедливы следующие эквивалентные выражения количества информации через энтропию:

$$I\{X, Y\} = H\{X\} + H\{Y\} - H\{X, Y\} = H\{Y\} - H\{Y | X\}. \quad (7.152)$$

Доказательство. Воспользуемся установленным ранее свойством иерархической аддитивности энтропии составной последовательности $X||Y = (x_1, \dots, x_n y_1, \dots, y_{n'}) \in \mathcal{A}^n \times \mathcal{B}^{n'}$:

$$\mathbf{H}\{X, Y\} = \mathbf{H}\{Y\} + \mathbf{H}\{X | Y\} = \mathbf{H}\{X\} + \mathbf{H}\{Y | X\}. \quad (7.153)$$

Из (7.153) имеем $\mathbf{H}\{X | Y\} = \mathbf{H}\{X, Y\} - \mathbf{H}\{Y\}$. Подставляя это в (7.149), получаем первое равенство в (7.152). Второе равенство в (7.152) получается подстановкой в его первую часть второго представления для $\mathbf{H}\{X, Y\}$ из (7.153). \square

Свойство 7.10. Функционал шенноновского количества информации обладает свойством симметричности: $I\{X, Y\} = I\{Y, X\}$.

Доказательство. Согласно (7.152) имеем симметричное выражение:

$$I\{X, Y\} = \mathbf{H}\{X\} + \mathbf{H}\{Y\} - \mathbf{H}\{X, Y\} = I\{Y, X\}. \quad \square$$

Замечание 7.9. Это свойство показывает, что Y содержит такое же количество информации об X , что и X об Y .

Свойство 7.11. Справедлива следующая формула для вычисления шенноновского количества информации:

$$I\{X, Y\} = \sum_{\substack{X \in \mathcal{A}^n \\ Y \in \mathcal{B}^{n'}}} p_{X,Y}(X, Y) \log \frac{p_{X,Y}(X, Y)}{p_X(X)p_Y(Y)}. \quad (7.154)$$

Доказательство. Вычислим правую часть (7.154), используя условие нормировки, свойство согласованности вероятностных распределений, а также (7.150) и (7.151):

$$\begin{aligned} B &= -\mathbf{H}\{X, Y\} - \sum_{X \in \mathcal{A}^n} \left(\sum_{Y \in \mathcal{B}^{n'}} p_{X,Y}(X, Y) \log p_X(X) \right) - \\ &\quad - \sum_{Y \in \mathcal{B}^{n'}} \left(\sum_{X \in \mathcal{A}^n} p_{X,Y}(X, Y) \log p_Y(Y) \right) = \\ &= -\mathbf{H}\{X, Y\} + \mathbf{H}\{X\} + \mathbf{H}\{Y\}, \end{aligned}$$

что согласно (7.152) совпадает с левой частью (7.154). \square

Свойство 7.12. Количество информации, содержащейся в сообщении X о нем самом, равно энтропии сообщения X :

$$I\{X, X\} = \mathbf{H}\{X\} = - \sum_{X \in \mathcal{A}^n} p_X(X) \log p_X(X). \quad (7.155)$$

Доказательство. Очевидно, что условное распределение X относительно X является вырожденным: $\mathbf{P}\{X = x | X = y\} = \delta_{xy}$. Поэтому $\mathbf{H}\{X | Y\} \equiv 0$ и из (7.149) следует (7.155). \square

Замечание 7.10. Соотношение (7.155) означает, что энтропия, свойства которой исследованы в гл. 4, может рассматриваться как частный случай функционала количества информации. Энтропию сообщения X поэтому иногда называют *собственной информацией* об X .

Свойство 7.13. Количество информации удовлетворяет неравенству

$$0 \leq I\{X, Y\} \leq \min\{\mathbf{H}\{X\}, \mathbf{H}\{Y\}\}. \quad (7.156)$$

Доказательство. По свойствам условной энтропии

$$0 \leq \mathbf{H}\{X | Y\} \leq \mathbf{H}\{X\},$$

поэтому из (7.149) и свойства 7.10 следует неравенство (7.156). \square

Свойство 7.14. Количество информации по Шеннону $I\{X, Y\}$ обращается в 0 тогда и только тогда, когда сообщения X, Y статистически независимы.

Доказательство. По свойству условной энтропии $\mathbf{H}\{X | Y\} = \mathbf{H}\{X\}$ тогда и только тогда, когда сообщения X, Y статистически независимы. Тогда из (7.149) получаем доказываемый результат.

Другой способ доказательства основан на использовании представления (7.155) и леммы 7.1. \square

Свойство 7.15. При функциональных преобразованиях сообщений $\tilde{X} = \varphi(X)$ или $\tilde{Y} = \psi(Y)$ количество информации не может возрасти:

$$I\{X, Y\} \geq I\{\varphi(X), Y\}; \quad (7.157)$$

$$I\{X, Y\} \geq I\{X, \psi(Y)\}, \quad (7.158)$$

причем равенства в (7.157) и (7.158) имеют место тогда и только тогда, когда $\varphi(\cdot), \psi(\cdot)$ – биекции.

Доказательство. По определению (7.149) имеем

$$I\{\tilde{X}, Y\} = \mathbf{H}\{Y\} - \mathbf{H}\{Y | \tilde{X}\}.$$

По свойству условной энтропии $\mathbf{H}\{Y | \varphi(X)\} \geq \mathbf{H}\{Y | X\}$, которое обращается в равенство лишь в случае, когда $\varphi(\cdot)$ – взаимно однозначное функциональное преобразование. Поэтому справедливо неравенство (7.157). Неравенство (7.158) доказывается аналогично. \square

Свойство 7.16. Если сообщения Y_1, Y_2 – независимы, то выполняется свойство аддитивности количества информации:

$$I\{X, (Y_1, Y_2)\} = I\{X, Y_1\} + I\{X, Y_2\}. \quad (7.159)$$

Доказательство. Воспользуемся формулами (7.152), (7.149) и свойством аддитивности энтропии:

$$\begin{aligned} I\{X, (Y_1, Y_2)\} &= H\{X, (Y_1, Y_2)\} - H\{X\} - H\{Y_1, Y_2\} = \\ &= H\{X, Y_1\} + H\{X, Y_2\} - H\{X\} - H\{Y_1\} - H\{Y_2\} = \\ &= I\{X, Y_1\} + I\{X, Y_2\}, \end{aligned}$$

что совпадает с (7.159). \square

Свойство 7.17. Справедливо неравенство

$$I\{X, (Y_1, Y_2)\} \geq \max\{I\{X, Y_1\}, I\{X, Y_2\}\}.$$

Доказательство. В силу (7.152) и свойств энтропии

$$I\{X, (Y_1, Y_2)\} = H\{X\} - H\{X | Y_1, Y_2\} \geq I\{X, Y_i\}, \quad i = 1, 2.$$

Пример 7.8. Пусть двоичное сообщение описывается однородной цепью Маркова $\xi_1, \xi_2, \dots \in \{0, 1\}$ с дискретным временем, с начальным распределением $\pi = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}$ и матрицей вероятностей одношаговых переходов

$$P = \begin{pmatrix} 1 - \alpha & \alpha \\ \alpha & 1 - \alpha \end{pmatrix}, \quad 0 \leq \alpha \leq 1.$$

Пусть для некоторого момента времени $t = 1, 2, \dots$ определены два соседних символа: $X := \xi_{t+1}$ (будущий символ = «пропущенный» символ), $Y := \xi_t$ (соседний наблюдаемый символ). Оценить количество информации о ξ_{t+1} , содержащееся в ξ_t .

Решение. По свойствам ОЦМ с дискретным временем имеем

$$\pi^* = \pi = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}, \quad p_X(X) = p_Y(Y) \equiv \frac{1}{2}, \quad X, Y \in \{0, 1\},$$

$$p_{X,Y}(X, Y) = \frac{1}{2} \begin{cases} 1 - \alpha, & \text{если } Y = X, \\ \alpha, & \text{если } Y \neq X. \end{cases}$$

Поэтому

$$\begin{aligned} I\{\xi_{t+1}, \xi_t\} &= \sum_{X, Y=0}^1 \frac{1}{2} \left((1 - \alpha) \delta_{Y,X} + \alpha (1 - \delta_{Y,X}) \right) \log \left(2((1 - \alpha) \delta_{Y,X} + \right. \\ &\quad \left. + \alpha (1 - \delta_{Y,X})) \right) = 1 + \alpha \log \alpha + (1 - \alpha) \log(1 - \alpha) = 1 + h(\alpha), \end{aligned}$$

где $h(\alpha) = -\alpha \log \alpha - (1 - \alpha) \log(1 - \alpha)$ – функция, исследованная в примере из п. 7.5.

Рассмотрим систему передачи информации, в которой действует источник непрерывных сообщений $X = (x_1, \dots, x_n) \in \mathbb{R}^n$, а принимаемое сообщение $Y = (y_1, \dots, y_{n'}) \in \mathbb{R}^{n'}$ также непрерывно.

Обозначим: $p_X(X)$ – плотность распределения вероятностей входного сообщения X ; $p_Y(Y)$ – плотность распределения вероятностей выходного сообщения Y ; $p_{Y|X}(Y | X)$ – условная плотность распределения выходного сообщения Y (при входном сообщении X);

$$p_{X|Y}(X | Y) = \frac{p_X(X)p_{Y|X}(Y | X)}{\int\limits_{\mathbb{R}^n} p_X(X')p_{Y|X}(Y | X')dX'} \quad (7.160)$$

есть условная плотность распределения входного сообщения X (при выходном сообщении Y); $p_{X,Y}(X, Y)$ – совместная плотность распределения;

$$\mathbf{H}_d\{X\} = - \int\limits_{\mathbb{R}^n} p_X(X) \log p_X(X)dX \quad (7.161)$$

есть безусловная дифференциальная энтропия входного случайного сообщения;

$$\mathbf{H}_d\{X | Y\} = - \int\limits_{\mathbb{R}^{n'}} \int\limits_{\mathbb{R}^n} p_{X,Y}(X, Y) \log p_{X|Y}(X | Y)dXdY \quad (7.162)$$

есть условная дифференциальная энтропия входного сигнала X относительно выходного сообщения Y .

Понятие количества информации при этом вводится аналогично дискретному случаю, рассмотренному выше.

Количеством информации по Шеннону, содержащейся в случайному выходном сигнале $Y \in \mathbb{R}^{n'}$, относительно случайного входного сообщения $X \in \mathbb{R}^n$ называется разность безусловной и условной дифференциальных энтропий (7.161) и (7.162):

$$I\{X, Y\} = \mathbf{H}_d\{X\} - \mathbf{H}_d\{X | Y\}. \quad (7.163)$$

Свойства шенноновского количества информации (7.163) для дискретного случая сохраняют силу и для источника непрерывных сообщений. В частности, аналогично свойству 7.11, удобна следующая формула для вычислений шенноновского количества информации:

$$I\{X, Y\} = \int\limits_{\mathbb{R}^{n'}} \int\limits_{\mathbb{R}^n} p_{X,Y}(X, Y) \log \frac{p_{X,Y}(X, Y)}{p_X(X)p_Y(Y)} dXdY \geq 0. \quad (7.164)$$

Пример 7.9. Пусть $X \in \mathbb{R}$, $Y \in \mathbb{R}$ – одномерные непрерывные сообщения (входной и выходной сигналы), имеющие совместное двумерное нормальное (гауссовское) распределение вероятностей:

$$\mathcal{L}\{X, Y\} = \mathcal{N}_2(\mu, \Sigma),$$

где $\mu = \begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix} \in \mathbb{R}^2$ – вектор-столбец математических ожиданий, а

$$\Sigma = \begin{pmatrix} \sigma_{XX} & \sigma_{XY} \\ \sigma_{XY} & \sigma_{YY} \end{pmatrix}$$

есть ковариационная матрица входного и выходного сигналов.

Проводя вычисления согласно (7.164), получаем

$$I\{X, Y\} = -\log \sqrt{1 - \rho_{X,Y}^2}, \quad (7.165)$$

где $\rho_{X,Y} = \sigma_{XY} / \sqrt{\sigma_{XX} \sigma_{YY}}$ – коэффициент корреляции входного и выходного сообщений.

Рассмотрим ситуацию, когда одно из сообщений X, Y непрерывно, а другое – дискретно. Пусть, например, имеется источник непрерывных сообщений $X \in \mathbb{R}^n$ с плотностью распределения $p_X(X)$, а канал связи является цифровым, поэтому $Y \in \mathcal{B}$ – дискретное выходное сообщение с дискретным распределением вероятностей $p_Y(Y)$, где \mathcal{B} – дискретное множество. Обозначим через $p_{X|Y}(X | Y)$ условную плотность распределения случайного входного сигнала X при условии, что выходное сообщение – Y . Аналогично (7.164) получаем формулу для вычисления шенноновского количества информации:

$$I\{X, Y\} = \sum_{Y \in \mathcal{B}} p_Y(Y) \int_{\mathbb{R}^n} p_{X|Y}(X | Y) \log \frac{p_{X|Y}(X | Y)}{p_X(X)} dX. \quad (7.166)$$

Свойства функционала (7.166) также совпадают со свойствами, доказанными ранее.

7.11. ШЕННОНОВСКИЕ МОДЕЛИ КРИПТОСИСТЕМ

В пп. 7.11, 7.12 рассмотрим применение шенноновской теории информации к построению математических моделей криптосистем и оценке стойкости простейших симметричных криптосистем.

В настоящее время криптосистемы принято разделять на два класса: симметричные (одноключевые) и асимметричные (двухключевые). Общая схема симметричной криптосистемы приведена на рис. 7.2.



Рис. 7.2. Общая схема криптосистемы

Здесь приняты следующие обозначения: $X = (x_1, \dots, x_n)$ – исходное сообщение (plaintext, message), генерируемое источником сообщений и представляющее собой последовательность n символов алфавита мощностью $v \geq 2$:

$$x_i \in A_v = \{0, 1, \dots, v-1\}, \quad i = \overline{1, n}, \quad X \in A_v^n;$$

$$\theta^0 = (\theta_1^0, \theta_2^0, \dots, \theta_L^0) \in A_\mu^L$$

есть истинное значение ключа (key), L символов которого принимают значения из алфавита A_μ , $\mu \geq 2$; $Y = (y_1, y_2, \dots, y_n) \in A_v^n$ – шифртекст (криптоматта, выходная последовательность, зашифрованное сообщение, ciphertext), получающаяся применением к X криптографического дискретного функционального преобразования (операции шифрования, cipher, encryption):

$$Y = f(X; \theta^0) : V_v^n \times V_\mu^L \rightarrow V_v^n \quad (7.167)$$

или

$$y_i = f_i(x_1, \dots, x_n; \theta^0), \quad i = \overline{1, n}.$$

Функция $f(\cdot)$ в (7.167) такова, что при любом фиксированном значении ключа θ^0 она является взаимооднозначным функциональным преобразованием (биекцией), и обратное преобразование (расшифрование, дешифрование, decryption) единственно и восстанавливает исходный текст:

$$X = f^{-1}(Y; \theta^0). \quad (7.168)$$

Особенностью симметричных (одноключевых) систем является симметричное использование одного и того же ключа θ^0 отправителем и получателем (этот ключ секретный и поставляется абонентам специальным конфиденциальным способом). Отсюда и название данного класса криптосистем – «одноключевые».

Приведем математические модели элементарных криптосистем, которые потребуются при оценке их стойкости, а также при решении задач криптоанализа и построения стандартных криптосистем.

1. Подстановка символов алфавита.

Пусть определена подстановка на множестве $\{1, 2, \dots, v\}$:

$$\begin{pmatrix} 1 & 2 & \dots & v \\ s_1 & s_2 & \dots & s_v \end{pmatrix} \quad (7.169)$$

и определен v -вектор ($L = v = \mu$):

$$\theta^0 = (\theta_1^0, \dots, \theta_v^0), \quad \theta_i^0 = s_i - 1, \quad i = \overline{1, v},$$

задающий перестановку символов алфавита A_v .

Тогда шифр простой подстановки – это криптографическое преобразование вида (7.167), осуществляемое поэлементно:

$$y_t = f_t(x_t; \theta^0) := \theta_{x_t+1}^0, \quad t = 1, 2, \dots. \quad (7.170)$$

Обратное преобразование (7.168) при этом будет иметь вид

$$x_t = \bar{\theta}_{y_t+1}^0 =: f_t^{-1}(y_t; \theta^0), \quad (7.171)$$

где $\bar{\theta}_i^0 = \bar{s}_i - 1$;

$$\begin{pmatrix} 1 & 2 & \dots & v \\ \bar{s}_1 & \bar{s}_2 & \dots & \bar{s}_v \end{pmatrix}$$

есть подстановка, обратная (7.169).

2. Перестановка символов с периодом T .

Пусть $T \in N$ – некоторый заданный период и определена некоторая подстановка на множестве $\{1, 2, \dots, T\}$:

$$\begin{pmatrix} 1 & 2 & \dots & T \\ s_1 & s_2 & \dots & s_T \end{pmatrix}.$$

Как и в предыдущем преобразовании, с помощью этой подстановки определяется ключ $\theta^0 = (\theta_1^0, \dots, \theta_T^0)$, $\theta_i^0 = s_i$, $i = \overline{1, T}$.

Криптопреобразование (7.167) осуществляется следующим образом. Исходное сообщение X разбивается на «блоки символов» длиной T , и внутри каждого блока производится перестановка символов в соответствии с заданным ключом θ^0 . Для произвольного номера символа $t = (i-1)T + \tau$, где $i \in \{1, 2, \dots\}$, $\tau \in \{1, 2, \dots, T\}$, такое преобразование записывается в виде

$$y_t = x_{(i-1)T+\theta_\tau^0}. \quad (7.172)$$

Например, если $T = 5$, а $\theta^0 = (2, 3, 1, 5, 4)$, то сообщение

$$X = (x_1, x_2, x_3, x_4, x_5 | x_6, x_7, x_8, x_9, x_{10} | \dots)$$

переходит в шифртекст

$$Y = (x_2, x_3, x_1, x_5, x_4 | x_7, x_8, x_6, x_{10}, x_9 | \dots).$$

Легко убедиться, что обратное преобразование (по отношению к (7.172)) имеет вид

$$x_t = y_{(i-1)T+\bar{\theta}_\tau^0}, \quad (7.173)$$

где

$$\begin{pmatrix} 1 & 2 & \dots & T \\ \bar{\theta}_1^0 & \bar{\theta}_2^0 & \dots & \bar{\theta}_T^0 \end{pmatrix}$$

есть подстановка, обратная θ^0 .

Для «усиления» стойкости криптопреобразования (7.172) к криптоанализу используют композицию нескольких перестановок с различными периодами. Если сделано $L \geq 2$ перестановок типа (7.172) с периодами T_1, \dots, T_L , то составная перестановка, очевидно, будет иметь период

$$T = \text{НОК}(T_1, T_2, \dots, T_L).$$

Следовательно, если периоды $\{T_1, \dots, T_L\}$ – взаимно прямые числа, то достигается максимальный период $T_{\max} = T_1 \cdot \dots \cdot T_L$.

3. Шифр Виженера и его модификации.

Как и в предыдущей криптосистеме, исходный текст X разбивается на блоки длиной T . Ключ θ^0 представляет собой фиксированный набор символов исходного алфавита A_ν ($\mu = \nu$):

$$\theta^0 = (\theta_1^0, \dots, \theta_T^0), \quad \theta_i^0 \in V_\nu, \quad i = \overline{1, T}.$$

Криптотекущая для произвольного номера $t = (i-1)T + \tau$, $i \in \{1, 2, \dots\}$, $\tau \in \{1, 2, \dots, T\}$, задается с помощью вычетов по модулю ν (см. п. 2.7):

$$y_t = (x_t + \theta_\tau^0) \bmod \nu. \quad (7.174)$$

Это криптопреобразование иногда называется преобразованием циклического сдвига с периодом T . Обратное преобразование по отношению к (7.174):

$$x_t = (y_t + \nu - \theta_\tau^0) \bmod \nu. \quad (7.175)$$

Пример шифртекста, построенного с помощью шифра Виженера с периодом $T = 6$ и ключом $\theta^0 = (4, 3, 2, 5, 1, 3)$, можно найти в романе Ж. Верна «Жангада» [10, с. 228].

Приведем ряд частных случаев криптопреобразования Виженера, известных с древних времен.

Шифр Цезаря – это частный случай преобразования Виженера с периодом $T = 1$ и ключом $\theta^0 \in A_\nu$:

$$y_t = (x_t + \theta^0) \bmod \nu, \quad t = 1, 2, \dots. \quad (7.176)$$

При этом согласно (7.176) каждый символ (буква) исходного текста заменяется символом, циклически сдвинутым на фиксированное количество мест θ^0 по алфавиту A_ν .

В качестве примера приведем шифртекст длиной $n = 41$:

$$Y = PELCGBYBTLVFPELCGBTENCULNAQPELCGBNANYLFVF,$$

полученного с помощью криптопреобразования Цезаря (7.176) при $\nu = 26$, $\theta^0 = 13$ из исходного отрывка английского текста:

$$X = CRYPTOLOGYISCRYPTOGRAPHYANDCRYPTOANALYSIS.$$

Иногда рассматривают обратный шифр Цезаря:

$$y_t = (\theta^0 + \nu - x_t) \bmod \nu. \quad (7.177)$$

Шифр Бефора – это модификация T -периодического шифра Виженера (7.174):

$$y_t = (\theta_\tau^0 + \nu - x_t) \bmod \nu, \quad t = 1, 2, \dots. \quad (7.178)$$

Повторное применение $L \geq 2$ шифров Виженера называется составным шифром Виженера. Пусть есть L шифров Виженера, которые имеют периоды T_1, \dots, T_L и ключи $\theta_1^0 = (\theta_{11}^0, \dots, \theta_{1T_1}^0), \dots, \theta_L^0 = (\theta_{L1}^0, \dots, \theta_{LT_L}^0)$. Если через $\{\theta_{it}^0 : t = 1, 2, \dots\}$ обозначить ключ θ_τ^0 , многократно периодически повторяемый с периодом T_i , то L -составной шифр Виженера имеет вид

$$y_t = (x_t + \theta_{1t}^0 + \dots + \theta_{Lt}^0) \bmod \nu, \quad t = 1, 2, \dots. \quad (7.179)$$

Легко показать, что L -составной шифр Виженера можно рассматривать как обычный шифр Виженера с периодом $T = \text{НОК}\{T_1, \dots, T_L\}$.

4. Криптопреобразование Вернама (поточчный шифр).

Криптопреобразование Вернама – специальный частный случай криптопреобразования Виженера (7.174), когда длина используемого ключа равна длине передаваемого сообщения n :

$$y_t = (x_t + \theta_t^0) \bmod \nu, \quad t = \overline{1, n}. \quad (7.180)$$

Обратное криптопреобразование имеет вид

$$x_t = (y_t + \nu - \theta_t^0) \bmod \nu, \quad t = \overline{1, n}.$$

В качестве ключа $\theta^0 = (\theta_1^0, \dots, \theta_n^0) \in A_\nu^n$ используется реализация последовательности n независимых, одинаково распределенных на A_ν случайных величин либо некоторая реализация текста. Ключ θ^0 такого типа (используемого всего один раз) в литературе называется бегущей строкой, одноразовым блокнотом (one-time pad) или гаммой. Иногда в качестве $\{\theta_t^0\}$ используется псевдослучайная последовательность, порождаемая специальным программным датчиком (см. гл. 6).

5. Биграммная (N -граммная) подстановка.

Это криптопреобразование использует тот же принцип, что и простая подстановка (7.170). Однако в отличие от (7.170) в данном случае вместо подстановки одного символа: $A_v \leftrightarrow A_v$ осуществляется подстановка пар символов (биграмм): $A_v^2 \leftrightarrow A_v^2$ либо набора N соседних символов (N -грамм): $A_v^N \leftrightarrow A_v^N$. Ключ θ^0 в биграммной подстановке представляет собой $(v \times v)$ -матрицу, (i, j) -й элемент которой – биграмма (i', j') , заменяющая биграмму (i, j) .

В заключение отметим, что если имеются две произвольные криптосистемы T и R , то их часто можно комбинировать для получения новой криптосистемы:

$$S = f(T, R).$$

Наиболее часто используются следующие два типа оператора комбинирования $f(\cdot)$.

1. Произведение криптосистем:

$$S = f_1(T, R) = TR, \quad S = f_2(T, R) = RT,$$

причем, вообще говоря, $TR \neq RT$. В существующих стандартных криптосистемах произведение шифров используется весьма часто. Например, после подстановки применяют транспозицию или после транспозиции – код Виженера.

2. Взвешенная сумма криптосистем:

$$S = pT + (1 - p)R, \quad p \in [0, 1].$$

Выбор преобразования T осуществляется с вероятностью p , а преобразования R – с вероятностью $1 - p$. Эта функция комбинирования поясняется схемой, приведенной на рис. 7.3.

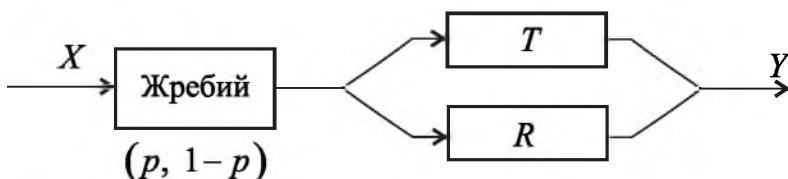


Рис. 7.3. Взвешенная сумма криптосистем

7.12. ТЕОРЕТИКО-ИНФОРМАЦИОННЫЕ ОЦЕНКИ СТОЙКОСТИ СИММЕТРИЧНЫХ КРИПТОСИСТЕМ

В этом пункте исследуем общие вопросы устойчивости симметричных криптосистем к криптоанализу, используя теоретико-информационный подход Шеннона. Свойство устойчивости криптосистемы к криптоанализу принято называть криптостойкостью.

Прежде всего рассмотрим вопросы, связанные с потенциальной (т. е. максимально возможной) криптостойкостью. Насколько устойчива криптосистема, если криптоаналитик не ограничен временем и вычислительными средствами для анализа шифртекстов? Имеет ли шифртекст Y единственное решение (т. е. однозначно ли определяет ключ θ^0)? Какой должна быть минимальная длина шифртекста n_{\min} , чтобы решение стало единственным? Существуют ли криптосистемы, в которых нельзя найти единственное решение независимо от длины n исследуемого шифртекста? Существуют ли криптосистемы, в которых криптоаналитик не получает никакой информации, сколько бы он ни перехватил шифртекстов? Для решения этих проблем К. Шенон применил аппарат математической теории информации. Будем рассматривать симметричные криптосистемы, описанные в п. 7.11.

Основное модельное предположение К. Шеннона об исходном сообщении X состоит в том, что язык источника сообщений может рассматриваться как некоторый вероятностный процесс, порождающий дискретную последовательность символов в соответствии с некоторой системой вероятностей (модели сообщений рассмотрены в гл. 4). Поэтому исходное сообщение X К. Шенон предполагал случайным n -вектором с дискретным распределением вероятностей:

$$\begin{aligned} \mathbf{P}\{X = X^{(i)}\} &= q_i, \quad i = \overline{1, \sqrt{n}}, \\ q_1 + q_2 + \dots + q_{\sqrt{n}} &= 1, \end{aligned} \tag{7.181}$$

где $X^{(i)} \in A_{\nu}^n$ – i -й возможный вариант исходного n -символьного сообщения из алфавита A_{ν} . Ключ $\theta^0 = (\theta_1^0, \dots, \theta_m^0)$ также предполагается (генерируется) случайным вектором, не зависящим от X , с дискретным распределением вероятностей:

$$\mathbf{P}\{\theta^0 = \theta^{(j)}\} = p_j, \quad j = \overline{1, \mu^m}, \quad \sum_{j=1}^{\mu^m} p_j = 1, \tag{7.182}$$

где $\theta^{(j)} \in A_{\mu}^m$ – j -й возможный вариант ключевой m -символьной последовательности в алфавите A_{μ} ; p_j – априорная вероятность ключа $\theta^{(j)}$.

Симметричная криптосистема называется *совершенно криптостойкой*, если апостериорное распределение вероятностей исходного случайного сообщения X при регистрации случайного шифртекста $Y = f(X; \theta^0)$ совпадает с априорным распределением вероятностей:

$$\mathbf{P}\{X = X^{(i)} | Y = Y^{(l)}\} = \mathbf{P}\{X = X^{(i)}\} = q_i, \quad i, l = \overline{1, \sqrt{n}}. \tag{7.183}$$

Смысл условия (7.183) в том, что хотя криптоаналитик и имеет шифртекст, он не добавляет ему информации о переданном сообщении.

Теорема 7.20. *Необходимое и достаточное условие совершенной криптостойкости состоит в том, что условное распределение вероятностей шифртекста $Y \in A_v^n$ при фиксированном сообщении $X \in A_v^n$ не зависит от X :*

$$\mathbf{P}\{Y = Y^{(l)} | X = X^{(i)}\} = \mathbf{P}\{Y = Y^{(l)}\}, \quad i, l = \overline{1, \sqrt{n}}. \quad (7.184)$$

Доказательство. По формуле Байеса имеем

$$\mathbf{P}\{X = X^{(i)} | Y = Y^{(l)}\} = \frac{q_i \mathbf{P}\{Y = Y^{(l)} | X = X^{(i)}\}}{\mathbf{P}\{Y = Y^{(l)}\}}.$$

Видно, что (7.183) выполняется тогда и только тогда, когда

$$\frac{\mathbf{P}\{Y = Y^{(l)} | X = X^{(i)}\}}{\mathbf{P}\{Y = Y^{(l)}\}} = 1,$$

что совпадает с (7.184). \square

Следствие 7.13. *Если выполняется условие совершенной криптостойкости, то количество информации по Шеннону, содержащейся в шифртексте Y об исходном сообщении X , равно нулю:*

$$I\{X, Y\} = I\{Y, X\} = 0. \quad (7.185)$$

Доказательство. Вычислим энтропию исходного сообщения X и условную энтропию X относительно шифртекста Y с учетом (7.183):

$$\begin{aligned} H\{X\} &= - \sum_{i=1}^{\sqrt{n}} q_i \log q_i, \\ H\{X|Y\} &= - \sum_{i=1}^{\sqrt{n}} \sum_{l=1}^{\sqrt{n}} \mathbf{P}\{X = X^{(i)}, Y = Y^{(l)}\} \times \\ &\quad \times \log \mathbf{P}\{X = X^{(i)} | Y = Y^{(l)}\} = - \sum_{i,l=1}^{\sqrt{n}} \mathbf{P}\{X = X^{(i)}, Y = Y^{(l)}\} \times \\ &\quad \times \log \mathbf{P}\{X = X^{(i)}\} = - \sum_{i=1}^{\sqrt{n}} q_i \log q_i = H\{X\}. \end{aligned}$$

Тогда по определению количества информации имеем

$$I\{X, Y\} = H\{X\} - H\{X|Y\} = 0.$$

Второе равенство в (7.185) вытекает из свойства симметричности функционала количества информации. \square

Следствие 7.14. Пусть $\{Y^{(l)} : l = \overline{1, \sqrt{n}}\} = A_{\sqrt{n}}^n$ – множество всевозможных шифртекстов, порождаемых криптофункцией $Y = f(X; \theta)$;

$$\mathcal{J}_{il} = \{j : f(X^{(i)}, \theta^{(j)}) = Y^{(l)}\} \subset N \quad (7.186)$$

есть подмножество номеров ключей, переводящих исходный текст $X^{(i)}$ в один и тот же шифртекст $Y^{(l)}$. Чтобы $f(\cdot)$ удовлетворяла свойству совершенной криптостойкости, необходимо и достаточно, чтобы выполнялось свойство

$$\sum_{j \in \mathcal{J}_{il}} p_j = a_l = \underset{1 \leq i \leq \sqrt{n}}{\text{invar}}. \quad (7.187)$$

Доказательство. Прежде всего отметим, что множество индексов \mathcal{J}_{il} не пусто в силу биективности $f(\cdot)$. Вычислим и сравним левую и правую части (7.184) с учетом (7.186) и свойства независимости θ^0 и X :

$$\begin{aligned} \mathbf{P}\{Y = Y^{(l)} | X = X^{(i)}\} &= \mathbf{P}\{f(X^{(i)}; \theta^0) = Y^{(l)} | X = X^{(i)}\} = \\ &= \sum_{j \in \mathcal{J}_{il}} \mathbf{P}\{\theta^0 = \theta^{(j)} | X = X^{(i)}\} = \sum_{j \in \mathcal{J}_{il}} \mathbf{P}\{\theta^0 = \theta^{(j)}\} = \sum_{j \in \mathcal{J}_{il}} p_j; \end{aligned} \quad (7.188)$$

$$\begin{aligned} \mathbf{P}\{Y = Y^{(l)}\} &= \mathbf{P}\{f(X; \theta^0) = Y^{(l)}\} = \\ &= \sum_{i, j} q_i p_j \delta_{f(X^{(i)}, \theta^{(j)}), Y^{(l)}} = \sum_{i=1}^{\sqrt{n}} q_i \sum_{j \in \mathcal{J}_{il}} p_j. \end{aligned} \quad (7.189)$$

Сравнивая (7.188) и (7.189), заключаем справедливость (7.187). \square

Теорема 7.21. Необходимым условием выполнения свойства совершенной криптостойкости является справедливость следующих неравенств энтропий:

$$\mathbf{H}\{\theta^0\} \geq \mathbf{H}\{X\}; \quad (7.190)$$

$$\mathbf{H}\{\theta^0\} \geq \mathbf{H}\{Y\}. \quad (7.191)$$

Доказательство. Как было установлено при доказательстве теоремы 7.20, справедливо равенство

$$\mathbf{H}\{X\} = \mathbf{H}\{X|Y\}. \quad (7.192)$$

Воспользуемся свойством иерархической аддитивности энтропии (см. гл. 4):

$$\begin{aligned} \mathbf{H}\{Y, X, \theta^0\} &= \mathbf{H}\{Y\} + \mathbf{H}\{X|Y\} + \mathbf{H}\{\theta^0|X, Y\} = \\ &= \mathbf{H}\{Y, \theta^0, X\} = \mathbf{H}\{Y\} + \mathbf{H}\{\theta^0|Y\} + \mathbf{H}\{X|\theta^0, Y\}. \end{aligned} \quad (7.193)$$

Поскольку при фиксированном шифртексте Y и ключе θ^0 исходный текст $X = f^{-1}(Y; \theta^0)$ не случаен, то $H\{X|\theta^0, Y\} = 0$. Поэтому из уравнения (7.193) находим $H\{X|Y\} = H\{\theta^0|Y\} - H\{\theta^0|X, Y\}$. По свойствам энтропии отсюда следует

$$H\{X|Y\} \leq H\{\theta^0|Y\} \leq H\{\theta^0\}.$$

Используя это в (7.192), приходим к (7.190). Неравенство (7.191) доказывается аналогично. \square

Теорему 7.21 в криптологии называют пессимистическим утверждением К. Шеннона, так как она требует, чтобы энтропия (неопределенность) ключа θ^0 была не меньше энтропии исходного текста X (или шифртекста Y). Поскольку распределение $\{q_i\}$ исходного текста X может быть произвольным, а $\max_{\{q_i\}} H\{X\} = \log v^n$, то неравенство (7.190) примет вид

$$H\{\theta^0\} \geq \log v^n.$$

Для его выполнения в случае $\mu = v$ требуется, чтобы длина ключа m была не меньше длины шифруемого текста: $m \geq n$. Для практики наиболее интересен случай самых коротких ключей: $m = n$.

Теорема 7.22. Если $\mu = v$, $m = n$ и для любых $i, l \in \{1, 2, \dots, v^n\}$ уравнение

$$f(X^{(i)}; \theta^{(j)}) = Y^{(l)} \quad (7.194)$$

имеет единственное решение $j = j_{il}$ (т. е. $\mathcal{J}_{il} = \{j_{il}\}$ – одноточечное множество), то необходимым и достаточным условием совершенной криптостойкости является равновероятность используемых ключей:

$$p_j \equiv \text{const} = \frac{1}{v^n}, \quad j = \overline{1, v^n}. \quad (7.195)$$

Доказательство. Согласно (7.187)

$$p_{j_{il}} = a_l \quad \forall i, j, l. \quad (7.196)$$

Очевидно (в силу биекции), что j_{il} зависит от i и l так, что при изменении $i \in \{1, 2, \dots, v^n\}$ индекс j_{il} пробегает v^n всевозможных значений. Поэтому (7.196) невозможно без выполнения (7.195). \square

Следствие 7.15. Криптореобразование Вернама (7.180) при условии равновероятности ключей (7.192) обладает свойством совершенной криптостойкости.

Доказательство. В силу (7.180) уравнение (7.194) имеет единственное решение для любых $X^{(i)}, Y^{(l)}$: $\theta^{(j)} = (Y^{(l)} + v - X^{(i)}) \bmod v$, где вычеты вычисляются покомпонентно. Поэтому из (7.192) и теоремы 7.22 получаем доказываемый результат. \square

Данное следствие объясняет, почему для передачи и защиты наиболее важной информации широко используются поточные криптосистемы, базирующиеся на криптопреобразовании Вернама. Следствие объясняет также, почему к качеству генерации ключевой последовательности $\theta^0 = (\theta_1^0, \theta_2^0, \dots, \theta_n^0)$ (гаммы) предъявляются столь высокие требования (см. гл. 6).

Рассмотрим еще одну важную характеристику криптосистем (связанную с криптостойкостью), введенную К. Шенноном, – расстояние единственности U . Для этого воспользуемся следующим соотношением:

$$\begin{aligned}\mathbf{H}\{\theta^0, X\} &= \mathbf{H}\{\theta^0\} + \mathbf{H}\{X\} && \text{(по условию независимости } \theta^0, X\text{);} \\ \mathbf{H}\{\theta^0|X\} &= \mathbf{H}\{Y|X\} && \text{(по свойствам криптосистемы);} \\ \mathbf{H}\{\theta^0, X\} &= \mathbf{H}\{Y, X\} && \text{(вытекает из предыдущего);} \\ \mathbf{H}\{Y|X\} &= \mathbf{H}\{Y\} + \mathbf{H}\{X|Y\} && \text{(по свойству энтропии).}\end{aligned}$$

Отсюда получаем выражение для условной энтропии исходного текста X относительно наблюдаемого шифртекста Y :

$$\mathbf{H}\{X|Y\} = \mathbf{H}\{X\} - (\mathbf{H}\{Y\} - \mathbf{H}\{\theta^0\}).$$

Поскольку условная энтропия неотрицательна, получаем неравенство

$$\mathbf{H}\{X|Y\} = \mathbf{H}\{X\} - (\mathbf{H}\{Y\} - \mathbf{H}\{\theta^0\}) \geq 0. \quad (7.197)$$

Расстояние единственности – такая минимальная длина шифртекста Y (и исходного текста), при которой исчезает неопределенность в исходном тексте X при наблюдении шифртекста Y :

$$U = \min\{n : \mathbf{H}\{X|Y\} = 0\}. \quad (7.198)$$

Следуя М. Хеллману (см. п. 7.8, 7.9), построим оценки энтропий, входящих в (7.197) и (7.198):

$$\mathbf{H}\{X\} = n \log v_X, \quad \mathbf{H}\{Y\} = n \log v_Y, \quad \mathbf{H}\{\theta^0\} = \log |\Theta|, \quad (7.199)$$

где v_X (v_Y) – число, подбираемое так, что приближенно v_X^n (v_Y^n) реализаций исходных текстов X (шифртекстов Y) имеют вероятности, значительно отличающиеся от нуля, а остальные реализации имеют пренебрежимо малую вероятность; $|\Theta|$ – мощность пространства используемых ключей; $v_X \leqslant v_Y < v$.

Подставляя (7.199) в (7.197) и (7.198), находим приближенное выражение для расстояния единственности:

$$U = \frac{\log |\Theta|}{\log(v_Y/v_X)}. \quad (7.200)$$

Для реальных криптосистем обычно оказывается, что шифртекст Y имеет распределение, близкое к равномерному, поэтому $\nu_Y \approx \nu$. Тогда формула (7.200) принимает следующий вид

$$U = \frac{\log |\Theta|}{k}, \quad k = \log \frac{\nu}{\nu_X} > 0,$$

где k – коэффициент, характеризующий избыточность языка. Например, для текстов на английском, немецком и французском языках этот коэффициент приближенно одинаков и равен $k \approx 0,53$.

7.13. ЗАДАНИЯ

1. Пусть $\xi \in \{0, 1\}$ – двоичный случайный символ с распределением вероятностей Бернулли: $\mathbf{P}\{\xi = 1\} = p$, $\mathbf{P}\{\xi = 0\} = 1 - p$. Построить график зависимости энтропии $\mathbf{H}\{\xi\}$ от элементарной вероятности $p \in [0, 1]$, исследовать эту функцию на экстремум.

2. Пусть $\xi \in \{0, 1, \dots, N - 1\}$ – случайный символ, причем элементарная вероятность $p_0 = \mathbf{P}\{\xi = 0\} = \varepsilon$ фиксирована. Доказать, что при произвольных $p_i = \mathbf{P}\{\xi = i\}$ ($i = \overline{1, N - 1}$)

$$\mathbf{H}\{\xi\} \leq -\varepsilon \log \varepsilon - (1 - \varepsilon) \log \frac{1 - \varepsilon}{N - 1},$$

причем равенство достигается тогда и только тогда, когда $p_i = (1 - \varepsilon)/(N - 1)$ ($i = \overline{1, N - 1}$).

3. Доказать, что энтропия $\mathbf{H}\{\xi\} = -\sum_{i=1}^N p_i \log p_i$ – выпуклая (вверх) функция в области D всех вероятностных векторов $p = (p_1, \dots, p_N)$, дающих дискретные распределения вероятностей.

4. Пусть $\xi \in \{0, 1, \dots, N - 1\}$ – случайный символ с биномиальным распределением вероятностей ($\mathcal{L}\{\xi\} = \text{Bi}(N - 1, p)$): $p_i = \mathbf{P}\{\xi = i\} = C_{N-1}^i p^i (1 - p)^{N-1-i}$ ($i = \overline{0, N - 1}$). Вычислить энтропию $\mathbf{H}\{\xi\}$ и исследовать на экстремум ее зависимость от элементарной вероятности p .

5. Пусть $\xi \in \{0, 1, 2, \dots\}$ – случайный символ с распределением вероятностей Пуассона ($\mathcal{L}\{\xi\} = \Pi(\lambda)$): $p_i = \mathbf{P}\{\xi = i\} = e^{-\lambda} \lambda^i / i!$. Вычислить энтропию $\mathbf{H}\{\xi\}$ и исследовать на экстремум ее зависимость от параметра $\lambda > 0$.

6. Пусть $\xi \in \{0, 1, 2, \dots\}$ – случайный символ с геометрическим распределением вероятностей ($\mathcal{L}\{\xi\} = G(p)$): $p_i = \mathbf{P}\{\xi = i\} = p(1 - p)^i$. Вычислить энтропию $\mathbf{H}\{\xi\}$ и исследовать на экстремум ее зависимость от параметра $p \in [0, 1]$.

7. Пусть $\xi \geq 0$ – случайная величина с экспоненциальным распределением ($\mathcal{L}\{\xi\} = E(\lambda)$), плотность которого равна $p_\xi(x) = \lambda e^{-\lambda x}$, $x \geq 0$. Вычислить дифференциальную энтропию $H_d\{\xi\}$ и исследовать на экстремум ее зависимость от параметра $\lambda > 0$.

8. Исследовать семейство одномерных абсолютно непрерывных распределений вероятностей, заданных на полуправой:

$$\mathcal{P}_4(\lambda) = \left\{ p_X(x), x \geq 0 : p_X(x) \geq 0, \int_0^\infty p_X(x) dx = 1, \right.$$

$$\left. \int_0^\infty x p_X(x) dx = \frac{1}{\lambda} \right\}, \lambda > 0.$$

Найти максимум дифференциальной энтропии на классе $\mathcal{P}_4(\lambda)$. Показать, что этот максимум достигается для плотности экспоненциального распределения вероятностей $E(\lambda)$:

$$p^*(x) = \lambda e^{-\lambda x}, x \geq 0.$$

9. Случайные символы ξ_1, ξ_2 в сообщении зависимы. Известно, что $H\{\xi_1\} = 8$ бит, $H\{\xi_2\} = 12$ бит. Какие значения может принимать условная энтропия $H\{\xi_2 | \xi_1\}$, если $H\{\xi_1 | \xi_2\}$ изменяется в максимально возможных пределах?

10. Пусть стационарный источник дискретных сообщений порождает случайную символьную последовательность $\Xi_n = (\xi_1, \dots, \xi_n) \in A^n$, обладающую марковским свойством s -го порядка ($s \geq 0$):

$$\begin{aligned} P\{\xi_n = a_n | \xi_{n-1} = a_{n-1}, \dots, \xi_1 = a_1\} &= \\ &= P\{\xi_n = a_n | \xi_{n-1} = a_{n-1}, \dots, \xi_{n-s} = a_{n-s}\}, \\ a_1, \dots, a_n &\in A, n > s. \end{aligned}$$

Доказать, что условная энтропия

$$H\{\xi_n | \Xi_{n-1}\} = H\{\xi_s | \xi_{s-1}, \dots, \xi_1\}, n \geq s,$$

и удельная энтропия

$$h = H\{\xi_s | \xi_{s-1}, \dots, \xi_1\}.$$

11. Пусть $f_1(x), f_2(x)$ – плотности распределения двух случайных символов сообщений ξ_1, ξ_2 , отличающиеся только параметром сдвига. Доказать, что дифференциальные энтропии символов совпадают: $H_d\{\xi_1\} = H_d\{\xi_2\}$.

12. Пусть ξ – случайный символ сообщения с плотностью распределения вероятностей $f(x)$, а $\eta = a\xi + b$, где a, b – фиксированные неслучайные величины, причем $a \neq 0$. Доказать, что $H_d\{\eta\} = H_d\{\xi\} + \log|a|$.

13. Для трех дискретных случайных символов ξ_1, ξ_2, ξ_3 энтропии одинаковы: $H\{\xi_1\} = H\{\xi_2\} = H\{\xi_3\} = h$. Вычислить количество информации $I\{\xi_1, \xi_2, \xi_3\}$, если:

- а) $H\{\xi_1, \xi_2, \xi_3\} = 3h$;
- б) $H\{\xi_1, \xi_2, \xi_3\} = h$.

14. Алфавит состоит из восьми согласных и восьми гласных букв. В сообщении все буквы алфавита равновероятны и независимы. После прохождения через канал связи согласные всегда принимаются безошибочно, а гласные – только в половине случаев; в другой половине имеют место ошибки «случайного перепутывания гласных букв». Какое количество информации по Шеннону содержит в принятом символе о переданном?

15. ИДС генерирует независимую последовательность из алфавита, состоящего из 16 равновероятных символов. При передаче по каналу связи символы искажаются так, что четверть всех символов алфавита принимается неправильно, причем все ошибки равновероятны. Вычислить шенноновское количество информации в принятом символе о переданном.

16. Студент может получить зачет с вероятностью 0,3, не проработав весь учебный материал, и с вероятностью 0,9, проработав его полностью. Какое количество информации о степени проработки учебного материала можно получить по результату сдачи зачета, если в среднем 90 % студентов полностью проработали учебный материал?

17. Имеется случайная последовательность двоичных символов $\xi_1, \dots, \xi_n \in \{0, 1\}$. Известно, что возможно появление только таких реализаций, для которых $\xi_1 + \dots + \xi_n = 0$, причем все они равновероятны. Вычислить:

- а) $I\{\xi_1, \xi_2\}$;
- б) $I\{(\xi_1, \xi_2), \xi_3\}$;
- в) $I\{(\xi_1, \dots, \xi_{n-1}), \xi_n\}$.

18. Статистика прогнозирования дождя в некотором городе характеризуется следующей таблицей вероятностей событий (ξ, η) :

		Истинное состояние ξ	
		Дождь	Нет дождя
η	Дождь	2/16	3/16
	Нет дождя	1/16	10/16

Студент ФПМИ заметил, что бюро прогнозов не ошибается лишь в 12/16 случаев. Если всегда давать прогноз «нет дождя», то доля правильных решений окажется больше: 13/16. Студент предложил этот способ прогнозирования с просьбой о гонораре. Однако начальник бюро прогнозов, как специалист по теории информации, посчитал проект студента необоснованным. Почему?

19. Доказать, что $I\{\xi, f(\eta)\} \leq I\{\xi, \eta\}$, каким бы ни было функциональное преобразование $f(\cdot)$.

20. Известно, что энтропия, приходящаяся на одну букву русскоязычного текста, составляет приблизительно 1,2 бита. Каково минимальное среднее количество десятичных символов, необходимых для передачи информации, содержащейся в телеграмме из 100 букв?

21. На входе канала связи имеется последовательность двоичных независимых равновероятных символов. Ошибки в канале приводят к изменению значений некоторых символов на обратные, причем вероятность ошибки в k -м символе зависит лишь от наличия ошибки в предыдущем $(k-1)$ -м символе: она равна 0,2 при наличии ошибки в $(k-1)$ -м символе и 0,05 при ее отсутствии. Найти среднее количество передаваемой информации в расчете на символ.

22. Пусть $\xi, \eta \in \mathbb{R}^N$ – два независимых случайных вектора, причем η – случайный вектор с независимыми компонентами: $\text{Cov}\{\eta, \eta\} = \text{diag}\{\lambda_1, \dots, \lambda_N\}$. Пусть $\zeta = \xi + \eta$. Доказать, что количество информации по Шенону удовлетворяет неравенству

$$I\{\xi, \zeta\} \leq \frac{1}{2} \sum_{i=1}^N \log \left(1 + \frac{D\{\xi_i\}}{\lambda_i} \right),$$

которое обращается в равенство, если ξ_1, \dots, ξ_N – независимые гауссовские случайные величины.

Пусть ξ_1, \dots, ξ_m – максимальная невырожденная подсистема системы $n > m$ случайных символов ξ_1, \dots, ξ_n . Доказать, что $I\{\xi_1, \dots, \xi_m, \eta\} = I\{\xi_1, \dots, \xi_m, \eta\}$.

23. Используя понятия энтропии и информации по Шенону, оценить, сколько вопросов надо задать, чтобы отгадать задуманное собеседником натуральное число, не превосходящее N , если он дает лишь двоичные ответы: «да» или «нет».

24. Имеется N монет одного достоинства; одна из них – фальшивая – легче или тяжелее остальных. Используя понятия энтропии и информации по Шенону, оценить наименьшее число k взвешиваний на чашечных весах без гирь, которое позволяет найти фальшивую монету и определить, легче она или тяжелее остальных.

Г л а в а 8

ЭЛЕМЕНТЫ ТЕОРИИ СЛОЖНОСТИ ВЫЧИСЛЕНИЙ

8.1. ВЫЧИСЛИТЕЛЬНЫЕ ЗАДАЧИ

Пусть, как обычно, $\{0, 1\}^*$ – множество всех слов конечной длины в двоичном алфавите. Для слова $x \in \{0, 1\}^*$ через $|x|$ обозначаем его длину, $\{0, 1\}^l$ – множество слов длины l , \perp – пустое слово (длины 0), α^l – слово, составленное из l экземпляров символа $\alpha \in \{0, 1\}$.

Под *вычислительной задачей* понимают задачу вычисления значений функции $f: \{0, 1\}^* \rightarrow \{0, 1\}^*$. Аргумент x называют *входными данными* или просто *входом* задачи, значение $f(x)$ – *выходными данными*, *выходом* или *ответом*.

Будем считать, что входы и выходы отличаются от пустого слова \perp . Это слово зарезервируем для описания различных ошибочных ситуаций при вычислениях.

Входы и выходы задачи могут представлять различные объекты: числа, многочлены, векторы, графы, функции и др. Входными данными интересующих нас задач, как правило, являются наборы натуральных чисел. Эти числа будем представлять их двоичной записью: слово $a_1a_2 \dots a_l$ кодирует

число $a = \sum_{i=1}^l a_i 2^{l-i}$. Длину l кодового представления обычно будем выбирать

минимально возможной:

$$l = \lfloor \log_2 a \rfloor + 1.$$

Если x представляет только a , то длина $l = |x|$ известна и a легко восстанавливается по x . Если же a является только частью x , то для восстановления a можно включить в x кодовое представление l . Например, можно закодировать число l серией из $l - 1$ нулей, и тогда получится гамма-код Элиаса. Это двоичное слово из не более чем $2\lfloor \log_2 a \rfloor + 1$ символов, которое однозначно представляет a . Существуют и другие способы кодирования натуральных чисел. Будем использовать те, которые дают кодовые слова длины $O(\log a)$.

Код целого числа b – это код его знака, дополненный в случае $b \neq 0$ кодом натурального $|b|$. Код рационального числа a/b – это объединение кодов a и b . Понятно, что можно составить разумные правила кодирования произвольных структурированных наборов данных. Интересно, что одно из таких правил – так называемая абстрактно-синтаксическая нотация версии 1

(АСН.1) – часто используется в криптографии для описания ключей, долговременных параметров, сообщений протоколов и пр.

Область определения функции f может включать не все двоичные слова, т. е. f может быть *частичной* (а не полной) функцией на $\{0, 1\}^*$. Исключение определенных входов связано со спецификой задачи и особенностями кодирования входных данных. Будем по возможности расширять область определения f до $\{0, 1\}^*$, устанавливая, например, что все недопустимые входы представляют один и тот же фиксированный допустимый вход. Тем не менее в необходимых случаях будем подчеркивать, что f является частичной функцией.

8.2. ЗАДАЧИ РАСПОЗНАВАНИЯ И ПОИСКА

В криптографии работают с вычислительными задачами двух типов: *распознавания* и *поиска*.

В задаче распознавания функция f может принимать только два значения: 1 (да) или 0 (нет). Такую функцию называют *предикатом*. С предикатом удобно отождествить *язык* (множество слов)

$$L = f^{-1}(1) = \{x \in \{0, 1\}^*: f(x) = 1\}$$

и вести речь о проверке принадлежности входа x языку L , т. е. о распознавании L .

Известной задачей распознавания является задача проверки простоты числа. Язык этой задачи:

$$PRIMES = \{p \in \mathbb{N}: p \text{ – простое}\}.$$

Здесь и далее мы предполагаем, что натуральные числа и другие объекты неявно кодируются двоичными словами.

В задаче поиска описывается множество $R \subseteq \{0, 1\}^* \times \{0, 1\}^*$. Входу x соответствует не один, а целое множество выходов – это элементы множества $R(x) = \{y \in \{0, 1\}^*: (x, y) \in R\}$. Функция f задается неявно. Говорят, что f решает R , если $f(x) \in R(x)$. Множество $R(x)$ может оказаться пустым, и тогда f не определена в точке x , т. е. f , вообще говоря, является частичной функцией.

Следующие задачи поиска часто используются в криптографии.

Задача факторизации (Factor). Входом Factor является составное натуральное n , выходом – нетривиальный (т. е. отличный от 1 и n) делитель d числа n . Другими словами, $Factor(n) = \{d \in \mathbb{N}: 1 < d < n, d | n\}$.

Задача дискретного логарифмирования (DL). Пусть имеется семейство конечных циклических групп и G – представитель этого семейства. Пусть G описывается словом $descr(G)$, которое определяет поря-

док $q = |G|$, строение элементов G и групповую операцию над ними. В криптографии используются два основных способа выбора (или, как еще говорят, *инстанцирования*) группы G . Во-первых, в качестве G может быть выбрана подгруппа мультипликативной группы простого поля \mathbb{F}_p : $G \subseteq \mathbb{F}_p^*$, $q \mid p - 1$. В этом случае для полного описания G достаточно использовать пару (p, q) . Во-вторых, G может быть группой точек эллиптической кривой над конечным полем (подробнее см. гл. 4).

Входными данными DL является тройка $(\text{descr}(G), g, g^a)$, где g – образующий G , $a \in \{0, 1, \dots, q - 1\}$. Фигурирующее здесь число a называется *дискретным логарифмом* g^a по основанию g и является выходом DL.

Задача Диффи – Хеллмана (CDH). В задаче CDH также используется циклическая группа G порядка q , ее образующий g и произвольный элемент g^a . Используется дополнительный элемент g^b , $b \in \{0, 1, \dots, q - 1\}$. Требуется по $(\text{descr}(G), g, g^a, g^b)$ найти g^{ab} .

На самом деле существует целое семейство задач типа Диффи – Хеллмана. Мы определили только одну из них – так называемую вычислительную (computational). Отсюда буква С в названии задачи.

На связанных между собой задачах DL и CDH базируются два больших направления в криптографии: FFC (Finite Field Cryptography, случай $G \subset \mathbb{F}_p^*$) и ECC (Elliptic Curve Cryptography, выбор в качестве G группы точек эллиптической кривой). На задаче Factor базируется направление IFC (Integer Factorization Cryptography).

8.3. МАШИНА ТЬЮРИНГА

Для решения задач служат алгоритмы. Неформально говоря, алгоритм – это однозначно определенная последовательность инструкций по преобразованию входных данных в выходные.

Приведенное определение не является математически строгим (что такое инструкция? что значит однозначно определенная?). Для полной формализации используются модели вычислительных устройств, реализующих алгоритмы. Наиболее известная модель – *машина Тьюринга* (MT), предложенная английским математиком А. Тьюрингом в 1936 г. Согласно общепризнанному тезису Чёрча – Тьюринга любой алгоритм в интуитивном смысле этого слова может быть реализован некоторой машиной Тьюринга. Другими словами, принято отождествлять алгоритмы и MT.

Машина Тьюринга M характеризуется следующими элементами.

1. *Лента.* Лента представляет собой набор ячеек, пронумерованных $1, 2, \dots$. Число ячеек не ограничено. В каждой ячейке хранится символ алфавита $\Sigma = \{0, 1, \dots\}$. Символы ячеек ленты образуют бесконечное вправо слово $a = a_1 a_2 \dots$

2. Управляющее устройство. Управляющее устройство выполняет манипуляции над символами ленты в зависимости от своего состояния $q \in Q$, где Q – конечное множество. Состояния меняются по правилам, заданным функцией переходов $\varphi: Q \times \Sigma \rightarrow Q \times \Sigma \times \{-1, 0, 1\}$. Определено начальное состояние q_0 .

3. Головка управляющего устройства. Головка передвигается по ленте и в конкретный момент времени находится над ячейкой с номером $d \in \mathbb{N}$. Управляющее устройство может читать символ ячейки под головкой и менять только его. Все организовано так, что вычисления меняют физическую среду (ленту) локально. Это основной тезис интуитивных представлений о вычислениях.

Состояние всей машины задается тройкой (a, q, d) . Управляющее устройство меняет состояния, выполняя следующие такты вычислений:

- 1) прочитывается символ a_d , находящийся под головкой;
- 2) вычисляется тройка $(q', a'_d, \Delta d) \leftarrow \varphi(q, a_d)$;
- 3) символ a'_d записывается в ячейку с номером d (вместо a_d);
- 4) состояние управляющего устройства меняется на q' ;
- 5) головка сдвигается на Δd позиций;
- 6) если $d + \Delta d = 0$, то машина останавливается.

В начале работы M в первые ячейки ленты записываются входные данные $x \in \{0, 1\}^*$, а остальные ячейки заполняются символом $_$, т. е. первонациально $a = x_ \dots$. Управляющее устройство начинает работу в состоянии q_0 , а головка устанавливается над первой ячейкой. После остановки машины слово, записанное на ленте, имеет вид $y_ \dots$, где $y \in \{0, 1\}^*$. Слово-префикс y объявляется выходными данными M на входе x : $y = M(x)$.

Для работы машины M может потребоваться бесконечно много ячеек на ленте. С другой стороны, для исчерпывающего описания M достаточно указать **конечные объекты**: множество Q , функцию переходов φ и начальное состояние s_0 . Таким образом, описание M можно закодировать двоичным словом $[M]$ конечной длины.

К сожалению, МТ является примитивным устройством и код его описания весьма неудобен. Далее мы будем использовать высокоуровневый способ кодирования, включающий стандартные алгоритмические конструкции типа **if – then – else, while, for**. При некоторых ограничениях программы на таких языках могут быть преобразованы к описанию $[M]$ с помощью специальной МТ (компилятора).

Будем задавать алгоритмические конструкции на русском языке, не придерживаясь жестких синтаксических ограничений. В наших описаниях алгоритмов обязательно будут рубрики «Вход», «Выход», «Шаги». Например, алгоритм Евклида определяется следующим образом.

АЛГОРИТМ ЕВКЛИДА

Вход: натуральные a и b , $a \geq b$.

Выход: НОД(a, b).

Шаги:

1. Пока $b \neq 0$ выполнить:

 1.1. $a \leftarrow a \bmod b$;

 1.2. $a \leftrightarrow b$.

2. Возвратить a .

8.4. РАЗРЕШИМЫЕ И НЕРАЗРЕШИМЫЕ ЗАДАЧИ

Пусть M останавливается на входе x с результатом y . Результат определен однозначно и, таким образом, M определяет функцию $f_M: x \mapsto y$. Функция f_M является частичной, ее значения не определены для тех входов x , на которых M не останавливается.

Определение 8.1. Частичная функция $f: \{0, 1\}^* \rightarrow \{0, 1\}^*$ вычислима (задача f разрешима), если найдется машина Тьюринга M такая, что $f = f_M$ (совпадают и области определения, и значения функций). Говорят, что M вычисляет функцию f (решает задачу f).

Для задач поиска последняя часть определения уточняется: машина M решает задачу поиска R , если f_M решает R .

Теорема 8.1. Существуют неразрешимые задачи (невычислимые функции).

Доказательство. Функций $f: \{0, 1\}^* \rightarrow \{0, 1\}^*$ больше, чем описаний машин Тьюринга $[M] \in \{0, 1\}^*$. \square

Возможно, самый известный пример неразрешимой задачи – это задача об остановке, предложенная самим Тьюрингом. Входными данными задачи является описание $[M]$ машины M и слово $x \in \{0, 1\}^*$. Выход: 1, если M останавливается на входе x , и 0 в противном случае.

Теорема 8.2. Задача об остановке неразрешима.

Доказательство. Пусть h – предикат задачи об остановке: $h([M], x) = 1$ тогда и только тогда, когда M остановится на входе x . Предположим, от противного, что предикат h вычислим на некоторой машине A .

Построим машину B , которая обрабатывает вход $[M]$ следующим образом: если $A([M], [M]) = 1$, то B переходит в бесконечный цикл; в противном случае B возвращает 0. Описание B незначительно расширяет описание A : добавляется проверка выхода и при необходимости выполняется переход в бесконечный цикл.

Подадим на вход B ее описание $[B]$. Если B остановится, то $A([B], [B]) = 0$ и остановки быть не должно по определению h . Если B не остановится, то $A([B], [B]) = 1$ и, наоборот, остановка должна быть. Найденные противоречия доказывают требуемый результат. \square

Еще одна знаменитая неразрешимая задача – это десятая проблема Гильберта. Входными данными здесь является многочлен от нескольких переменных с целыми коэффициентами (например, $x^n + y^n - z^n$, $n \in \mathbb{N}$). Требуется определить, имеет многочлен целочисленные корни или нет. Неразрешимость десятой проблемы Гильберта доказал в 1971 г. советский математик Ю. Матиясевич.

8.5. РЕСУРСЫ

Для решения задач требуются ресурсы. Будем рассматривать два типа ресурсов: *время* и *память*. Нас будет интересовать их пиковое потребление. Пусть x – допустимый вход машины M в том смысле, что M останавливается на этом входе. Через $t_M(x)$ обозначим число тактов, которое выполнит M при вычислениях на входе x , а через $s_M(x)$ – крайнюю правую позицию головки при вычислениях.

Определение 8.2. Машина Тьюринга M работает за время $T_M(l)$, $l \in \mathbb{N}$, если $T_M(l) = \max_x t_M(x)$, где максимум берется по допустимым $x \in \{0, 1\}^l$. Аналогично M работает на памяти $S_M(l)$, если $S_M(l) = \max_x s_M(x)$.

Ясно, что $S_M(l) \leq T_M(l)$ для любой M . Поэтому время работы является более универсальной характеристикой сложности, чем память.

Будем рассматривать характеристики сложности в асимптотике $l \rightarrow \infty$. Напомним обозначения, которые касаются сравнения скорости роста функций:

- $u(l) = o(v(l))$, если $u(l)/v(l) \rightarrow 0$;
- $u(l) = O(v(l))$, если $|u(l)| \leq Cv(l)$ для некоторой константы C и всех достаточно больших l ;
- $u(l) = \Omega(v(l))$, если $v(l) = O(u(l))$;
- $u(l) = \Theta(v(l))$, если $u(l) = O(v(l))$ и $v(l) = O(u(l))$.

Говорят, что M работает за *полиномиальное* время или является *полиномиальной* машиной Тьюринга (ПМТ), если найдется $c \in \mathbb{N}$ такое, что $T_M(l) \leq l^c$ для всех достаточно больших l . Последнее условие можно записать по-другому: $T_M(l) = l^{O(1)}$. Среди ПМТ выделяют машины, которые работают за *линейное* время: $T_M(l) = \Theta(l)$, за *квазилинейное*: $T_M(l) = l(\log l)^{O(1)}$, за *квадратичное*: $T_M(l) = \Theta(l^2)$, за *кубическое*: $T_M(l) = \Theta(l^3)$ и т. д. Согласно распространенному тезису Кобхэма полиномиальные алгоритмы и только они являются *эффективными*, т. е. имеющими практическое значение.

Если время работы M нельзя ограничить многочленом, т. е. $T_M(l) = \Omega(l^c)$ для всех $c \in \mathbb{N}$, то говорят, что M работает за *суперполиномиальное* время. Среди суперполиномиальных машин выделяют *квазиполиномиальные*: $T_M(l) = 2^{(\log l)^{O(1)}}$, *субэкспоненциальные*: $T_M(l) = 2^{o(l)}$ и *экспоненциальные*: $T_M(l) = 2^{l^{O(1)}}$. Есть и *суперэкспоненциальные* машины, но мы их рассматривать не будем.

Оценки времени работы алгоритмов, решающих базовые арифметические задачи с натуральными числами, приводятся в таблице.

Базовые арифметические задачи

Задача ($a \geq b$)	Время
Сложение $(a, b) \mapsto a + b$	$O(\log a + \log b)$
Вычитание $(a, b) \mapsto a - b$	$O(\log a + \log b)$
Умножение $(a, b) \mapsto ab$	$O(\log a \log b)$
Деление $(a, b) \mapsto (q, r): a = qb + r, 0 \leq r < b$	$O(\log q \log b)$
Наибольший общий делитель $(a, b) \mapsto \text{НОД}(a, b)$	$O(\log^2 a)$
Обращение по модулю $(a, b) \mapsto b^{-1} \bmod a$ (a и b взаимно просты)	$O(\log^2 a)$
Возведение в степень по модулю $(a, b, e) \mapsto b^e \bmod a$	$O(\log e \log^2 a)$
Китайская система сравнений $(a_1, \dots, a_k, b_1, \dots, b_k) \mapsto b: b \equiv b_i \pmod{a_i}$ (a_i попарно взаимно просты)	$O(\log^2 a)$ ($a = a_1 \dots a_k$)
Совершенная степень $a \mapsto (b, e): a = b^e$, где e максимально	$O((\log a)^{1+o(1)})$

Оценки могут уточняться. Например, умножение может быть выполнено за время $O((\log a)^{\log_2 3})$ с помощью алгоритма Карацубы – Офмана или даже за время $O(\log a \cdot \log \log a \cdot \log \log \log a)$ с помощью алгоритма Шёнхаге – Штрассена. Тем не менее даже приведенные оценки означают эффективность базовых арифметических операций.

8.6. ВЕРОЯТНОСТНЫЕ МАШИНЫ

Пусть p – нечетное простое, $Q_p = \{b^2 : b \in \mathbb{F}_p^*\}$ – множество квадратичных вычетов по модулю p , $\bar{Q}_p = \mathbb{F}_p^* \setminus Q_p$ – множество квадратичных невычетов.

Рассмотрим задачу поиска квадратичного невычета по заданному модулю p . Для решения этой задачи можно использовать следующий алгоритм.

Алгоритм Поиск невычета (ДЕТЕРМИНИРОВАННЫЙ)

Вход: p – нечетное простое.

Выход: $a \in \bar{Q}_p$.

Шаги:

1. Для $a = 1, 2, \dots, p - 1$:

 1.1. Если $a^{(p-1)/2} \equiv -1 \pmod{p}$, то возвратить a .

Алгоритм обязательно остановится с верным результатом в силу следующих фактов:

1) $a \in \bar{Q}_p$ тогда и только тогда, когда $a^{(p-1)/2} \equiv -1 \pmod{p}$ (критерий Эйлера);

2) $|Q_p| = |\bar{Q}_p| = (p - 1)/2$.

При попытке обосновать эффективность алгоритма возникают трудности. С одной стороны, проверка на шаге 1.1 выполняется эффективно (за кубическое от $\log p$ времени). С другой стороны, нельзя гарантировать, что потребуется полиномиальное число таких проверок. Пусть $n_2(p)$ – минимальный квадратичный невычет по модулю p . Доказано (теорема Бургесса), что $n_2(p) \leq p^{1/(4\sqrt{e})+\varepsilon}$, $\varepsilon > 0$. С такой оценкой время работы алгоритма будет экспоненциальным. Более сильная оценка $n_2(p) \leq 2 \ln^2 p$, справедливая при выполнении расширенной гипотезы Римана, дает полиномиальное время. К сожалению, гипотеза на сегодня не доказана. Можно организовать поиск a не в начале натурального ряда, можно учитывать при поиске вид p (например, $p - 1$ будет невычетом, если $p \equiv 3 \pmod{4}$), но в целом трудности обоснования эффективности алгоритма сохранятся.

Выходом является переход к вероятностному поиску.

Алгоритм Поиск невычета (ВЕРОЯТНОСТНЫЙ)

Вход: p – нечетное простое.

Выход: $a \in \bar{Q}_p$.

Шаги:

1. Для $i = 1, 2, \dots$:

 1.1. $a \xleftarrow{R} \{1, 2, \dots, p - 1\}$;

 1.2. Если $a^{(p-1)/2} \equiv -1 \pmod{p}$, то возвратить a .

В новом алгоритме запись $u \xleftarrow{R} U$ означает случайный равновероятный выбор u из множества U . Выбранное на шаге 1.1 число a окажется невычетом с вероятностью $1/2$, и в среднем потребуется всего 2 попытки генерации.

Алгоритмы, в которых используется случайный выбор, называются *вероятностными*. Моделью таких алгоритмов является *вероятностная машина Тьюринга* (ВМТ). ВМТ снабжается дополнительной лентой, в ячейках которой записаны символы слова $\omega = \omega_1\omega_2\dots$. Эти символы являются реализациями независимых в совокупности случайных величин с равномерным распределением на $\{0, 1\}$. Ячейки случайной ленты читаются слева направо, без повторений. В функцию переходов φ , определяющую функционирование управляющего устройства машины, добавляется еще один аргумент – очередной символ ω .

Число тактов и сам результат работы M на фиксированном входе x являются случайными величинами. В частности, машина может допускать ошибки в определении правильного ответа, т. е. решать задачу не наверняка, а только с какой-то вероятностью. Мы будем рассматривать три типа ошибок.

1. *Нульсторонние ошибки*. Машина M возвращает либо правильный ответ, либо пустое слово \perp , которое означает «ответ не найден».

2. *Односторонние ошибки*. Машина M решает задачу распознавания языка L , не ошибаясь на входах $x \in L$ и может ошибаться на входах $x \notin L$. Возможно альтернативное определение: M не ошибается на входах $x \notin L$ и может ошибаться на входах $x \in L$.

3. *Двусторонние ошибки*. Машина M решает задачу распознавания языка L и может ошибаться как на входах $x \in L$, так и на входах $x \notin L$.

Введем характеристики трудоемкости работы вероятностных машин.

Определение 8.3. *ВМТ M работает за среднее время $ET_M(l)$, $l \in \mathbb{N}$, если $ET_M(l) = \max_x \mathbf{E} t_M(x)$. Здесь максимум берется по допустимым $x \in \{0, 1\}^l$, а $\mathbf{E} t_M(x)$ – среднее число тактов, которое выполнит M при вычислениях на входе x . Усреднение выполняется по всевозможным заполнениям случайной ленты.*

Определение 8.4. *ВМТ M решает задачу f с вероятностью $P_M(l)$, $l \in \mathbb{N}$, если $P_M(l) = \min_x \mathbf{P}\{M(x) = f(x)\}$, где минимум берется по допустимым $x \in \{0, 1\}^l$.*

Если для $T_M(l)$, $ET_M(l)$ важна полиномиальная ограниченность, то для $P_M(l)$ важно быть достаточно большой. Будем говорить, что M работает с преобладающей вероятностью успеха, если $P_M(l) \geq 1 - \nu(l)$, где ν – пренебрежимо малая функция.

Определение 8.5. *Функция $\nu: \mathbb{N} \rightarrow [0, 1]$ пренебрежимо мала, если для любого $c \in \mathbb{N}$ неравенство $\nu(l) < l^{-c}$ выполняется при всех достаточно больших $l \in \mathbb{N}$.*

Будем также говорить, что M работает с *обратно полиномиальной* вероятностью успеха, если найдется $c \in \mathbb{N}$ такое, что $1/P_M(l) \leq l^c$ при всех достаточно больших l .

8.7. АЛГОРИТМЫ ЛАС-ВЕГАС И МОНТЕ-КАРЛО

Вероятностный алгоритм, который мы рассмотрели в п. 8.6, всегда возвращает правильный ответ и работает за полиномиальное в среднем время. Такие алгоритмы принято называть *алгоритмами Лас-Вегас*.

Даже если среднее время работы вероятностной машины M полиномиально, вычисления на определенных входах при определенных ω могут выполняться суперполиномиально долго. В некоторых случаях время работы необходимо ограничивать. *Полиномиальная вероятностная машина Тьюринга* (ПВМТ) – это ВМТ, которая всегда работает за полиномиальное время независимо от заполнения случайной ленты. Платой за ограничение по времени является возможность появления ошибок в ответах.

Полиномиальным ВМТ соответствуют *алгоритмы Монте-Карло*. Вот пример такого алгоритма.

АЛГОРИТМ ПОИСК НЕВЫЧЕТА (ПОЛИНОМИАЛЬНЫЙ ВЕРОЯТНОСТНЫЙ)

Вход: p – нечетное простое.

Выход: $a \in \bar{Q}_p$ или \perp .

Шаги:

1. Для $i = 1, \dots, k$:

1.1. $a \xleftarrow{R} \{1, 2, \dots, p-1\}$;

1.2. Если $a^{(p-1)/2} \equiv -1 \pmod{p}$, то возвратить a .

2. Возвратить \perp .

Здесь $k = (\log p)^{O(1)}$ – параметр, который определяет время работы ($O(k \log^3 p)$) и вероятность получения правильного ответа $(1 - 2^{-k})$. Ошибки алгоритма могут быть только нульсторонними. Рассмотрим такие алгоритмы подробнее.

Пусть A – ВМТ, которая допускает только нульсторонние ошибки. Тогда A может быть преобразована в машину B , которая последовательно обращается к A до тех пор, пока не будет получен ответ, отличный от \perp . Машина B не ошибается. Если $\beta > 0$ – вероятность успеха A на некотором входе, то B на том же входе потребуется выполнить $1/\beta$ обращений к A в среднем.

Действительно, среднее число обращений

$$\sum_{t=1}^{\infty} t \mathbf{P}\{\text{нужно } t \text{ обращений}\} = \sum_{t=0}^{\infty} \mathbf{P}\{\text{нужно } >t \text{ обращений}\} = \sum_{t=0}^{\infty} (1 - \beta)^t = \frac{1}{\beta}.$$

Отсюда получаем оценку для среднего времени работы B :

$$ET_B(l) \leq \frac{T_A(l)}{P_A(l)}.$$

Если A работает за полиномиальное время с обратно полиномиальной вероятностью успеха, то B работает за полиномиальное в среднем время. Таким образом, переход от A к B есть преобразование алгоритма Монте-Карло в алгоритм Лас-Вегас. Возможно обратное преобразование, которое состоит в принудительной остановке B после определенного числа тактов работы с возвратом \perp после остановки. Фактически этот прием применен в нашем последнем алгоритме поиска невычета.

Машине B может запустить A только k раз, снова ожидая ответ, отличный от \perp . Искомый ответ будет получен с вероятностью $\beta_B = 1 - (1 - \beta)^k$ за β_B/β обращений к A в среднем. Машине B , как и A , допускает нульсторонние ошибки, но их вероятность быстро уменьшается с ростом k .

Предположим теперь, что A решает задачу распознавания языка L и допускает односторонние ошибки:

$$\mathbf{P}\{A(x) = 1\} = \begin{cases} 1, & x \in L, \\ \varepsilon, & x \notin L. \end{cases}$$

Здесь $0 < \varepsilon < 1$. Построим машину B , который получает на вход x , запускает A на этом входе k раз и фиксирует ответы y_1, \dots, y_k . Если все $y_i = 1$, то B возвращает 1. Если хотя бы один ответ $y_i = 0$, то B возвращает 0. Машине B также распознает язык L , также допускает односторонние ошибки, но уже с меньшими вероятностями:

$$\mathbf{P}\{B(x) = 1\} = \begin{cases} 1, & x \in L, \\ \varepsilon^k, & x \notin L. \end{cases}$$

С ростом k время работы B увеличивается линейно, а вероятность ошибки уменьшается экспоненциально. Изменяя k , можно контролировать качество распознавания, сохраняя приемлемым время работы. Такой подход применяется в вероятностных алгоритмах проверки простоты.

Снизить можно и вероятности двусторонних ошибок. Пусть A допускает двусторонние ошибки и вероятность ошибки на входе x не превосходит $\varepsilon < 1/2$. Перестроим алгоритм B так, чтобы он определял свой ответ по y_1, \dots, y_k , руководствуясь *правилом большинства*: если среди y_i единиц больше, чем нулей, то B возвращает 1, в противном случае B возвращает 0. Тогда вероятность ошибки B на входе x не превышает

$$\sum_{i=\lceil k/2 \rceil}^k \binom{k}{i} \varepsilon^i (1-\varepsilon)^{k-i} \leq \exp\left(-k \frac{(1/2 - \varepsilon)^2}{(1/2 + \varepsilon)}\right).$$

Последняя оценка следует из неравенства Чернова (см. задание 5).

8.8. СВЕДЕНИЕ

Пусть g – некоторая задача (распознавания или поиска). *Оракульная машина Тьюринга* M^g – это машина Тьюринга, снабженная дополнительной лентой, на которую можно записать допустимый вход задачи g и за один такт получить на его месте ответ. Символы оракульной ленты являются дополнительными входами и выходами функции переходов M^g .

Оракульная машина описывает работу обычной машины с гипотетическим внешним вычислительным устройством – *оракулом*. Считается, что ресурсы оракула неограничены и он вычисляет ответ задачи g в течение одного такта работы M^g . Время работы M^g учитывает число запросов к оракулу, но игнорирует сложность их обработки.

Определение 8.6. Говорят, что имеется полиномиальная сводимость задачи f к задаче g и пишут $f \leqslant_P g$, если найдется оракульная ПМТ M^g , которая решает f . Если $f \leqslant_P g$ и $g \leqslant_P f$, то f и g полиномиально эквивалентны: $f \sim_P g$.

Определение 8.7. Говорят, что имеется вероятностная полиномиальная сводимость задачи f к задаче g и пишут $f \leqslant_R g$, если найдется оракульная ПВМТ M^g , которая решает f с преобладающей вероятностью успеха. Если $f \leqslant_R g$ и $g \leqslant_R f$, то f и g вероятностно полиномиально эквивалентны: $f \sim_R g$.

Сведёние устанавливает частичный предпорядок на множестве задач. Доказывая факт сведёния, мы устанавливаем тем самым, что одна задача в некотором смысле не сложнее другой.

Проиллюстрируем сведёние на примере соотношения между задачами Factor и Sqrt. Sqrt – это задача поиска квадратных корней по модулю. Входными данными Sqrt является пара взаимно простых натуральных чисел (n, a) , в которой a – квадратичный вычет по модулю n . Выходными данными является число $b \in \mathbb{Z}_n^*$ такое, что $b^2 \equiv a \pmod{n}$. Выход b называют *квадратным корнем из a по модулю n* . Корней может быть несколько. Запись $b \in \sqrt{a} \pmod{n}$ означает, что b – один из этих корней.

Теорема 8.3. $\text{Sqrt} \leqslant_R \text{Factor}$.

Доказательство. Для решения $\text{Sqrt}(n, a)$ представим n в виде $\prod_{i=1}^k p_i^{e_i}$, где p_i – различные простые; e_i – натуральные числа. Для этого выполним $O(\log n)$ обращений к оракулу Factor, последовательно разлагая n на множители до тех пор, пока все они не окажутся простыми. Простоту множителя можно проверить за полиномиальное время (см. п. 8.9).

Если мы найдем $b_i \in \sqrt{a} \pmod{p_i^{e_i}}$, то сможем определить b как решение китайской системы сравнений

$$b \equiv \pm b_i \pmod{p_i^{e_i}}, \quad i = 1, 2, \dots, k.$$

Решение можно найти за время $O(\log^2 n)$ (см. таблицу на с. 220). Знаки в системе можно расставлять произвольно. Отметим (нам это потребуется немного позже), что если все p_i нечетны, то имеется 2^k различных корней b , определяемых различными вариантами расстановки знаков.

Корень $r \in \sqrt{a} \pmod{p^e}$, $p \in \{p_1, \dots, p_k\}$, можно найти следующим образом.

1. Для $p \neq 2$ определим невычет по модулю p . Для этого используем алгоритм из п. 8.7 с числом итераций k . Алгоритм работает за время $O(k \log^3 p)$ с вероятностью успеха $1 - 2^{-k}$.

2. Для $p \neq 2$ определим корень $r_0 \in \sqrt{a} \pmod{p}$. Для этого используем алгоритм Тонелли – Шэнкса. На вход алгоритма передаются p , a и невычет, найденный на предыдущем шаге. Алгоритм работает за время $O(\log^4 p)$. Если $p = 2$, то $a \equiv 1 \pmod{p}$ и $r_0 = 1$.

3. Последовательно определим корни $r_i \in \sqrt{a} \pmod{p^{2^i}}$, $i=1, 2, \dots, \lceil \log_2 e \rceil$, а затем найдем $r = r_{\lceil \log_2 e \rceil} \pmod{p^e}$. Корень r_i будем искать в виде $r_i = r_{i-1} + p^{2^{i-1}} s$. Должно выполняться сравнение

$$(r_{i-1} + p^{2^{i-1}} s)^2 = r_{i-1}^2 + 2p^{2^{i-1}} r_{i-1} s + p^{2^i} s^2 \equiv a \pmod{p^{2^i}},$$

откуда

$$s = \left(\frac{(a - r_{i-1}^2)/p^{2^{i-1}}}{2r_{i-1}} \right) \pmod{p^{2^{i-1}}}$$

(все деления нацело). Время расчетов – $O(e^2 \log^2 p \log e)$.

В целом все корни b_i будут найдены за время

$$\sum_{i=1}^k O(k \log^3 p_i + \log^4 p_i + e_i^2 \log^2 p_i \log e_i) = O(\log^5 n)$$

с вероятностью успеха $(1 - 2^{-k})^k \geq 9/16$. Алгоритм нахождения корней допускает только нульсторонние ошибки. Повторяя алгоритм полиномиальное число раз, можно сделать вероятность успеха преобладающей. \square

Теорема 8.4. Factor $\leq_R \text{Sqrt}$.

Доказательство. Пусть требуется найти нетривиальный делитель d составного n . Если n – четное, то найти решение легко: $d = 2$. Поэтому будем считать, что n – нечетное. Если n имеет вид p^e , где p – простое, то p можно найти за время $O((\log p)^{1+o(1)})$ (см. таблицу на с. 220).

Таким образом, мы можем ограничиться случаем, когда n имеет не менее двух различных нечетных простых делителей.

В этом случае, как было отмечено в предыдущем доказательстве, из любого квадратичного вычета по модулю n извлекается не менее четырех различных квадратных корней.

Пусть A – алгоритм, который решает Sqrt . Следующий алгоритм решает Factor с помощью A .

1. Сгенерировать $c \xleftarrow{R} \{2, 3, \dots, n - 1\}$.
2. Если $\text{НОД}(n, c) \neq 1$, то возвратить $\text{НОД}(n, c)$.
3. Установить $a \leftarrow c^2 \bmod n$.
4. Установить $b \leftarrow A(n, a)$.
5. Если $b \equiv \pm c \pmod{n}$, то возвратить \perp .
6. Возвратить $\text{НОД}(b + c, n)$.

Построенный алгоритм является вероятностным полиномиальным. Продолжим анализировать результаты его работы. Ясно, что на шаге 2 возвращается нетривиальный делитель n . Число b , полученное на шаге 4, удовлетворяет сравнениям $0 \equiv b^2 - a \equiv b^2 - c^2 \equiv (b - c)(b + c) \pmod{n}$. Поэтому n делит произведение $(b - c)(b + c)$. Если условие на шаге 5 не выполняется, то n не делит ни один из множителей произведения и $\text{НОД}(b + c, n)$ – нетривиальный делитель n .

Всего имеется не менее 4 корней из a по модулю n . Поэтому условие на шаге 5 будет нарушаться не менее чем в половине случаев и искомый делитель n будет найден с вероятностью

$$\begin{aligned}\beta &= \mathbf{P}\{\text{НОД}(n, c) \neq 1\} + \mathbf{P}\{\text{НОД}(n, c) = 1, b \not\equiv \pm c \pmod{n}\} \geq \\ &\geq \mathbf{P}\{\text{НОД}(n, c) \neq 1\} + \frac{1}{2} \mathbf{P}\{\text{НОД}(n, c) = 1\} > \frac{1}{2}.\end{aligned}$$

Повторяя алгоритм полиномиальное число раз, можно сделать вероятность успеха преобладающей. \square

Следствие 8.1. $\text{Sqrt} \sim_R \text{Factor}$.

8.9. КЛАССЫ СЛОЖНОСТИ

В этом пункте мы обсудим ряд важных аспектов сложности вычислений. Эти аспекты связаны только с задачами распознавания. Нас не должно это смущать, поскольку задачи поиска, которые встречаются в криптографии, как правило, легко сводятся к задачам распознавания. Например, Factor сводится к распознаванию языка

$$L_{\text{Factor}} = \{(n, m) : n, m \in \mathbb{N}, \exists d \in \{2, 3, \dots, m\}, \text{которое делит } n\}.$$

Действительно, нетривиальный делитель d числа n можно найти, обращаясь $O(\log n)$ раз к машине, которая распознает L_{Factor} , т. е. проверяет, что $2 \leq d \leq m$. Граница m задается при этих обращениях так, чтобы реализовать дихотомию множества $\{2, 3, \dots, n - 1\}$.

Будем рассматривать далее задачу распознавания с предикатом f и языком L .

Определение 8.8. Язык L (предикат f) принадлежит классу \mathbf{P} , если существует ПМТ M такая, что:

- 1) если $x \in L$, то $M(x) = 1$;
- 2) если $x \notin L$, то $M(x) = 0$.

Определение 8.9. Язык L (предикат f) принадлежит классу \mathbf{NP} , если существует ПМТ M и многочлен p такие, что:

- 1) если $x \in L$, то $M(x, y) = 1$ для некоторого слова y длины $|y| \leq p(|x|)$;
- 2) если $x \notin L$, то $M(x, y) = 0$ для любого y длины $|y| \leq p(|x|)$.

Класс \mathbf{P} – это класс языков, распознаваемых за полиномиальное время. В \mathbf{NP} входят языки, которые также распознаются за полиномиальное время, но с использованием дополнительных данных – слов y . Слово y , которое часто называется *сертификатом*, подтверждает принадлежность $x \in L$, причем подтверждает доказательно, в том смысле, что для $x \notin L$ ни один из сертификатов не будет принят машиной M .

Знаменитая гипотеза $\mathbf{P} \neq \mathbf{NP}$ означает существование предиката f , для которого ответ $f(x)$ не может быть вычислен за полиномиальное время, но может быть проверен за полиномиальное время с помощью сертификата. Большинство специалистов считает, что гипотеза $\mathbf{P} \neq \mathbf{NP}$ справедлива.

В классе \mathbf{NP} выделяют подкласс \mathbf{NPC} так называемых \mathbf{NP} -полных языков. Языки из \mathbf{NPC} имеют максимальную сложность: $L \in \mathbf{NPC}$, если для любого $L' \in \mathbf{NP}$ найдется вычислимая на ПМТ функция $g: \{0, 1\}^* \rightarrow \{0, 1\}^*$ такая, что $x \in L'$ тогда и только тогда, когда $g(x) \in L$. Если $\mathbf{NPC} \cap \mathbf{P} \neq \emptyset$, то $\mathbf{P} = \mathbf{NP}$, и наоборот.

Рассмотрим несколько примеров.

Пример 8.1. Задача SAT (от англ. *satisfiability*) формулируется следующим образом. Задана булева формула, составленная из переменных, скобок, символов \neg (отрицание), \vee (и), \wedge (или). Требуется ответить, можно ли назначить переменным значения 1 (истина) и 0 (ложь) так, чтобы формула стала истинной. В начале 1970-х гг. С. Кук и Л. Левин получили результаты, которые означают, что SAT лежит в классе \mathbf{NPC} (фактически Кук и Левин ввели этот класс).

Пример 8.2. Задача $SubsetSum$ определяется языком

$$\left\{ (x_1, x_2, \dots, x_n, y) : x_i, y \in \mathbb{Z}, \exists S \subseteq \{1, 2, \dots, n\} \text{ т. ч. } \sum_{i \in S} x_i = y \right\}.$$

Доказано, что этот язык является \mathbf{NP} -полным.

Пример 8.3. Пусть G – игра, в которой участвуют Алиса и Боб. Конфигурация (состояние) игры кодируется словом $x \in \{0, 1\}^*$. Введем язык $L = \{x \in \{0, 1\}^* : \text{Алиса имеет выигрышную стратегию в конфигурации } x\}$.

Найдены примеры игр G , для которых $L \notin \text{NP}$. Для этих игр нельзя найти сертификат, который позволял бы проверить наличие выигрышной стратегии за полиномиальное время.

Еще один класс сложности связан с вероятностными алгоритмами.

Определение 8.10. Язык L принадлежит классу **BPP**, если существует ПВМТ M такая, что:

- 1) если $x \in L$, то $\mathbf{P}\{M(x) = 1\} \geq 2/3$;
- 2) если $x \notin L$, то $\mathbf{P}\{M(x) = 0\} \geq 2/3$.

Другими словами, языки из класса **BPP** распознаются на машинах Монте-Карло с двусторонними ошибками, вероятность которых $\leq 1/3$. Выбор порога $1/3$ условен. Важно только, чтобы вероятность ошибки не пре-восходила $1/2 - \delta$ для некоторого фиксированного $\delta > 0$. Применяя M для распознавания x несколько раз и вынося решение по правилу большинства (см. п. 8.7), можно достаточно быстро приблизить вероятность успешного распознавания к 1.

Гипотетическое (признаваемое большинством специалистов) соотношение между описанными классами сложности представлено на рисунке.

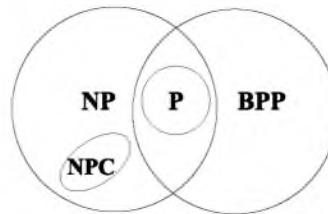


Диаграмма классов сложности (гипотетическая)

8.10. ЯЗЫК PRIMES

В этом пункте мы рассмотрим сложность распознавания языка **PRIMES**.

Теорема 8.5. $\text{PRIMES} \in \text{NP}$.

Доказательство. Пусть для $a \in \mathbb{Z}_n^*$ выполняется: $a^{n-1} \equiv 1 \pmod{n}$ и $a^{(n-1)/q_i} \not\equiv 1 \pmod{n}$, $i = 1, 2, \dots, k$, где q_1, q_2, \dots, q_k – все простые делители $n - 1$. Тогда порядок a в группе \mathbb{Z}_n^* равняется $n - 1$. По теореме Лагранжа порядок элемента группы делит порядок группы. Следовательно, $n - 1$ делит $|\mathbb{Z}_n^*| = \varphi(n)$, что справедливо только тогда, когда n – простое. Таким образом, выполнение указанных условий доказывает простоту числа n .

Рассмотрим машину M , которая берет на вход натуральное n и сертификат

$$\text{cert}(n) = \left(a, (q_1, \text{cert}(q_1)), \dots, (q_k, \text{cert}(q_k)) \right).$$

Простоту чисел q_i демонстрируют сертификаты $\text{cert}(q_i)$, которые имеют такую же структуру, как и $\text{cert}(n)$, т. е. включают основание из $\mathbb{Z}_{q_i}^*$, простые делители числа $q_i - 1$ и сертификаты этих делителей. В сертификаты делителей могут быть вложены новые сертификаты и т. д. В целом получается целое дерево сертификатов. Сертификаты $\text{cert}(1)$, $\text{cert}(2)$ отдельно определяются как пустые слова.

Общее число вершин в дереве сертификатов для $n = 2$ и для нечетного $n \geq 3$ не превосходит $2 \log_2 n - 1$. Действительно, это верно для $n = 2$ и $n = 3$. Если $n > 3$, то $n - 1 = q_1 q_2 \dots q_k$ – составное ($k \geq 2$) и число сертификатов по индукции не больше

$$1 + \sum_{i=1}^k (2 \log_2 q_k - 1) = 1 + 2 \log_2(q_1 q_2 \dots q_k) - k < 2 \log_2 n - 1.$$

Длина каждого сертификата за вычетом длины вложенных сертификатов есть $O(\log n)$. Поэтому $|\text{cert}(n)| = O(\log^2 n)$.

Машине M обрабатывает $(n, \text{cert}(n))$ следующим образом.

1. Если $n = 1$, то возвратить 0.
2. Если $n = 2$, то возвратить 1.
3. Если n – четное, то возвратить 0.
4. Если $a^{n-1} \not\equiv 1 \pmod{n}$, то возвратить 0.
5. Если $\prod_{i=1}^k q_i \neq n - 1$, то возвратить 0.
6. Для $i = 1, \dots, k$:
 - 6.1. Если $a^{(n-1)/q_i} \equiv 1 \pmod{n}$, то возвратить 0;
 - 6.2. Если $M(q_i, \text{cert}(q_i)) = 0$, то возвратить 0.
7. Возвратить 1.

Машине M работает за полиномиальное время (проверить самостоятельно), всегда дает ответ 1 на простых n и никогда не дает ответ 1 на составных n . Мы находимся в условиях определения класса **NP**, и теорема доказана. \square

Дополнительный к $PRIMES$ язык $\overline{PRIMES} = \{n \in \mathbb{N} : n \text{ – составное}\}$ также лежит в классе **NP** – сертификатом принадлежности n языку является любой нетривиальный делитель. Анализ вычислительных задач показывает, что если и язык L , и дополнительный язык \overline{L} одновременно лежат в **NP**, то, как правило, $L \in \mathbf{P}$. Действительно, в 2002 г. индийские математики М. Агравал, Н. Каяла и Н. Саксена (AKC) разработали полиномиальный алгоритм распознавания простоты, т. е. доказали, что $PRIMES \in \mathbf{P}$.

К сожалению, алгоритм AKC является довольно медленным, даже самые эффективные его редакции работают за время $O(\log^6 n)$. В криптографии для проверки простоты в основном применяют вероятностный алгоритм Рабина – Миллера, который работает за время $O(\log^3 n)$, допускает только

односторонние ошибки (составное может быть признано простым), вероятность ошибки не превосходит $1/4$. Существование этого алгоритма даже без результатов АКС доказывает, что $PRIMES \in \text{BPP}$.

8.11. ОДНОСТОРОННИЕ ФУНКЦИИ

Криптографические системы строятся по принципу «легко для легального пользователя (Алиса), трудно для противника (Виктор)». Дело сводится к построению функций, образы которых легко определить по прообразам, но прообразы трудно восстановить по образам. Формализуем интуитивное представление о таких функциях, отождествляя Алису и Виктора с машинами Тьюринга, ресурсы которых полиномиально ограничены.

Определение 8.11. Функция $f: \{0, 1\}^* \rightarrow \{0, 1\}^*$ называется односторонней, если:

- 1) существует ПМТ, которая вычисляет f ;
- 2) для любой ПВМТ M найдется пренебрежимо малая функция ν_M такая, что

$$\mathbf{P} \left\{ f(M(1^l, f(x))) = f(x) : x \xleftarrow{R} \{0, 1\}^l \right\} \leq \nu_M(l).$$

Здесь вероятности определяются случайным выбором x и случайной лентой, которую M использует при работе.

Машине M , которая фигурирует в определении, получает на вход слово $y = f(x)$, полученное по случайному x . Машине надо обратить f , т. е. найти слово x' , которое не обязательно совпадает с x , но для которого $f(x') = f(x)$. Требуется, чтобы любая полиномиальная машина находила искомое слово лишь с пренебрежимо малой вероятностью.

Обратим внимание на то, что на вход M вместе с $f(x)$ подается длина $l = |x|$. Число l представляется не словом длины $O(\log l)$, как обычно, а унарным кодом 1^l . Это значит, что M может работать за полиномиальное от $|x|$ время, даже если слово $f(x)$ будет значительно короче слова x . При этом односторонними не будут признаваться некоторые функции, интуитивно не подходящие на эту роль. Например, функция $f_{\text{len}}(x) = |x|$. Этую функцию нельзя обратить за полиномиальное от $|f(x)|$ время (просто не хватит времени выписать ответ), хотя она легко обращается за линейное от $|x|$ время.

Существование односторонних функций до настоящего времени не доказано. Как показывает следующая теорема, вопрос существования связан с известными перешенными проблемами теории сложности.

Теорема 8.6. Условия $P \neq NP$, $NP \not\subseteq \text{BPP}$ являются необходимыми для существования односторонних функций.

Доказательство. Пусть $f: \{0, 1\}^* \rightarrow \{0, 1\}^*$ – односторонняя функция. Введем в рассмотрение язык L , составленный из троек $(1^l, y, S)$, в которых $y \in \{0, 1\}^*$ и $S \subseteq \{1, 2, \dots, l\}$. Тройка $(1^l, y, S) \in L$, если найдется слово $x \in \{0, 1\}^l$ такое, что $f(x) = y$ и $x_i = 1$ для всех $i \in S$. Слово x является сертификатом принадлежности тройки языку, проверка принадлежности выполняется за полиномиальное время, следовательно, L лежит в NP .

Предположим, что $\text{P} = \text{NP}$. Тогда $L \in \text{P}$, т. е. существует машина A , которая проверяет принадлежность $(1^l, y, S) \in L$ за полиномиальное время без сертификата. По A построим машину B , которая берет на вход пару $(1^l, y)$. В этой паре $y = f(x)$ для некоторого $x \in \{0, 1\}^l$. Машина B работает следующим образом.

1. Установить $S \leftarrow \emptyset$.
2. Для $i = 1, 2, \dots, l$:
 - 2.1. Если $A(1^l, y, S \cup \{i\}) = 1$, то $S \leftarrow S \cup \{i\}$.
3. Построить $x' \in \{0, 1\}^l$ такое, что $x'_i = 1$, только если $i \in S$.
4. Возвратить x' .

Машина B обращает f , делает это за полиномиальное время, и следовательно, f не является односторонней. Условие $\text{P} \neq \text{NP}$ действительно необходимо для односторонности f .

Аналогично доказывается необходимость условия $\text{NP} \not\subseteq \text{BPP}$. Машины A и B становятся вероятностными, A распознает L с вероятностью ошибки $\leqslant 1/3$. Машина B на шаге 2.1 вызывает A не один раз, а несколько и обрабатывает ответы A по правилу большинства. Число обращений к A выбирается так, чтобы вероятность ошибки распознавания L не превосходила $1/l$. При этом машина B обратит f с вероятностью успеха не менее $1 - (1 - 1/l)^l \geqslant 1 - e^{-1}$, которая не является пренебрежимо малой. \square

В доказательстве теоремы использована полиномиальная сводимость задачи обращения f к задаче распознавания языка $L \in \text{NP}$. При построении f можно воспользоваться сводимостью другого вида: задачи распознавания языка L' к задаче обращения f . Можно строить f так, чтобы язык L' было трудно распознать, например, чтобы $L' \in \text{NPC}$. Можно ожидать при этом, что f будет трудно обратить. Оказывается, что это не так: сложность обращения f оценивается в среднем, в то время как сложность распознавания языков характеризуется наихудшим случаем (максимальная сложность на входах определенной длины). Поэтому обращение f является трудным в наихудшем случае (что подтверждается сведением), но может быть простым в среднем (что не противоречит сведению).

Пример 8.4. Пусть

$$f(x_1, \dots, x_n, S) = \left(x_1, \dots, x_n, \sum_{i \in S} x_i \right), \quad x_i \in \mathbb{Z}, \quad S \subseteq \{1, 2, \dots, n\}.$$

Задача распознавания NP-полного языка SubsetSum сводится к задаче обращения f . Это, однако, не означает, что f является односторонней: почти для всех входов f множество S может быть найдено по выходу за полиномиальное время (с помощью LLL-алгоритма или его модификаций). На этом наблюдении базируются атаки на *рюкзачные криптосистемы* – криптосистемы, основанные на различных уточнениях задачи SubsetSum .

8.12. ФУНКЦИИ С ЛАЗЕЙКОЙ

Известна аналогия между односторонней функцией и телефонным справочником: по фамилии x легко определить номер телефона $f(x)$, однако определение фамилии по номеру – трудная задача. Функции с лазейкой, которые мы сейчас введем, являются хитро устроенным спарвочниками – обладая специальным секретом (лазейкой), поиск в справочнике нужного номера можно выполнить очень быстро.

Определение 8.12. Функция $f: \{0, 1\}^* \rightarrow \{0, 1\}^*$ называется функцией с лазейкой, если:

- 1) f – односторонняя функция;
- 2) существует ПВМТ I и слова $t_1, t_2, \dots \in \{0, 1\}^*$, $|t_l| = l^{O(1)}$, такие, что $f(x) = f(x')$ для всех $x \in \{0, 1\}^l$ и соответствующих $x' = I(1^l, f(x), t_l)$.

Функции с лазейкой являются односторонними и соответствуют правилу «легко вычислить, трудно обратить». Дополнительно поддерживается правило «легко обратить с лазейкой». Имеется в виду, что кроме Алисы есть еще один легальный пользователь – Боб, который отождествляется с машиной I . Бобу известна лазейка t_l , с помощью которой он находит x' по $(1^l, f(x))$.

В криптографии функции с лазейкой могут использоваться для построения систем ЭЦП и криптосистем с открытым ключом. Эти системы будут подробно рассмотрены в соответствующих главах книги. Здесь мы остановимся на принципах построения. Сразу оговоримся, что упомянутые криптографические системы не обязательно должны строиться на основе функций с лазейкой (хотя на практике функции с лазейкой явно или неявно используются при построении).

В общем случае применяется не фиксированная функция с лазейкой, а семейство таких функций. Функции семейства, как правило, являются частичными. Функции семейства имеют конечные области определений. С помощью вероятностного алгоритма Gen , который работает за полиномиальное в среднем время, Боб выбирает одну из таких функций. Алгоритм Gen берет на вход слово 1^l и возвращает описание f и лазейку t , нужную для обращения f . Описание f включает спецификацию области определения D и области значений E . Входной параметр l определяет размерности D и E .

Описание f называется *открытым ключом*, лазейка t – *личным*. Открытый ключ Боб делает общедоступным, личный ключ хранит в секрете.

В криптосистемах с открытым ключом функция f биективна. Алисе требуется передать Бобу конфиденциальное сообщение $x \in D$ – *открытый текст*. Алиса использует открытый ключ Боба и за полиномиальное время находит $y = f(x)$ – *шифртекст*. Боб за полиномиальное время определяет x по (y, t) . Виктор не знает личный ключ t , ему требуется определить x по y , что является трудной задачей обращения f .

В системах ЭЦП Боб подтверждает подлинность документа $y \in E$. Зная t , Боб за полиномиальное время находит $x \in D$ такое, что $f(x) = y$. Слово x называется *электронной цифровой подписью* y . Алиса проверяет подпись, сравнивая $f(x)$ с y . Для определения подписи x без личного ключа t Виктору снова требуется обратить f .

Детали могут отличаться. Например, Алиса может зашифровывать открытый текст, дополненный случайными данными, или Боб может подписывать не сам документ, а его хеш-значение. Отличия важны, но не принципиальны. Суть использования функций с лазейкой сохраняется.

8.13. ФУНКЦИЯ РАБИНА

Функции, которые доказательно являются функциями с лазейкой, пока не известны. Тем не менее построены функции, которые гипотетически (по мнению большинства специалистов) являются таковыми. Рассмотрим одну из них – *функцию Рабина*.

Функция Рабина f (точнее, представитель семейства функций Рабина) описывается натуральным $n = p q$, где p и q – различные простые, сравнимые с 3 по модулю 4. Такое n называется *числом Блюма*. Будем считать, что битовые длины p и q примерно равны, а битовая длина n равняется l : $\log_2 p \approx \log_2 q$, $\lceil \log_2 n \rceil = l$. Длина l выбирается Бобом при вызове алгоритма *Gen*. Этот алгоритм находит два случайных подходящих простых (сделать это можно за полиномиальное в среднем время), а затем их произведение.

Область определения $D = \mathbb{Z}_n$, область значений E – множество квадратов в \mathbb{Z}_n , лазейкой является пара (p, q) . Действие f :

$$f(x) = x^2 \bmod n.$$

Рассмотрим задачи, которые решают Алиса, Боб и Виктор при использовании f как функции с лазейкой.

1. Алисе требуется вычислить $y = f(x)$. Сделать это можно за время $O(l^2)$.
2. Бобу по (y, p, q) требуется найти $x' \in \sqrt{y} \bmod n$. Модуль n выбран так, чтобы упростить извлечение корней по модулям его делителей:

$$x_1 = y^{(p+1)/4} \bmod p \in \sqrt{y} \bmod p, \quad x_2 = y^{(q+1)/4} \bmod q \in \sqrt{y} \bmod q.$$

Боб находит x_1, x_2 , а затем решает китайскую систему

$$x' \equiv \pm x_1 \pmod{p}, \quad x' \equiv \pm x_2 \pmod{q}$$

с произвольной расстановкой знаков \pm . Искомый корень Боб найдет за время $O(l^3)$.

3. Виктору также требуется найти $x' \in \sqrt{y} \pmod{n}$, но уже по (n, y) . Пусть $d = \text{НОД}(y, n)$. Если $d > 1$, то $d \in \{p, q\}$, и Виктор узнает личный ключ Боба. После этого Виктор может определить x' , действуя как Боб. Но при $x \xleftarrow{R} \mathbb{Z}_n$ вероятность

$$\mathbf{P}\{d > 1\} = \frac{n - \varphi(n)}{n} = \frac{1}{p} + \frac{1}{q} - \frac{1}{pq}$$

пренебрежимо мала. Если же $d = 1$, то Виктору требуется решить задачу **Sqrt**, вероятно полиномиально эквивалентную **Factor** (см. следствие 8.1). Но **Factor** – задача, признаваемая трудной, следовательно, задача **Sqrt** также трудна, и x' снова определяется лишь с пренебрежимо малой вероятностью.

Если f предполагается использовать для шифрования, то область определения D сужают так, чтобы получить биекцию. Например, зашифровывают слова x со специальным фиксированным префиксом, а после расшифрования используют тот из корней, который удовлетворяет выбранному формату.

8.14. ЗАДАНИЯ

1. Язык $L = \{0, 1\}^*1$ состоит из непустых слов, которые заканчиваются единицей. Описать таким же образом следующие языки:

- 1) $\{0, 1\}^*1\{0, 1\}^{10}$;
- 2) $\{0, 1\}^*1\{0, 1\}^*1\{0, 1\}^*$;
- 3) 0^*0100^* .

2. Доказать, что множество $\{0, 1\}^*$ счетное. Доказать, что множество функций $\{0, 1\}^* \rightarrow \{0, 1\}$ – континuum.

3. Подтвердите оценки таблицы (первые 4 строки).

4. Пусть $n_2(p)$ – минимальный квадратичный невычет по нечетному простому модулю p . Доказать, что $n_2(p)$ будет простым числом.

5. Пусть ξ_1, \dots, ξ_k – независимые одинаково распределенные бернуlliевские случайные величины, $\mathbf{P}\{\xi_1 = 1\} = \varepsilon < 1/2$. Пусть $S = \xi_1 + \dots + \xi_k$. Тогда справедливо неравенство Чебышева:

$$\mathbf{P}\{S \geqslant (1 + \delta)\mu\} \leqslant \left(\frac{\varepsilon^\delta}{(1 + \delta)^{1+\delta}}\right)^\mu, \quad \mu = \mathbf{E}S = k\varepsilon.$$

Используя это неравенство, доказать, что

$$\mathbf{P} \{S \geq k/2\} \leq \exp \left(-k \frac{(1/2 - \varepsilon)^2}{(1/2 + \varepsilon)} \right)$$

(подсказка: воспользоватьсяся оценкой $\ln(1 + \delta) > \frac{2\delta}{2 + \delta}$, справедливой для $\delta > 0$).

6. Доказать полиномиальную сводимость задачи распознавания L_{Factor} к задаче **Factor**.

7. Доказать полиномиальную сводимость задачи факторизации чисел Блюма к задаче вычисления значений функции Эйлера.

8. Пусть f – предикат с языком

$$L = \{(p, g) : p \text{ – простое, } g \text{ – первообразный корень } \bmod p\}.$$

Доказать, что $f \leq_P \text{Factor}$.

9. Доказать, что $\mathbf{P} \subseteq \mathbf{NP}$, $\mathbf{P} \subseteq \mathbf{BPP}$.

10. Пусть $L = \{(p, g) : p \text{ – простое, } g \text{ – примитивный элемент } \mathbb{F}_p^*\}$. Доказать, что L и дополнительный язык \bar{L} лежат в \mathbf{NP} .

11. Пусть $L = \{n : n \text{ свободно от квадратов}\}$. Доказать, что $L, \bar{L} \in \mathbf{NP}$.

12. Пусть $L = \{(p, a) : p \text{ – простое, } a \text{ – квадратичный вычет } \bmod p\}$. Доказать, что $L, \bar{L} \in \mathbf{NP}$.

13. Пусть p – нечетное простое. Доказать, что корень $b \in \sqrt{a} \bmod p^e$ можно определить по корню $c \in \sqrt{a} \bmod p$ следующим образом (формула Тонелли):

$$b = c^{p^{e-1}} a^{(p^{e-2}p^{e-1}+1)/2} \bmod p^e.$$

14. Обратить функцию Рабина: найти все решения сравнения $x^2 \equiv y \pmod n$ при $n = 19 \cdot 11$ и $y = 100$.

КОММЕНТАРИИ

Развернутую информацию по различным (в том числе криптографическим) аспектам теории сложности можно найти в книгах [63, 100, 101]. Использованная в главе терминология в целом соответствует русскоязычным книгам [21, 27]. На сайте <http://www.cryptography.ru> размещен справочник по математической криптографии, в котором представлена подробная информация по вопросам, затронутым в последних пунктах.

Гамма-код, а также некоторые другие способы кодирования натуральных чисел введены П. Элиасом в [93].

Базовые теоретико-числовые алгоритмы описаны в гл. 9, а также в книгах [23, 64, 84, 133].

Характеристика $s_M(x)$, которая используется в определении 8.2, учитывает как число вспомогательных ячеек памяти, так и длину входа и выхода. Чтобы учесть в $s_M(x)$ только вспомогательные ячейки (например, чтобы выделить класс машин, которые работают на логарифмической вспомогательной памяти), вводят машины с несколькими лентами. Одна лента используется для чтения, на нее записываются входные данные. Еще одна лента используется для записи выходных данных. Остальные ленты называются рабочими, именно они поддерживают вспомогательную память. На каждой ленте имеется свое управляющее устройство. Функция переходов усложняется – теперь она описывает управление не одной, а несколькими лентами. Характеристика $s_M(x)$ определяется как суммарное число ячеек рабочих лент, использованное при вычислениях на входе x .

Некоторые специалисты считают субэкспоненциальным не время $2^{o(l)}$, а время $2^{l^{o(1)}}$. При этом субэкспоненциальными не признаются многие алгоритмы (например, алгоритмы решета числового поля с временем работы $2^{(c+o(1))l^{1/3}(\log l)^{2/3}}$), которые другие специалисты классифицируют все-таки как субэкспоненциальные.

Имеются технические трудности использования понятия «полиномиальное в среднем время», если оно вводится в соответствии с определением 8.3. Можно подобрать пример, когда $\mathbf{Et}_M(x)$ полиномиально увеличивается с ростом $|x|$, но $\mathbf{Et}_M^2(x)$ растет уже с суперполиномиальной скоростью. Такая ситуация неприемлема при организации сведений между задачами. Поэтому полиномиальное в среднем время определяют по-другому: требуют, чтобы существовали положительные константы C и ϵ , для которых

$$\mathbf{E} \frac{t_M(x)^\epsilon}{l} \leq C$$

при $x \xleftarrow{R} \{0, 1\}^l$. Новое определение позволяет применить неравенство Маркова и получить оценку:

$$\mathbf{P} \left\{ t_M(x) \geq (Cd)^{1/\epsilon} \right\} = \mathbf{P} \left\{ \frac{t_M(x)^\epsilon}{l} \geq Cd \right\} \leq 1/d.$$

Оценка означает, что машина M с высокой вероятностью работает за полиномиальное время.

Терминология, касающаяся алгоритмов Лас-Вегас, разнится. В некоторых источниках алгоритмам Лас-Вегас разрешается возвращать символ \perp (ответ не найден). При этом алгоритмы, которые все-таки всегда возвращают правильный ответ, принято называть алгоритмами Шервуд.

Упомянутые результаты С. Кука и Л. А. Левина представлены в работах [28, 85]. Кук рассматривал задачи распознавания, а Левин – задачи поиска.

Р. Карп опубликовал знаменитый список, в который вошла 21 задача из класса **NPC** [114]. Интересно, что Кук обратил внимание на затруднения при классификации $PRIMES \stackrel{?}{\in} \textbf{NPC}$, а Карп – на затруднения при классификации $\overline{PRIMES} \stackrel{?}{\in} \textbf{NPC}$. Теорема 8.5 о классификации $PRIMES$ доказана в [144].

Различные классы сложности подробно описаны и классифицированы в [112]. Интернет-каталог классов сложности поддерживает С. Аронзон (см. https://complexityzoo.uwaterloo.ca/Complexity_Zoo).

Функции, соответствующие определению 8.11, иногда называют сильно односторонними. Выделяют еще слабо односторонние функции – условие « M обратит f с пренебрежимо малой вероятностью» меняется на « M не обратит f с обратно полиномиальной вероятностью». Точнее, второе условие в определении 8.11 принимает следующий вид:

2') найдется $c \in \mathbb{N}$ такое, что для любой ПВМТ M

$$\mathsf{P} \left\{ f(M(1^l, f(x))) \neq f(x) : x \xleftarrow{R} \{0, 1\}^l \right\} \geq l^{-c}$$

при всех достаточно больших $l \in \mathbb{N}$.

Доказано, что если существуют слабо односторонние функции, то существуют и сильно односторонние [101, предложение 2.3.1].

Определение 8.12 соответствует [102]. Семейством функций с лазейкой называется множество функций $f: D_f \rightarrow E_f$, где $D_f, E_f \subseteq \{0, 1\}^*$ – конечные множества. Семейство должно удовлетворять следующим ограничениям:

1) существует ПВМТ Gen , которая берет на вход слово 1^l и возвращает слова $\text{descr}(f)$ и t , длины которых полиномиально ограничены (в зависимости от l);

2) существует ПМТ, которая берет на вход $\text{descr}(f)$ и $x \in D_f$ и возвращает $f(x)$;

3) существует ПВМТ, которая берет на вход $\text{descr}(f)$, t и $y \in E_f$ и возвращает $x' \in D_f$ такое, что $f(x') = y$;

4) существует ПВМТ, которая берет на вход $\text{descr}(f)$ и возвращает реализацию случайной величины с равномерным распределением на D_f ;

5) для любой ПВМТ M найдется пренебрежимо малая функция ν_M такая, что

$$\mathsf{P} \left\{ f(M(1^l, \text{descr}(f), f(x))) = f(x) : (\text{descr}(f), t) \leftarrow \mathsf{Gen}(1^l), x \xleftarrow{R} D_f \right\} \leq \nu_M(l).$$

Функция Рабина введена в [148] как альтернатива функции RSA. М. Рабин предложил использовать новую функцию для построения систем ЭЦП. Криптосистема с открытым ключом на основе функции Рабина не противостоит атаке при выбираемом шифртексте.

Г л а в а 9

БАЗОВЫЕ АЛГОРИТМЫ

9.1. АЛГОРИТМЫ АРИФМЕТИКИ БОЛЬШИХ ЧИСЕЛ

Задавшись натуральным *основанием* $b \geq 2$, неотрицательному целому числу u поставим в соответствие набор $(u_{k-1} \dots u_1 u_0)_b$ целых чисел такой, что

$$u = u_{k-1}b^{k-1} + \dots + u_1b + u_0, \quad 0 \leq u_0, u_1, \dots, u_{k-1} < b.$$

Если $u_{k-1} \neq 0$, то говорят, что u является *k-разрядным числом по основанию* b или *k-битовым числом*, если $b = 2$. Число u_0 называют *младшим разрядом* u , а число u_{k-1} – *старшим разрядом*. Разряды записи по основанию 16 принято обозначать цифровыми и буквенными символами (табл. 9.1).

Таблица 9.1

Разряды шестнадцатеричной записи

$0 = (0000)_2 = (0)_{16}$	$8 = (1000)_2 = (8)_{16}$
$1 = (0001)_2 = (1)_{16}$	$9 = (1001)_2 = (9)_{16}$
$2 = (0010)_2 = (2)_{16}$	$10 = (1010)_2 = (\text{A})_{16}$
$3 = (0011)_2 = (3)_{16}$	$11 = (1011)_2 = (\text{B})_{16}$
$4 = (0100)_2 = (4)_{16}$	$12 = (1100)_2 = (\text{C})_{16}$
$5 = (0101)_2 = (5)_{16}$	$13 = (1101)_2 = (\text{D})_{16}$
$6 = (0110)_2 = (6)_{16}$	$14 = (1110)_2 = (\text{E})_{16}$
$7 = (0111)_2 = (7)_{16}$	$15 = (1111)_2 = (\text{F})_{16}$

Приведем алгоритмы арифметических операций над числами с произвольным количеством разрядов (*большими числами*). При практической реализации таких алгоритмов основание b выбирается равным размеру машинного слова (как правило, это степень 2). Предполагается, что архитектура ЭВМ дает возможность выполнять элементарные арифметические операции над одно- и двухразрядными числами.

Сложение $u + v = (w_k \dots w_1 w_0)_b$ чисел $u = (u_{k-1} \dots u_1 u_0)_b$ и $v = (v_{k-1} \dots v_1 v_0)_b$.

1. Установить $c \leftarrow 0$.
2. Для $i = 0, \dots, k - 1$ выполнить:
 - а) $w_i \leftarrow (u_i + v_i + c) \bmod b$;
 - б) положить $c \leftarrow 0$, если $u_i + v_i + c < b$, и $c \leftarrow 1$ в противном случае.

3. Установить $w_k \leftarrow c$.

Вычитание $u - v = (w_{k-1} \cdots w_1 w_0)_b$ числа $v = (v_{k-1} \cdots v_1 v_0)_b$ из числа $u = (u_{k-1} \cdots u_1 u_0)_b$, $u \geq v$.

1. Установить $c \leftarrow 0$.

2. Для $i = 0, \dots, k-1$ выполнить:

а) $w_i \leftarrow (u_i - v_i - c) \bmod b$;

б) положить $c \leftarrow 0$, если $u_i - v_i + c \geq 0$, и $c \leftarrow 1$ в противном случае.

Если $u < v$, то по окончании выполнения алгоритма $c = 1$ и результат $w = u - v + b^k$.

Умножение $uv = (w_{k+l-1} \cdots w_1 w_0)_b$ чисел $u = (u_{k-1} \cdots u_1 u_0)_b$ и $v = (v_{l-1} \cdots v_1 v_0)_b$.

1. Для $i = 0, \dots, k+l-1$ установить $w_i \leftarrow 0$.

2. Для $i = 0, \dots, l-1$ выполнить:

а) $c \leftarrow 0$;

б) для $j = 0, \dots, k-1$ вычислить $(xy)_b \leftarrow w_{i+j} + u_j v_i + c$ и установить $w_{i+j} \leftarrow y$, $c \leftarrow x$;

в) $w_{i+k} \leftarrow c$.

Деление числа $u = (u_{k+l-1} \cdots u_1 u_0)_b$ на число $v = (v_{k-1} \cdots v_1 v_0)_b$, $k \geq 2$, $v_{k-1} \neq 0$, т. е. нахождение частного $q = (q_l \cdots q_1 q_0)_b$ и остатка $r = (r_{k-1} \cdots r_1 r_0)_b$ таких, что $u = qv + r$ и $0 \leq r < v$.

1. Выбрать произвольное целое d , для которого $vd < b^k$ и $\left\lfloor \frac{b}{2} \right\rfloor \leq v_{k-1}d < b$.

Установить:

а) $u \leftarrow ud$, $u = (u_{k+l} \cdots u_1 u_0)_b$;

б) $v \leftarrow vd$, $v = (v_{k-1} \cdots v_1 v_0)_b$.

2. Для $i = k+l, k+l-1, \dots, k$:

а) вычислить пробное частное

$$\hat{q} \leftarrow \min \left(\left\lfloor \frac{u_i b + u_{i-1}}{v_{k-1}} \right\rfloor, b-1 \right);$$

б) пока $\hat{q}(v_{k-1}b + v_{k-2}) > u_i b^2 + u_{i-1}b + u_{i-2}$, выполнить $\hat{q} \leftarrow \hat{q} - 1$;

в) $u \leftarrow u - \hat{q}vb^{i-k}$;

г) если $u < 0$, то установить: $\hat{q} \leftarrow \hat{q} - 1$, $u \leftarrow u + vb^{i-k}$;

д) $q_{i-k} \leftarrow \hat{q}$.

3. Установить $r \leftarrow u/d$.

Действия на шаге 1 алгоритма называются *нормализацией*. При нормализации цикл 2, б выполняется не более двух раз. Корректирующее сложение на шаге 2, г выполняется с вероятностью $\approx \frac{2}{b}$. Если основание b является степенью 2, то нормализация может состоять в сдвиге битов u и v влево на такое минимальное количество позиций, что старший бит v_{k-1} равен 1. В общем случае можно выбрать $d = \lfloor b/(v_{k-1} + 1) \rfloor$.

9.2. ОПЕРАЦИЯ МОНТГОМЕРИ И РЕДУКЦИЯ БАРРЕТА

Располагая алгоритмами арифметики больших чисел, можно организовать вычисления и в кольце \mathbb{Z}_n . Для приведения по модулю результата аддитивной операции достаточно прибавить или вычесть n , а при умножении следует найти остаток от деления результата на n . Кроме того, известны методы приведения по модулю, использующие вычислительно трудоемкое деление только на стадии предварительных вычислений. Опишем два таких метода.

Метод Барретта [66]. Пусть модуль n является k -разрядным числом по основанию b , $x = (x_{2k-1} \dots x_1 x_0)_b$. Для вычисления $r = x \bmod n$ предварительно находят число $m = \lfloor b^{2k}/n \rfloor$. Если $x = qn + r$, то

$$q = \left\lfloor \frac{x}{b^{k-1}} \times \frac{b^{2k}}{n} \times \frac{1}{b^{k+1}} \right\rfloor \approx \hat{q} = \left\lfloor \left\lfloor \frac{x}{b^{k-1}} \right\rfloor \frac{m}{b^{k+1}} \right\rfloor.$$

Точнее, известно, что $q - 2 \leq \hat{q} \leq q$. Поэтому число $\hat{r} = x - \hat{q}n$ совпадает либо с r , либо с $r + n$, либо с $r + 2n$. Таким образом, для вычисления r требуется использовать два умножения ($\lfloor x/b^{k-1} \rfloor m$, $\hat{q}n$) и не более трех вычитаний ($x - \hat{q}n$, $\hat{r} - n$, $(\hat{r} - n) - n$) больших чисел. Деление на b^{k+1} состоит в сдвиге разрядов делимого на $k \pm 1$ позиций вправо.

Метод Монтгомери [135]. Пусть R – натуральное число, взаимно простое с модулем n , $n' = -n^{-1} \bmod R$, $x \in \{0, 1, \dots, nR - 1\}$. Вычисляется $xR^{-1} \bmod n$. Если $y = xn' \bmod R$, то:

- 1) $x + yn \equiv x(1 + nn') \equiv 0 \pmod{R}$ и $(x + yn)/R$ – целое число;
- 2) $(x + yn)/R \equiv xR^{-1} \pmod{n}$;
- 3) $x + yn < 2nR$.

Таким образом, $(x + yn)/R = xR^{-1} \bmod n$ или $(x + yn)/R = xR^{-1} \bmod n + n$. Если используется представление чисел по основанию b , а R является степенью b , то для вычисления $xR^{-1} \bmod n$ достаточно выполнить два умножения xn' и yn , одно сложение $x + yn$ и, возможно, одно вычитание $(x + yn)/R - n$. Деление на R состоит в сдвиге разрядов делимого вправо.

Полагаем далее, что $n = (n_{k-1} \dots n_1 n_0)_b$, $(n, b) = 1$. Определим $R = b^k$, $n' = -n^{-1} \bmod b$ и приведем алгоритмы умножения и возведения в степень, основанные на методе Монтгомери.

Умножение по Монтгомери – вычисление $w = uvR^{-1} \bmod n$ для $u = (u_{k-1} \dots u_1 u_0)_b$, $v = (v_{k-1} \dots v_1 v_0)_b$, $0 \leq u, v < n$.

1. Установить $w \leftarrow 0$. Считаем, что $w = (w_k \dots w_1 w_0)_b$.
2. Для $i = 0, 1, \dots, k - 1$ выполнить:
 - a) $y_i \leftarrow (w_0 + u_i v_0)n' \bmod b$;
 - б) $w \leftarrow (w + u_i v + y_i n)/b$.

3. Если $w \geq n$, то $w \leftarrow w - n$.

При выполнении шага 2, б $w_0 + u_i v_0 + y_i n \equiv 0 \pmod{b}$, т. е. число $w + u_i v + y_i n$ делится нацело на b . Если перед выполнением шага $w < 2n - 1$, то и после выполнения

$$w \leq \frac{1}{b}(2n - 2 + (b - 1)(n - 1) + (b - 1)n) = 2n - 1 - \frac{1}{b} < 2n - 1.$$

Таким образом, окончательный результат вычислений w всегда меньше n . Кроме того,

$$w \equiv v(u_0 + u_1 b + \dots + u_{k-1} b^{k-1}) b^{-k} \equiv uvR^{-1} \pmod{n}.$$

Возведение в степень по Монтгомери – определение $w = u^e \pmod{n}$ для $e = (e_{l-1} \dots e_1 e_0)_2$, $e_{l-1} = 1$, $1 \leq u < n$.

1. Установить $w \leftarrow R \pmod{n}$, $\tilde{u} \leftarrow \text{Mont}(u, R^2 \pmod{n})$.
2. Для $i = l - 1, l - 2, \dots, 0$ выполнить:
 - а) $w \leftarrow \text{Mont}(w, w)$;
 - б) если $e_i = 1$, то $w \leftarrow \text{Mont}(w, \tilde{u})$.
3. $w \leftarrow \text{Mont}(w, 1)$.

Здесь $\text{Mont}(u, v)$ есть результат $uvR^{-1} \pmod{n}$ умножения по Монтгомери чисел u и v .

9.3. ВЕРОЯТНОСТНЫЕ И ДЕТЕРМИНИРОВАННЫЕ АЛГОРИТМЫ ТЕСТИРОВАНИЯ НА ПРОСТОТУ

Для криптографии с открытым ключом важно как построение тестов, позволяющих установить *простоту данного большого числа*, так и построение *методов разложения (факторизации)* большого числа на простые множители. Рассмотрим лишь простейшие методы.

Теорема 9.1 (теорема Вильсона). *Натуральное n тогда и только тогда является простым, когда $(n + 1)! + 1 \equiv 0 \pmod{n}$.*

Доказательство. По теореме Ферма сравнение $x^{p-1} - 1 \equiv 0 \pmod{p}$ имеет $p - 1$ решений $1, \dots, p - 1$. Поэтому

$$x^{p-1} - 1 = (x - 1)(x - 2) \cdots (x - p + 1) + pF(x).$$

Положим $x = 0$, тогда $(p + 1)! \equiv -1 \pmod{p}$. Если же n – составное, то оно содержит простой множитель $q < n$, который является делителем $(n - 1)!$, так что $(n - 1)! + 1$ не делится на q , а значит, и на n . \square

В качестве следствия отсюда получается теорема.

Теорема 9.2. Числа n и $n+2$ тогда и только тогда являются простыми числами-близнецами, когда

$$4((n-1)! + 1) + n \equiv 0 \pmod{n(n+2)}, \quad n > 1.$$

Доказательство. Если условие выполнено, то, поскольку $n \neq 2, 4$, должно быть $(n+1)! + 1 \equiv 0 \pmod{n}$. Остается применить теорему Вильсона, чтобы убедиться в простоте n . Далее $n \equiv -2 \pmod{n+2}$. Поэтому $(n+1) \equiv \equiv 2(n-1) \pmod{n+2}$. Отсюда следует:

$$\begin{aligned} 0 &\equiv 4((n-1)! + 1) + n \equiv 2(n+1)! + 2 \equiv \\ &\equiv 2((n+1)! + n) \pmod{n+2}. \end{aligned} \tag{9.1}$$

Поэтому простым является и число $n+2$.

Пусть, наоборот, n и $n+2$ – простые числа. Тогда $4((n+1)! + 1) + n$ по теореме Вильсона делится на n , а вследствие (9.1) – и на $n+2$, так что оно делится и на $n(n+2)$. \square

Еще один критерий простоты можно сформулировать в виде следующей теоремы.

Теорема 9.3. Если существует такое число a , взаимно простое с n , что $a^{n-1} \equiv 1 \pmod{p}$, но $a^{(n-1)/p} \not\equiv 1 \pmod{n}$ для любого простого делителя p числа $n-1$, то n – простое число.

Доказательство. Поскольку каждый нетривиальный делитель t числа $n-1$ входит в один из делителей $(n-1)/p$, то из предположения теоремы следует, что $a^t \not\equiv 1 \pmod{n}$, так как в противном случае существовало бы p , для которого было бы $a^{(n-1)/p} \equiv 1 \pmod{n}$. Если a принадлежит показателю d , то $a^d \equiv 1 \pmod{n}$ и d должно быть делителем $n-1$, что возможно лишь для $d = n-1$. Таким образом, $n-1$ является делителем $\varphi(n)$. Для составных n $\varphi(n) < n-1$. Поэтому n должно быть простым числом. Если, наоборот, $n = p$, то первообразный корень удовлетворяет условиям теоремы. \square

Существует два подхода к реализации тестирования простоты числа – вероятностный и детерминированный. Результатом работы детерминированного теста является доказуемое высказывание о том, что тестируемое число простое (составное). Для вероятностного теста на простоту характерно то, что, как правило, со 100 %-ной гарантией можно определить, является ли число составным, и только с вероятностью, близкой к 1 (но не равной 1), можно определить случай простого числа.

В основе большинства тестов на простоту лежит малая теорема Ферма. Она дает необходимый признак простоты числа p . Отсюда вытекает алгоритм тестирования числа n на простоту. Очевидно, что если при некотором a таком, что n/a , имеет место $a^{n-1} \not\equiv 1 \pmod{n}$, то число n – составное.

Однако для доказательства простоты необходим какой-нибудь достаточный признак простоты числа. Оказалось, что теорема, обратная малой теореме Ферма, не верна, а именно, существуют составные числа n такие, что для любых $(a, n) = 1$ имеет место сравнение

$$a^{n-1} \equiv 1 \pmod{n}.$$

Такие числа называются *числами Кармайкла* и имеют вид $p_1 \dots p_k$, где все нечетные простые p_i различны, причем $(p_i - 1) \mid (n - 1)$. Наименьшее число Кармайкла равно $561 = 3 \cdot 7 \cdot 11$ (табл. 9.2). Однако такие числа встречаются редко и малую теорему Ферма можно применять для построения простых чисел.

Таблица 9.2

Числа Кармайкла

561	=	$3 \times 11 \times 17$
1105	=	$5 \times 13 \times 17$
1729	=	$7 \times 13 \times 31$
2465	=	$5 \times 17 \times 29$
2821	=	$7 \times 13 \times 31$
6601	=	$7 \times 23 \times 41$
8911	=	$7 \times 19 \times 67$
41041	=	$7 \times 11 \times 13 \times 41$
825265	=	$5 \times 7 \times 17 \times 19 \times 73$
413631505	=	$5 \times 7 \times 17 \times 73 \times 89 \times 107$

Пусть n – нечетное составное число и a – целое такое, что $(a, n) = 1$. Число n называется *псевдопростым по основанию* a , если

$$a^{n-1} \equiv 1 \pmod{n}.$$

Оказывается, что если составное n – число Кармайкла, то существует не менее $n/2$ оснований a , для которых n – не псевдопростое. Таким образом, если заранее знать, что n не является числом Кармайкла, то тестировать n на простоту можно, выбрав случайное основание a и проверив псевдопростоту n по основанию a . Если условие не выполнено, то согласно малой теореме Ферма n – составное. Допустим, что выдерживается k проверок, тогда вероятность того, что n – составное, равна $1/2^k$. Выбирая k достаточно большим, можно находить такие числа n , что вероятность того, что n – составное, сколь угодно мала. Очевидный недостаток этого теста – необходимость заранее знать, что n – не число Кармайкла. От этого недостатка можно избавиться, используя теорему Эйлера (см. гл. 2, теоремы 2.9, 2.13).

Теперь можем описать следующий вероятностный тест на простоту.

9.3.1. Тест Соловая – Штрассена

1. Выбрать k различных оснований a таких, что $1 < a < n$.

2. Для каждого a проверить условие (2.13), и если оно не выполнено хотя бы для одного a , то n – составное. Если же для всех a (2.13) выполнено, то n является составным с вероятностью, не превышающей $1/2^k$.

Дальнейшее усовершенствование данной методики приводит к следующему широко используемому вероятностному тесту простоты.

Нечетное составное число $n = 2^s t + 1$ (t – нечетное) называется *сильно псевдопростым по основанию a* , если выполняется одно из условий:

1) $a^t \equiv 1 \pmod{n}$;

2) существует $0 \leq r < s$ такое, что $a^{2^r t} \equiv -1 \pmod{n}$.

Если число n является простым, то для него одно из этих условий всегда выполняется. Действительно, в этом случае $a^{n-1} \equiv 1 \pmod{n}$. Тогда в силу простоты n $a^{(n-1)/2} \pmod{n}$ равно 1 или -1 . Если получено -1 , то, очевидно, n – сильно псевдопростое. В случае $a^{(n-1)/2} \equiv 1 \pmod{n}$ рассматриваем число $a^{(n-1)/4} \pmod{n}$, которое в силу простоты n также равно 1 или -1 . Проделав эту процедуру не более s раз, получим требуемое утверждение.

9.3.2. Тест Миллера – Рабина

Предположим, необходимо определить, простым или составным является нечетное число n . Запишем n в виде $n = 2^s + t$, где t – нечетное, и выберем случайное целое a в диапазоне $1 < a < n$. Вычислим $a^t \pmod{n}$. Если получим 1 или -1 , то заключаем, что n проходит тест для данного значения a , и переходим к выбору следующего a . В противном случае вычисляем $(a^t)^2 \pmod{n}$ и сравниваем с -1 . Если это условие выполняется, то n проходит тест и переходим к новому a . В противном случае возводим полученное выражение в квадрат и т. д. Если, дойдя до $a^{(n-1)/2} \pmod{n}$, не получили -1 , то n – составное. Если тест прошел для k различных значений a , то по теореме 9.4 с вероятностью не более $1/4^k$ число n – составное.

Теорема 9.4. *Если n – нечетное составное число, то оно является сильно псевдопростым по основанию a для не более чем $n/4$ оснований a , $0 < a < n$.*

Для доказательства теоремы потребуются следующие две леммы.

Лемма 9.1. *Пусть $d = (k, m)$. Тогда существует в точности d элементов в группе $\{g, g^2, \dots, g^m = 1\}$, удовлетворяющих уравнению $x^k = 1$.*

Лемма 9.2. *Пусть p – нечетное простое число, $p = 2^{s'} t' + 1$, где t' – нечетное. Тогда количество $x \in (\mathbb{Z}/p\mathbb{Z})^*$, удовлетворяющих $x^{2^r t} \equiv -1 \pmod{p}$ (t – нечетное), равно 0 в случае $r \geq s'$ и равно $d2^r$, если $r < s'$, здесь $d = (t, t')$.*

Доказательство. Пусть g – порождающий элемент группы $(\mathbb{Z}/p\mathbb{Z})^*$ и $x = g^j$, где $0 \leq j < p - 1$. Поскольку $g^{(p-1)/2} \equiv -1 \pmod{p}$ и $p - 1 = 2^{s'}t'$, то сравнение в условии леммы эквивалентно сравнению

$$2^r t j \equiv 2^{s'-1} t' \pmod{2^{s'} t'}, \quad (9.2)$$

где j – неизвестное. Если $r > s' - 1$, то, очевидно, данное сравнение не имеет решений. Если $r \leq s' - 1$, то, разделив модуль и обе части (9.2) на $2^r d$, получим

$$\frac{t}{d} j \equiv 2^{s'-r-1} \frac{t'}{d} \pmod{2^{s'-r} \frac{t'}{d}}. \quad (9.3)$$

Поскольку $(t/d, 2^{s'-r} t'/d) = 1$, то существует единственное решение j_0 сравнения (9.3). Тогда $j_0 k$, где $k = 1, 2, \dots, 2^r d$, являются решениями сравнения из условия леммы. \square

Доказательство. Рассмотрим три возможных случая для доказательства теоремы 9.4.

Случай 1. Предположим, что n делится на квадрат некоторого простого p , т. е. $p^\alpha \mid n$, $\alpha \geq 2$. Покажем, что в этом случае n не может быть псевдопростым (тем более сильно псевдопростым) для более чем $(n - 1)/4$ оснований a , $0 < a < n$. Для этого предположим $a^{n-1} \equiv 1 \pmod{n}$, откуда $a^{n-1} \equiv 1 \pmod{p^2}$. Поскольку $(\mathbb{Z}/p^2\mathbb{Z})^*$ – циклическая группа порядка $p(p - 1)$, то существует целое g такое, что

$$(\mathbb{Z}/p^2\mathbb{Z})^* = \{g, g^2, \dots, g^{p(p-1)}\}.$$

Согласно лемме 12.2 количество тех a , для которых $a^{n-1} \equiv 1 \pmod{p^2}$, равно $d = (p(p - 1), n - 1)$. Поскольку $p \mid n$, то $p \nmid n - 1$ и $d \mid p - 1$, откуда $d \leq p - 1$. Это значит, что доля всех a , не делящихся на p^2 в интервале от 0 до n и удовлетворяющих сравнению $a^{n-1} \equiv 1 \pmod{p^2}$, не превосходит величины

$$\frac{p - 1}{p^2 - 1} \leq \frac{1}{p + 1} \leq \frac{1}{4}.$$

Это доказывает первый случай теоремы.

Случай 2. Предположим, что n является произведением двух различных простых p и q , $n = pq$. Пусть $p = 2^{s_1}t_1 + 1$, $q = 2^{s_2}t_2 + 1$, t_1, t_2 – нечетные. Без ограничения общности можно предположить $s_1 \leq s_2$. Чтобы элемент a был основанием, по которому n было бы сильно псевдопростым, должно выполняться одно из следующих условий:

- 1) $a^t \equiv 1 \pmod{p}$ и $a^t \equiv 1 \pmod{q}$;
- 2) $a^{2^rt} \equiv -1 \pmod{p}$ и $a^{2^rt} \equiv -1 \pmod{q}$ для некоторого r , $0 \leq r \leq s$. \square

По лемме 9.1 количество тех a , для которых выполнено первое условие, равно произведению (t, t_1) (количество классов вычетов по модулю p) на (t, t_2) (количество классов вычетов по модулю q), что, очевидно, не превосходит $t_1 t_2$. По лемме 9.2 для каждого $r < \min(s_1, s_2) = s_1$ количество таких a , что $a^{2^r t} \equiv -1 \pmod{n}$, равно $2^r(t, t_1)2^r(t, t_2) \leq 4^r t_1 t_2$. Из $n - 1 > \varphi(n) = 2^{s_1+s_2}t_1 t_2$ следует, что доля тех a , $0 < a < n$, для которых n является сильно псевдопростым, не превосходит

$$\frac{t_1 t_2 + t_1 t_2 + 4t_1 t_2 + 4^2 t_1 t_2 + \dots + 4^{s_1-1} t_1 t_2}{2^{s_1+s_2} t_1 t_2} = 2^{-s_1-s_2} \left(1 + \frac{4^{s_1} - 1}{4 - 1} \right).$$

Если $s_2 > s_1$, то указанная выше величина не превосходит $2^{-2s_1-1}(2/3 + 4^{s_1}/3) \leq 2^{-3}(2/3) + 1/6 = 1/4$. В случае $s_1 = s_2$ заметим, что одно из двух неравенств $(t, t_1) \leq t_1$, $(t, t_2) \leq t_2$ является строгим, как если бы $t_1 \mid t$ и $t_2 \mid t$, то из того, что $p \equiv 1 \pmod{t_1}$ и $2^s t = pq - 1$ получим $t_1 \mid q - 1 = 2^{s_2} t_2$, т. е. $t_1 \mid t_2$. Аналогично получаем $t_2 \mid t_1$, откуда $t_1 = t_2$ и $p = q$, что противоречит условию рассматриваемого случая.

Следовательно, либо $(t, t_1) < t_1$, либо $(t, t_2) < t_2$, и поскольку мы имеем дело с нечетными числами, то $(t, t_1)(t, t_2) \leq t_1 t_2 / 3$. Таким образом, доля тех a , для которых n – сильно псевдопростое, оценивается сверху величиной

$$\frac{1}{3} 2^{-2s_1} \left(\frac{2}{3} + \frac{4^{s_1}}{3} \right) \leq \frac{1}{18} + \frac{1}{9} < \frac{1}{4}.$$

Тем самым второй случай доказан.

Случай 3. Наконец предположим, что n – произведение более чем двух различных простых чисел: $n = p_1 \cdots p_k$, $k \geq 3$. Пусть $p_j - 1 = 2^{s_j} t_j$, где t_j – нечетные числа, и пусть s_1 – минимальное из s_j . Поступая так же, как во втором случае, получим следующую верхнюю границу доли тех a , для которых n – сильно псевдопростое:

$$\begin{aligned} 2^{-s_1-s_2-\dots-s_k} \left(1 + \frac{2^{ks_1} - 1}{2^k - 1} \right) &\leq 2^{-ks_1} \left(\frac{2^k - 2}{2^k - 1} + \frac{2^{ks_1}}{2^k - 1} \right) = \\ &= 2^{-k} \frac{2^k - 2}{2^k - 1} + \frac{1}{2^k - 1} = 2^{1-k} \leq \frac{1}{4}. \end{aligned}$$

Тест Миллера – Рабина применяется при генерации простых чисел для стандартов цифровой подписи DSA, ECDSA.

9.4. ПОСТРОЕНИЕ БОЛЬШИХ ПРОСТЫХ ЧИСЕЛ

Таким образом, генерация больших чисел может производиться с помощью теста Миллера – Рабина посредством выбора случайного нечетного числа в заданном диапазоне и последующего его тестирования на простоту.

Также может применяться другой подход, позволяющий получать детерминированно простые числа, однако имеющие специальную структуру. Именно такой подход применяется при генерации простых чисел для ЭЦП в ГОСТ Р 34.10-94: детерминированный тест простоты, основанный на следующем утверждении.

Теорема 9.5. Пусть q – нечетное простое и $p = qN + 1$, где N – четное. Если также $p < (2q + 1)^2$ и выполняются условия:

- 1) $2^{qN} \equiv 1 \pmod{p}$;
- 2) $2^N \not\equiv 1 \pmod{p}$, то p – простое.

Доказательство. Пусть $p = p_1^{\alpha_1} \dots p_s^{\alpha_s}$, d – порядок числа 2 по модулю p . Тогда:

- a) $d \mid p - 1$ в силу 1);
- б) $d \nmid \frac{p - 1}{q}$ в силу 2);
- в) $d \mid \varphi(p)$ по теореме Эйлера, где $\varphi(p)$ – функция Эйлера.

Из а) и б) следует, что $q \mid d$, и значит, $q \mid \varphi(p)$. Известно, что $\varphi(p) = p_1^{\alpha_1-1} \dots p_s^{\alpha_s-1} (p_1 - 1) \dots (p_s - 1)$. Предположим, что q совпадает с одним из p_i . Тогда существует натуральное n такое, что $p = qn$, и значит, в силу условия данной теоремы $qn = qN + 1$. Это противоречие, так как $q \neq 1$.

Таким образом, q должен делить один из множителей $p_i - 1$, т. е. $p_i = qn + 1$ для некоторого n . Отсюда имеем $p = p_i m = (qn + 1)m = qN + 1$ и, следовательно, $m - 1 = q(N - nm)$. Итак, $p = (qn + 1)(qs + 1)$, где s и n – четные числа, причем $n \geq 2$, $s \geq 0$.

Предположим, что p – составное. Тогда $s \geq 2$, откуда $p \geq (2q + 1)^2$. Из этого противоречия следует, что $s = 0$, значит, $p = p_i$ – простое. \square

Алгоритм, приведенный в ГОСТ Р 34.10-94, позволяет строить простые числа, у которых длина двоичного разложения больше или равна 17. Идея алгоритма заключается в следующем. Пусть требуется построить простое число p длиной t битов ($t \geq 17$). Для этого построим убывающий набор натуральных чисел t_0, \dots, t_s таких, что $t_0 = t$, $t_s < 17$ и $t_{i+1} = [t_i/2]$, т. е. либо $t_{i-1} = 2t_i + 1$, либо $t_{i-1} = 2t_i$. В процессе работы алгоритма будем получать последовательно простые числа p_s, p_{s-1}, \dots, p_0 , причем длина каждого p_i в точности равна t_i битов. На первом шаге алгоритма посредством алгоритма пробного деления получаем простое p_s длиной $t_s < 17$.

Далее несколько раз выполняется следующий шаг итерации. Пусть имеется простое число p_i длиной t_i битов, тогда простое число p_{i-1} длиной t_{i-1} битов ищется в виде $p_{i-1} = p_i N + 1$, где N удовлетворяет следующим условиям:

- 1) N – четное;
- 2) N – такое, что длина числа $p_i N + 1$ в точности равна t_i битов.

Такое N получают с помощью датчика случайных чисел. После этого число $p_{i-1} = p_i N + 1$ тестируют на простоту с помощью следующего теста.

Проверим два условия:

- 1) $2^{p_i N} \equiv 1 \pmod{p_{i-1}}$;
- 2) $2^N \not\equiv 1 \pmod{p_{i-1}}$.

Если оба они выполняются одновременно, то число p_{i-1} считается простым.

Если хотя бы одно из условий не выполнено, то число p_{i-1} считается составным, N увеличивается на 2 и тест повторяется для нового $p_{i-1} = p_i N + 1$.

Процедура повторяется до получения простого числа p_0 длиной t_0 битов.

9.5. АЛГОРИТМЫ СЛОЖЕНИЯ ТОЧЕК ЭЛЛИПТИЧЕСКОЙ КРИВОЙ

Эффективная работа любого криптографического алгоритма, в котором используются эллиптические кривые (ЭК), в том числе и любого алгоритма дискретного логарифмирования на эллиптических кривых, существенно зависит от эффективности реализации групповой операции. В настоящем разделе будут рассмотрены современные способы задания групповой операции на ЭК и методы ее эффективной реализации.

Эллиптические кривые можно представлять в разных формах, так же как и вычисления на ЭК можно выполнять разными способами. Наличие большого количества представлений группы точек ЭК и способов вычисления на этих кривых позволяет в различных ситуациях использовать тот или иной способ для достижения максимальной производительности.

Далее мы опишем два популярных и эффективных подхода к реализации операции сложения точек на эллиптической кривой.

1. Представление ЭК в форме Вейерштрасса: $y^2 = x^3 + Ax + B$, где точки ЭК представлены либо аффинными координатами (x, y) , либо тройками взвешенных (иногда называемых якобиевыми) координат: (X, Y, Z) . Здесь связь с аффинными координатами устанавливается соотношениями $x = X/Z^2$, $y = Y/Z^3$. В этом случае тройки (X, Y, Z) удовлетворяют взвешенному уравнению

$$Y^2 = X^3 + AXZ^4 + BZ^6.$$

Представление во взвешенных координатах позволяет вычислять групповую операцию на ЭК быстрее, чем при представлении точек в обычных проективных координатах.

2. Представление ЭК в форме Монтгомери: $By^2 = x^3 + Cx^2 + x$, где для представления точки используется только первая координата x или ее проективное представление в виде пары чисел (X, Z) , связанных с x по правилу $x = X/Z$. Особенностью этого представления является то, что операцию сложения точек можно выполнять используя только координату x точки, не

привлекая для вычислений координату y . Однако недостатком этого алгоритма является то, что для нахождения суммы двух точек необходимо знать координату x разности этих точек.

Далее рассмотрим подробнее эти два представления и способы вычисления групповой операции в этих случаях.

9.5.1. Алгоритмы сложения в полях нечетной характеристики

Пусть задано поле F , характеристика которого отлична от 2 и 3. Напомним, что эллиптическая кривая в форме Вейерштрасса – это кривая, уравнение которой с помощью рациональной биективной замены переменных может быть приведено к виду

$$y^2 = x^3 + Ax + B, \text{ где } \Delta = -(4A^3 + 27B^2) \neq 0, \quad A, B \in F. \quad (9.4)$$

Группа точек кривой, задаваемой уравнением (9.4), состоит из пар $(x, y) \in \bar{F} \times F$, удовлетворяющих уравнению (9.4), и дополнительной точки \mathcal{O} . Здесь \bar{F} – алгебраическое замыкание поля F . Точка \mathcal{O} является нейтральным элементом группы. Групповая операция сложения для точек этой кривой определяется по следующему правилу.

1. Для любой точки $P = (x_1, y_1)$ противоположная точка вычисляется по правилу $-P = (x_1, -y_1)$.

2. Сумма $R = (x_3, y_3)$ точек $P = (x_1, y_1)$ и $Q = (x_2, y_2)$ в случае $P \neq -Q$ и $P, Q \neq \mathcal{O}$ вычисляется по следующим формулам:

$$x_3 = m^2 - x_1 - x_2, \quad y_3 = m(x_1 - x_3) - y_1, \quad (9.5)$$

где

$$m = \begin{cases} \frac{y_1 - y_2}{x_1 - x_2}, & \text{если } P \neq Q, \\ \frac{3x_1^2 + A}{2y_1}, & \text{если } P = Q. \end{cases} \quad (9.6)$$

Выполнение операции по данным формулам требует деления (вычисления обратного элемента) в поле F , что, как правило, является трудоемкой операцией. Избежать этого можно, представляя точку в проективных или взвешенных координатах. Во взвешенных координатах точка кривой задается тройкой (X, Y, Z) . Переходы между ними происходят по правилу

$$(x, y) \rightarrow (x, y, 1)$$

и обратно: $(X, Y, Z) \rightarrow (X/Z^2, Y/Z^3)$.

Нейтральный элемент во взвешенных координатах задается тройкой вида $(\lambda^2, \lambda^3, 0)$ при любом $\lambda \neq 0$ (обычно полагают $\lambda = 1$). Запишем операцию сложения точек во взвешенных координатах. Подставляя $x_i = X_i/Z_i^2$ и $y_i = Y_i/Z_i^3$ в формулы для суммы точек (9.5) и (9.6), получим, что взвешенные координаты точки $R = (X_1, Y_1, Z_1)$, которая является суммой точек $P = (X_1, Y_1, Z_1)$ и $Q = (X_2, Y_2, Z_2)$ с условием $Q \neq \pm P$ (и $P, Q \neq \mathcal{O}$), вычисляются по формулам:

$$X_3 = T_1^2 - H_2 H_1^2, \quad Y_3 = (V T_1 - T_2 H_1^3)/2, \quad Z_3 = Z_1 Z_2 H_1, \quad (9.7)$$

где $T_1 = S_1 - S_2$, $T_2 = S_1 + S_2$, где $S_1 = Y_1 Z_2^3$, $S_2 = Y_2 Z_1^3$; $H_1 = U_1 - U_2$, $H_2 = U_1 + U_2$, где $U_1 = X_1 Z_2^2$, $U_2 = X_2 Z_1^2$; $V = H_2 H_1^2 - 2X_3$.

Обозначим теперь через M – количество единиц времени, необходимое для выполнения одного умножения, S – количество единиц времени, необходимое для выполнения одного возведения в квадрат (в зависимости от реализации и архитектуры ЭВМ величина S лежит в диапазоне $0,66M < S < M$). Приведенные выше формулы показывают, что для реализации сложения двух различных точек требуется $4S + 12M$ единиц времени. При реализации формул сложения во взвешенных координатах на тех архитектурах, где возведение в квадрат выполняется существенно быстрее умножения, можно воспользоваться равенством $2Z_1 Z_2 = (Z_1 + Z_2)^2 - Z_1^2 - Z_2^2$ для вычисления $Z_1 Z_2$ в формулах (9.7), что дает $5S + 11M$ единиц времени.

Удвоение (т. е. случай $P = Q$) точки $P = (X_1, Y_1, Z_1)$ происходит по формулам:

$$X_3 = M^2 - 2S, \quad Y_3 = M(S - X_3) - 8N, \quad Z_3 = 2Y_1 Z_1, \quad (9.8)$$

где $M = 3X_1^2 + AZ_1^4$; $N = Y_1^4$; $S = 4X_1 Y_1^2$. Если обозначить через c количество единиц времени, необходимых для выполнения одного умножения на константу, то общее время удвоения точки получится равным $6S + 3M + 1c$ или $8S + 1M + 1c$, если вычислять S и Z_3 , используя только возведения в квадрат:

$$S = 2 \left((X_1 + Y_1^2)^2 - X_1^2 - N \right) \text{ и } Z_3 = (Y_1 + Z_1)^2 - Y_1^2 - Z_1^2.$$

Заметим, что формулы (9.8) годятся и для удвоения точки \mathcal{O} .

Имеется ряд частных случаев, которые часто встречаются на практике и которые позволяют существенно ускорить выполнение групповой операции во взвешенных координатах. В случае, когда $Z_2 = 1$, а такая ситуация часто возникает при вычислении кратной точки, формулы (9.7) упрощаются, и требуется только $4S + 7M$ единиц времени для сложения неравных точек.

Еще один частный случай возникает, когда коэффициент кривой $A = -3$. В этом случае промежуточная величина M в формулах (9.8) может быть вычислена как $M = 3(X_1 + Z_1^2)(X_1 - Z_1^2)$, что дает $5S + 3M$ единиц времени для удвоения точки.

П. Монтгомери предложил способ очень быстрого вычисления групповой операции на кривой специального вида при условии, что известны только первые координаты складываемых точек и первая координата их разности [136]. Было предложено использовать кривые вида $By^2 = x^3 + Cx^2 + x$, для которых операция сложения выполняется по следующим правилам.

1. Для любой точки $P = (x_1, y_1)$ кривой Монтгомери противоположная точка вычисляется как $-P = (x_1, -y_1)$.

2. Сумма $R = (x_3, y_3)$ точек $P = (x_1, y_1)$ и $Q = (x_2, y_2)$ в случае $P \neq -Q$ и $P, Q \neq \mathcal{O}$ вычисляется по формулам:

$$x_3 = Bm^2 - x_1 - x_2 - C, \quad y_3 = m(x_1 - x_3) - y_1, \quad (9.9)$$

где

$$m = \begin{cases} \frac{y_1 - y_2}{x_1 - x_2}, & \text{если } P \neq Q, \\ \frac{3x_1^2 + 2Cx_1 + 1}{2By_1}, & \text{если } P = Q. \end{cases} \quad (9.10)$$

Эти формулы легко получить, если заметить, что замена переменной $x = x' - C/3$ приводит кривую Монтгомери к форме Вейерштрасса. Для кривых в форме Монтгомери имеет место следующая теорема.

Теорема 9.6. Пусть F – поле, характеристика которого не равна 2, а B, C – элементы F такие, что $B(C^2 - 4) \neq 0$. Определим кривую Монтгомери E как эллиптическую кривую, задаваемую уравнением $By^2 = x^3 + Cx^2 + x$. Пусть $P = (x_1, y_1)$ и $Q = (x_2, y_2)$ – две точки, лежащие на E и удовлетворяющие условиям $P \neq \mathcal{O}$, $Q \neq \mathcal{O}$, $P \neq -Q$. Пусть также $(x_3, y_3) = P + Q$. Тогда:

1) если $P \neq Q$, то

$$x_- x_3 = \frac{(x_1 x_2 - 1)^2}{(x_1 - x_2)^2}, \quad (9.11)$$

где x_- – первая координата точки $P - Q$;

2) если $P = Q$, то

$$x_3 = \frac{(x_1^2 - 1)^2}{4(x_1^3 + Cx_1^2 + x_1)}. \quad (9.12)$$

Доказательство этих формул можно найти в [136]. Далее, переходя к проективным координатам с помощью замены $x_i = X_i/Z_i$ и подставляя ее в формулы (9.11), получим следующие формулы для сложения точек кривой Монтгомери в проективных координатах в случае $P \neq Q$:

$$\begin{aligned} X_3 &= Z_-((X_1 - Z_1)(X_2 + Z_2) + (X_2 - Z_2)(X_1 + Z_1))^2; \\ Z_3 &= X_-((X_1 - Z_1)(X_2 + Z_2) - (X_2 - Z_2)(X_1 + Z_1))^2. \end{aligned}$$

Это позволяет вычислять сумму за $2S + 4M$ единиц времени, или, если $Z_- = 1$ (а при вычислении кратной точки, заданной в аффинных координатах, возникает именно такая ситуация), то время работы составляет $2S + 3M$ единиц времени.

В случае $P = Q$, замена $x_i = X_i/Z_i$ в (9.12) дает

$$X_3 = (X_1 + Z_1)^2(X_1 - Z_1)^2; \quad Z_3 = 4X_1Z_1 \left((X_1 - Z_1)^2 + \frac{C+2}{4} \cdot 4X_1Z_1 \right),$$

где величина $4X_1Z_1$ может быть вычислена по формуле $4X_1Z_1 = (X_1 + Z_1)^2 - (X_1 - Z_1)^2$. Таким образом, удвоение точки требует $2S + 2M + 1c$ единиц времени.

Эти результаты для кривых Монтгомери, очевидно, существенно превосходят результаты для кривых Вейерштрасса, точки которых представлены во взвешенных координатах. Однако условие известности x_- требует изменения алгоритма вычисления кратной точки. А именно, используется метод, получивший название «лестница Монтгомери» (Montgomery ladder), суть которого заключается в том, что на каждом шаге хранятся первые координаты двух точек: nP и $(n+1)P$ – в отличие от стандартного бинарного алгоритма, в котором хранится только одна промежуточная точка. Это дает возможность использовать формулы Монтгомери, поскольку разность точек $(n+1)P$ и nP известна и равна P .

9.6. ВЫЧИСЛЕНИЕ КРАТНОЙ ТОЧКИ ЭЛЛИПТИЧЕСКОЙ КРИВОЙ

АЛГОРИТМ Вычисление кратной точки для кривой Монтгомери

Вход: Коэффициенты кривой в форме Монтгомери. Точка P , лежащая на кривой Монтгомери, заданная только парой проективных координат (X, Z) , соответствующих первой аффинной координате точки. Натуральное число k .

Выход: Точка kP , заданная парой проективных координат (X, Z) , соответствующих первой аффинной координате точки.

1. Представить k в двоичной системе счисления: $k = k_0 + k_12^1 + k_22^2 + \dots + k_{l-1}2^{l-1} + 2^l$.
 2. Присвоить $Q := P$, $R := 2P$.
 3. Для каждого i от $l - 1$ до 0 выполнить следующие действия:
 - а) если $k_i = 0$, то по формулам Монтгомери вычислить $Q' := 2Q$, $R' := Q + R$;
 - б) если $k_i = 1$, то по формулам Монтгомери вычислить $Q' := Q + R$, $R' := 2R$;
 - в) присвоить $Q := Q'$, $R := R'$.
 4. Вывести Q .
-

В данном алгоритме после каждого очередного шага цикла фактически выполняется присваивание $Q := 2Q + k_i P$ – именно это присваивание выполняется на каждом шаге в стандартном бинарном алгоритме вычисления кратной точки. При этом на каждом шаге цикла выполнено условие $R - Q = P$. Это значит, что для вычисления каждой суммы точек $Q + R$ применимы формулы Монтгомери, так как разность $R - Q$ известна. В начальный момент это соотношение также выполняется.

9.7. ЗАДАНИЯ

1. Доказать корректность следующих бинарных методов возведения числа u в степень $e = (e_{l-1} \dots e_1 e_0)_2$, $e_{l-1} \neq 0$ (см. п. 9.1).

Справа налево	Слева направо
1. Установить $w \leftarrow 1$, $v \leftarrow u$. 2. Для $i = 0, 1, \dots, l-1$ выполнить: а) если $e_i \neq 0$, то $w \leftarrow wv$; б) $v \leftarrow vv$. 3. Вернуть w .	1. Установить $w \leftarrow 1$. 2. Для $i = l-1, l-2, \dots, 0$ выполнить: а) $w \leftarrow ww$; б) если $e_i \neq 0$, то $w \leftarrow wi$. 3. Вернуть w .

2. Предложить представление больших чисел в машинной памяти. Разработать алгоритм деления на одноразрядное число. Реализовать алгоритмы сравнения, сложения, вычитания, умножения и деления больших чисел (см. п. 9.1).

3. Разработать и реализовать алгоритмы обращения в \mathbb{Z}_n^* для случая, когда n является степенью 2. Использовать, предварительно доказав, следующие утверждения:

1) если $n = 2^s$, a – нечетное, $b_1 = 1$ и

$$b_t = \begin{cases} b_{t-1}, & \text{если } (ab_{t-1} \bmod 2^t) < 2^{t-1}, \\ b_{t-1} + 2^{t-1} & \text{в противном случае,} \end{cases} \quad t = 2, 3, \dots,$$

то $b_s = a^{-1}(\bmod n)$;

2) если $n = 2^{2^s}$, a – нечетное, $c_1 = a$ и

$$c_t = (2c_{t-1} - ac_{t-1}^2) \bmod n, \quad t = 2, 3, \dots,$$

то $c_s = a^{-1}(\bmod n)$ (см. п. 9.1).

4. Для увеличения быстродействия метода Барретта число \hat{r} при $b > 3$ определяют по следующему алгоритму:

- 1) $r_1 \leftarrow x \bmod b^{k+1}$, $r_2 \leftarrow \hat{q}n \bmod b^{k+1}$;
- 2) $\hat{r} \leftarrow r_1 - r_2$;

3) если $\hat{r} < 0$, то установить $\hat{r} \leftarrow \hat{r} + b^{k+1}$.

Доказать, что и в этом случае $\hat{r} = x - \hat{q}n$ (см. п. 9.2).

5. Используя задание 1, доказать корректность алгоритма возведения в степень по Монтгомери (см. п. 9.2).

6. Реализовать алгоритмы умножения и возведения в степень по Монтгомери. Выбрать основание b как степень 2, а для вычисления n' использовать алгоритмы из задания 3 (см. п. 9.2).

7. Пусть $n = b^k - m$, где m – l -разрядное число по основанию b , $l < k$. Для натурального $x = q_0b^k + r_0$, $0 \leq r_0 < b^k$, по правилу

$$q_{i-1}m = q_ib^k + r_i, \quad 0 \leq r_i < b^k,$$

определенны числа $q_i, r_i, i = 1, 2, \dots$. Доказать:

1) существует натуральное t , для которого $q_t = 0$;

2) $x + (q_0 + \dots + q_{t-1})m = (q_0 + \dots + q_{t-1})b^k + (r_0 + \dots + r_t)$;

3) $x \equiv (r_0 + \dots + r_t) \pmod{n}$.

Используя данные выводы, разработать алгоритм и написать программу вычисления $x \pmod{n}$ (см. п. 9.2).

8. Написать программу вычисления символа Якоби $\left(\frac{a}{n}\right)$, не требующую факторизации n (см. п. 9.2). Использовать следующие свойства:

1) если $a \equiv b \pmod{n}$, то $\left(\frac{a}{n}\right) = \left(\frac{b}{n}\right)$;

2) $\left(\frac{1}{n}\right) = 1$;

3) $\left(\frac{-1}{n}\right) = (-1)^{(n-1)/2}$;

4) $\left(\frac{ab}{n}\right) = \left(\frac{a}{n}\right) \left(\frac{b}{n}\right)$;

5) $\left(\frac{2}{n}\right) = (-1)^{(n^2-1)/8}$, иначе говоря, $\left(\frac{2}{n}\right) = 1$, если $n \equiv \pm 1 \pmod{8}$, и $\left(\frac{2}{n}\right) = -1$, если $n \equiv \pm 3 \pmod{8}$;

6) если a и n – взаимно простые нечетные числа, то

$$\left(\frac{a}{n}\right) = (-1)^{(a-1)(n-1)/4} \left(\frac{n}{a}\right).$$

9. Согласно теореме Эйлера, если n – нечетное простое и $(a, n) = 1$, то

$$a^{(n-1)/2} \equiv \left(\frac{a}{n}\right) \pmod{n}. \quad (9.13)$$

Доказать, что если n – нечетное составное число, то соотношение (9.13) не выполняется по крайней мере для половины всех a с условием $(a, n) = 1$ (см. п. 9.3).

10. Доказать, что если (9.13) выполняется для a_1 и не выполняется для a_2 , то (9.13) не выполняется для a_1a_2 . Применить данное утверждение для доказательства того, что если (9.13) не выполняется хотя бы для одного a , то количество таких a не меньше количества тех a , для которых (9.13) выполнено (см. п. 9.3).

11. Если n делится на квадрат простого числа, то показать, как найти целое a такое, что $a^{(n-1)/2} \not\equiv \pm 1 \pmod{n}$ (см. п. 9.3).

12. Доказать, что если n – произведение различных простых чисел, p – одно из них, a обладает свойством $\left(\frac{a}{p}\right) = -1$ и $a \equiv 1 \pmod{n/p}$, то (9.13) не выполняется для этого a . Показать, что такое a всегда существует (см. п. 9.3).

13. Числа вида $M_n = 2^n - 1$, где $n = 2, 3, \dots$, называют *числами Мерсенна*. Известно, что M_n является простым тогда и только тогда, когда:

- 1) n – простое;
- 2) $(n-2)$ -й элемент последовательности $s_0 = 4$, $s_t = (s_{t-1}^2 - 2) \pmod{n}$, $t = 1, 2, \dots$, равен 0.

Написать программу, доказывающую простоту первых 20 чисел Мерсенна из табл. 9.3 (см. п. 9.3).

Таблица 9.3

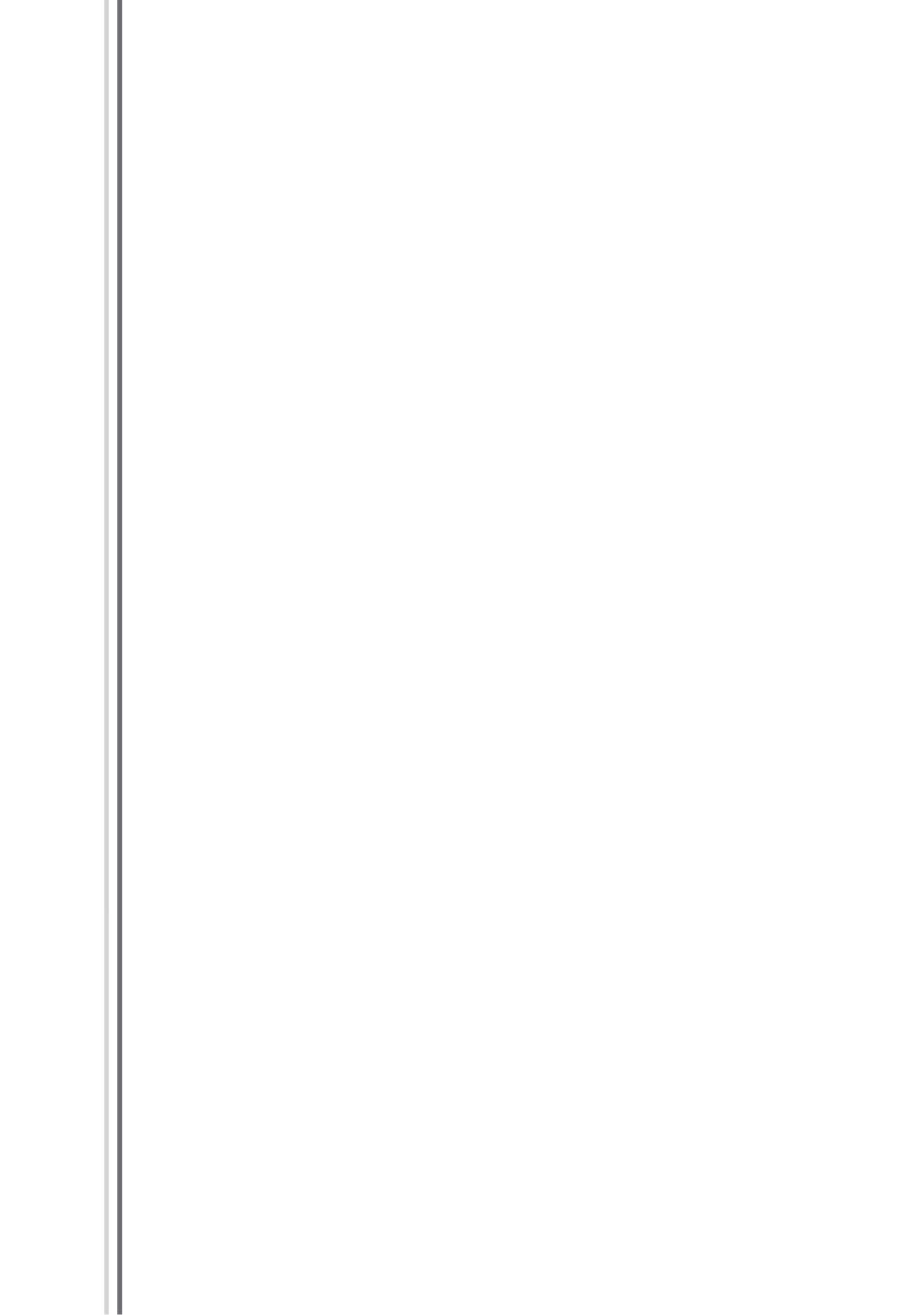
**Последовательные простые
числа Мерсенна $M_n = 2^n - 1$**

n	M_n	n	n	n	n
2	3	61	2281	21701	859433
3	7	89	3217	23209	1257787
5	31	107	4253	44497	1398269
7	127	127	4423	86243	2976221
13	8191	521	9689	110503	3021377
17	131071	607	9941	132049	6972593
19	524287	1279	11213	216091	13466917
31	2147483647	2203	19937	756839	

Часть II

КРИПТОГРАФИЧЕСКИЕ СИСТЕМЫ И МЕТОДЫ КРИПТОАНАЛИЗА

sceitzt6w4vjas88kd4y27h1m7xh10c0gwqqtyti2r
:46h38i8oeg5rq2ucwtuvtdqo364w0t0lna44xqtjc
ev6emk0041n0urusrj4vv4f4dfhzdl11jgaa693lz
guae9og3s6j46x61iwshy038zhyre32ro5obs7aa4
bd6csdsz7xz1zfxtcg5banmbewxxwrhqhlew6ckbc
x8su14a49an5dq8ezvjp26pk9bwnjcmydwi5yv33u
phig9b1jtj5tam8ojhiastniziq7k69piqc9eaaur
jvhz7hmhm30b7qbe13washk9hdzk41exex2u34fk8
tylwbst34olqg6qkyktqeu3wpkk7m1tdudgr0y38c
vi5svuuyptzi8hdxdh7cnvhid4uhmo42cyw1sr6vjk
a2weud80j1hjfcowh2aedtrdzrd90dpprc0m6rl0r
c2010ni3wosdt6esmt202b1uqlh2bfj6ksolm2zbu
8n54msbmq5vh386bqs3a1pu6oaviohk677u9pvk12
.ifqgk5f7b2hvxa8v3zecj7uf1dw8j503r3kwjx0ggi
v0fwf6h4c7mmq057vv6fj0p3dry59dlsdm2wvsfxu
zeulibgnqh3ac52h6tx1gj2mq3aa5i6vlkzeswoc
or2zw318rf2p95nztu2la4b151bxsx1dsr3tdmmwl
4ajz1n0w4o7pjxgx1av9qf5djjad0b2o8b4fuqt2y
mo58zwt3x43y23th5ufrbjazzxdpp25mnak0ab09h
yqut7vts03c1lu0r2vcx5o5z370xpx9woq4hafhh5
bvwx5o85mhvsat1e8rvc5575ddjiyldsp6wzdwmay
.21jeib1g9iyb2no7fi5ujss7gilxohato9ivugl5y
iaua7c1vld3ttfx4awf05cne56v67cnhmbik8pxpw4



Г л а в а 10

ГЕНЕРАЦИЯ СЛУЧАЙНЫХ И ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

10.1. КЛАССИФИКАЦИЯ АЛГОРИТМОВ ГЕНЕРАЦИИ

Случайные числа и их генераторы являются неотъемлемыми элементами современных криптосистем. Приведем конкретные примеры использования случайных чисел в криптологии:

- 1) сеансовые и другие ключи для симметричных криптосистем, таких как DES, ГОСТ 28147-89, Blowfish;
- 2) стартовые значения для программ генерации ряда математических величин в асимметричных криптосистемах, например «больших простых чисел» в криптосистемах RSA, ElGamal;
- 3) случайные слова, комбинируемые с парольными словами для нарушения «атаки угадывания» пароля криptoаналитиком;
- 4) вектор инициализации для блочных криптосистем, работающих в режиме обратной связи;
- 5) случайные значения параметров для многих систем электронной цифровой подписи, например, DSA или белорусского стандарта СТБ 1176.2-99;
- 6) случайные выборы в протоколах аутентификации, например в протоколе Kerberos;
- 7) случайные параметры протоколов для обеспечения уникальности различных реализаций одного и того же протокола, например в протоколах SET и SSL.

Отметим, что для некоторых из этих криптографических применений необходимы огромные массивы случайных чисел, которые по своему назначению требуют конфиденциального использования. Например, в протоколе Цербер сетевой сервер генерирует тысячи сессионных ключей ежесекундно. К сожалению, компьютеры по своей конструкции предназначены быть детерминированными системами, поэтому на современных компьютерах генерация случайных чисел весьма затруднительна.

Известно, что проблема генерации случайной последовательности с произвольным законом распределения вероятностей сводится к проблеме генерации, рассмотренной в п. 5.2, *равномерно распределенной случайной последовательности* (РРСП), или, как ее часто называют в криптографических приложениях, «чисто случайной» последовательности.



Рис. 10.1. Классификация алгоритмов генерации псевдослучайных последовательностей

С учетом свойств C_1-C_{10} (см. п. 5.2) определим понятия генератора случайной последовательности и его типов.

Генератор РРСП – устройство, позволяющее по запросу получить реализацию равномерно распределенной случайной последовательности $x_1, \dots, x_n \in \mathcal{A}$ длиной $n \in \mathbb{N}$; элементы x_1, \dots, x_n этой реализации принято называть случайными числами. Существует два типа генераторов РРСП: 1) программный; 2) физический.

Программный генератор РРСП – это программа для имитации на компьютере реализации РРСП с помощью псевдослучайной последовательности $\{x_t\}$, которая: 1) вычисляется на компьютере по известному детерминированному рекуррентному соотношению; 2) по своим статистическим свойствам неотличима от РРСП.

Классификация существующих алгоритмов генерации псевдослучайных последовательностей представлена на рис. 10.1. Выделяются три основных подхода к построению алгоритмов генерации:

1) *прямые методы* построения элементарных рекуррентных последовательностей $x_t = f(x_{t-1}, x_{t-2}, \dots, x_{t-m}) : \mathcal{A}^m \rightarrow \mathcal{A}$, проходящих статистические тесты из гл. 7;

2) *методы «улучшения элементарных последовательностей*, заключающиеся в специальных функциональных (фильтрующих) преобразованиях этих последовательностей для уменьшения отклонения их статистических свойств от свойств РРСП;

3) *комбинирование алгоритмов генерации*, построенных с помощью первого или второго подхода.

Описание каждого из алгоритмов генерации псевдослучайных последовательностей, указанного на рис. 10.1, приводится в пп. 10.3–10.10 данной главы. Физические генераторы рассмотрены в п. 10.2

10.2. ФИЗИЧЕСКИЕ ГЕНЕРАТОРЫ СЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Стойкость многих криптографических протоколов и алгоритмов шифрования зависит от качества используемого источника случайных последовательностей. Так, если для формирования ключей криптографического алгоритма используется генератор псевдослучайных последовательностей, то для взлома системы, основанной на данном алгоритме, достаточно знания начального состояния генератора. Для того чтобы система была защищена, требуется обеспечение непредсказуемости случайной последовательности. Это возможно только с использованием физических генераторов случайных чисел, основанных на случайных процессах.

Физические генераторы случайных последовательностей основаны на измерении определенных характеристик случайных физических процессов.

Опишем некоторые типы процессов, использующихся при построении генераторов.

1. *Процессы, кажущиеся случайными.* Случайность физического процесса может быть обусловлена его сложностью на макроуровне, в то время как на микроуровне процесс может иметь скрытую детерминированную структуру. К таким процессам относятся, например, хаотические процессы, описываемые сложной системой дифференциальных уравнений, а также шумовые процессы, протекающие в системах, состоящих из многих частиц. Примером используемого хаотического оптического процесса могут служить флуктуации поляризации излучения, генерируемого в полупроводниковых лазерах [78]. Среди шумовых процессов на практике распространены тепловой (джонсоновский) шум, дробовой шум, лавинный шум, которые имеют электрически измеряемые характеристики и основаны на случайном поведении электронов или материала. Некоторые стандартные компоненты электронных схем могут помимо своей основной роли использоваться в качестве генераторов случайных сигналов. Например, резисторы являются источниками теплового и дробового шума, стабилитроны – источниками джонсоновского шума. Также разработаны специальные электронные схемы генерации случайных последовательностей для повышения качества генерируемых последовательностей. Так, в стандартизированной схеме компании Intel имеется два генератора шума: быстрый и медленный. Медленный генератор выступает в качестве генератора моментов считывания сигнала быстрого генератора, после чего сигнал усиливается и с помощью компаратора формируется бинарная последовательность [167].

2. *Процессы, являющиеся «истинно случайными».* В настоящее время активно развиваются исследования в области квантовых генераторов случайных последовательностей, так как квантовые процессы «истинно случайны». Подобные генераторы переводят квантово-механические процессы в случайные биты. Например, в области ядерной физики таким процессом является распад радиоактивных элементов, и в качестве генератора случайных чисел может выступать счетчик Гейгера, подсчитывающий число попавших в него ионизирующих частиц [60]. В области квантовой оптики рассматриваются процессы, связанные с выбором пути одиночным фотоном. Так, фирмой ID Quantique выпускаются генераторы, основанные на детектировании одиночного фотона после его попадания на светоделительную пластину. Также существуют реализации генераторов, основанных на квантовой природе вакуума, приводящей к флуктуации числа фотонов в разделенном лазерном пучке слабой интенсивности [159]. Достоинством квантовых генераторов является простота реализации, в силу которой генераторы имеют рассчитываемые характеристики [20].

Описанные выше источники случайности относятся к аппаратным. На практике также встречаются генераторы, использующие в качестве источника случайности действия человека, события, происходящие в компьютере. Например, время между нажатиями клавиш, щелчками мыши, системное время, время обращения к диску (являющееся случайной величиной из-за турбулентности воздушных потоков), данные микрофона и видеокарты. Недостатками этих методов являются сложность их реализации, медленная скорость, неизвестное количество энтропии этих источников случайности.

В общем случае физический генератор случайных чисел состоит из источника физической случайности, измерителя (детектора), переводящего сигнал источника в двоичную последовательность, и цифрового постобработчика. Источник генерирует «сырую» числовую последовательность с некоторым распределением, отличным от чисто случайного – модели независимых симметричных испытаний Бернулли. Цифровой постобработчик улучшает свойства «сырой» последовательности, приближая ее распределение к чисто случайному. Физические генераторы обладают недостатками, которые влияют на качество генерируемой последовательности. Среди них можно отметить, во-первых, чувствительность генераторов к внешним воздействиям: например, увеличение температуры в шумовом диоде может приводить к изменению параметров распределения выходной последовательности во времени. Во-вторых, особенности измерителя могут ухудшать параметры источника. Например, в генераторах, основанных на квантовой природе света, детекторы фотонов имеют ограниченную эффективность (вероятность срабатывания меньше 1), «мертвое» время, в течение которого детектор не регистрирует фотон после предыдущего измерения, а также разброс характеристик от прибора к прибору. Эти недостатки, а также недостатки, обусловленные конкретной архитектурой генератора, приводят к появлению нежелательных свойств выходной последовательности. Перечислим основные из них.

1. Наличие смещения – неравновероятности появления 0 и 1 в последовательности [90]. Смещение может быть вызвано изменением условий эксплуатации генератора (например, при увеличении температуры изменяются характеристики физического процесса), особенностями детектора и др.

2. Наличие автокорреляций между соседними генерируемыми битами и более сложных зависимостей. Зависимости могут появляться вследствие «мертвого» времени детектора, большой скорости считывания состояний и др.

Для устранения нежелательных свойств и приближения параметров распределения «сырой» последовательности к чисто случайной может использоваться архитектура генератора, включающая в себя несколько независимых источников случайности, а также постобработчики «сырых» последовательностей.

При использовании нескольких источников случайности суммирование по модулю 2 их «сырых» двоичных последовательностей приводит к получению выходной последовательности с улучшенными характеристиками. На практике распространен второй вариант архитектуры, в которой более медленный источник генерирует времена считывания показаний быстрого источника [167]. Считанные показания формируют «сырую» последовательность.

Перечислим постобработчики, улучшающие свойства «сырых» последовательностей.

1. Корректор Неймана [90]. Последовательность разбивается на непересекающиеся пары битов. Пара «01» преобразуется в «1», пара «10» преобразуется в «0», пары «00» и «11» отбрасываются. Такое преобразование позволяет полностью устраниТЬ смещение, если величина его постоянна и элементы последовательности независимы. Также такое преобразование позволяет уменьшить некоторые зависимости между битами, например избавиться от зависимости, если «сырая» последовательность является цепью Маркова первого порядка с постоянной вероятностью смены состояния. Недостатком корректора Неймана является переменная скорость генерации данных: невозможно заранее определить длину «сырой» последовательности для получения выходной последовательности заданной длины.

2. Суммирование по модулю 2 (операция XOR) [90]. Исходная двоичная последовательность разбивается на непересекающиеся подпоследовательности длиной N бит. Каждая подпоследовательность преобразуется в 1 бит суммированием всех ее элементов по модулю 2. Такое преобразование уменьшает смещение, если величина его постоянна и элементы последовательности независимы. В таблице приведены значения длины подпоследовательности N в зависимости от величины смещения – вероятности появления «1» в «сырой» последовательности, – при котором отклонение вероятности появления «1» от 0,5 в преобразованной последовательности не будет превышать 0,001.

**Длина подпоследовательности N
при разных величинах смещений**

$P\{1\}$	0,5	0,6	0,7	0,8	0,9	0,95	0,99
N	1	4	7	13	28	59	308

Следует, однако, заметить, что и первое, и второе преобразование может увеличить имеющиеся в последовательности стохастические зависимости.

3. Дискретное преобразование Фурье (ДПФ) (или быстрое преобразование Фурье, являющееся алгоритмом быстрого вычисления ДПФ). ДПФ вектора (x_0, \dots, x_{N-1}) имеет вид [143]

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{i2\pi kn}{N}}, \quad k = \overline{0, N-1}.$$

Данное преобразование уменьшает как смещение, так и сильные зависимости [87, 90].

4. Криптографические функции. Распространенным методом улучшения характеристик «сырой» последовательности является использование хэш-функций. Каждый блок последовательности некоторой длины преобразуется в образ хэш-функции фиксированной длины. В этом случае свойства выходной последовательности определяются свойствами используемой хэш-функции, что говорит о необходимости использования хэш-функций, признанных криптографическим сообществом надежными. В настоящее время являются, например, SHA-1 (SHA-2), RIPEMD-160. Функция SHA-1 используется в качестве постобработчика в физических генераторах Intel [167], а также при генерации случайных последовательностей из накопленных случайных системных данных в операционной системе Linux [106].

Также в качестве постобработчиков могут использоваться сжимающие алгоритмы, регистры сдвига с обратной связью, коррекция обратной связью (перенастройка компонент постобработчика для устранения обнаруженного смещения в сгенерированном блоке последовательности) и др. Как указывалось выше, чувствительность генераторов к внешним воздействиям приводит к нестабильности характеристик генерируемой случайной последовательности. Поэтому на практике необходимо статистическое тестирование каждой сгенерированной последовательности на предмет согласия с гипотезой о чистой случайности.

10.3. ЛИНЕЙНЫЕ И МУЛЬТИПЛИКАТИВНЫЕ КОНГРУЭНТНЫЕ ГЕНЕРАТОРЫ

Линейным конгруэнтным генератором (ЛКГ) с параметрами (x_0, a, c, N) называется программный генератор РРСП, порождающий псевдослучайную последовательность $x_1, x_2, \dots \in \mathcal{A}$, $\mathcal{A} = \{0, 1, \dots, N - 1\}$ с помощью рекуррентного соотношения

$$x_{t+1} = (ax_t + c) \bmod N, \quad t = 0, 1, \dots \quad (10.1)$$

Параметры генератора (10.1) имеют следующий смысл: $x_0 \in \mathcal{A}$ – начальное, или *стартовое, значение* (seed); $a \in \mathcal{A} \setminus \{0\}$ – ненулевой множитель; $c \in \mathcal{A}$ – приращение; $N \in \mathbb{N}$ – модуль, равный мощности алфавита \mathcal{A} .

Если приращение $c = 0$, то генератор (10.1) называется *многипликативным конгруэнтным генератором* (МКГ), а если $c \neq 0$, то *смешанным конгруэнтным генератором* (СКГ).

Перечислим свойства псевдослучайной последовательности, порождаемой ЛКГ.

Свойство C₁. Для общего члена последовательности (10.1) справедлива формула

$$x_t = \left(a^t x_0 + \frac{a^t - 1}{a - 1} c \right) \bmod N, \quad t \geq 1. \quad (10.2)$$

Свойство C₂. Найдется номер $\tau \in \mathcal{A}$, начиная с которого последовательность (10.1) «зацикливается» с периодом $T \leq N - \tau$.

Свойство C₃. Для любого $k \geq 2$ подпоследовательность $x_0, x_k, x_{2k}, x_{3k}, \dots \in \mathcal{A}$, полученная из псевдослучайной последовательности (10.1) удалением всех членов, не кратных k , оказывается псевдослучайной последовательностью, порожденной ЛКГ (10.1), с параметрами $(x_0, \tilde{a}, \tilde{c}, N)$, где

$$\tilde{a} = a^k \bmod N, \quad \tilde{c} = (c(a^k - 1)/(a - 1)) \bmod N.$$

Свойство C₄. Псевдослучайная последовательность (10.1), порожденная ЛКГ, достигает максимального значения периода $T_{\max} = N$ ($\tau = 0$) тогда и только тогда, когда выполнены следующие три условия:

- а) c и N – взаимно простые, т. е. $(c, N) = 1$;
- б) число $b = a - 1$ кратно p для любого простого числа $p < N$, являющегося делителем N ;
- в) число b кратно 4, если N кратно 4.

Свойство C₅. Для МКГ ($c = 0$), если x_0 и N – взаимно простые, a – первообразный элемент по модулю N , а $\varphi(N)$ – максимально возможный порядок по модулю N , то псевдослучайная последовательность имеет максимальный период $T_{\max} = \varphi(N)$.

Свойство C₆. Для МКГ, если $N = 10^q$, $q \geq 5$ и x_0 не кратно 2 или 5, то максимально возможное значение периода $T_{\max} = 5 \cdot 10^{q-2} = N/20$ псевдослучайной последовательности достигается тогда и только тогда, когда вычет $a \bmod 200$ принимает одно из 32 следующих «магических» значений:

$$3, 11, 13, 19, 21, 27, 29, 37, 53, 59, 61, 67, 69, 77, 83, 91, 109, 117, 123, \\ 131, 133, 139, 141, 147, 163, 171, 173, 179, 181, 187, 189, 197.$$

Свойство C₇. Для МКГ, если $N = 2^q$, $q \geq 4$, то максимально возможное значение периода $T_{\max} = 2^{q-2} = N/4$ псевдослучайной последовательности достигается, если $x_0 \geq 1$ – нечетно и вычет $a \bmod 8 \in \{3, 5\}$.

Свойство C₈. Для МКГ, если $x_0 \neq 0$, N – простое число и справедливо разложение на множители: $N - 1 = p_1^{m_1} \dots p_s^{m_s}$, где p_1, \dots, p_s – простые числа, а $m_1, \dots, m_s \in \mathbb{N}$, то максимально возможное значение периода $T_{\max} = N - 1$ псевдослучайной последовательности достигается для случая, когда

$$a^{(N-1)/p_j} \neq 1 \pmod{N}, \quad j = \overline{1, s}.$$

Свойство С₉. «Слабость» ЛКГ и МКГ заключается в том, что если рассматривать последовательные биграммы $(z_1^{(t)}, z_2^{(t)})$: $z_1^{(t)} = x_t$, $z_2^{(t)} = x_{t-1}$, то точки $z^t = (z_1^{(t)}, z_2^{(t)})$, $t = 1, 2, \dots$, на плоскости \mathbb{R}^2 будут лежать на прямых из семейства $z_2 = az_1 + c - kN$, $k = 0, 1, \dots$.

10.4. НЕЛИНЕЙНЫЕ КОНГРУЭНТНЫЕ ГЕНЕРАТОРЫ

Свойство С₉ линейного и мультиплекативного конгруэнтного генераторов псевдослучайных последовательностей (см. п. 10.3) представляет «слабость» этих генераторов и может активно использоваться для построения криптоатак в целях оценки параметров a , c , x_0 . Для устранения этого недостатка используют нелинейные конгруэнтные генераторы псевдослучайных последовательностей, среди которых известны: квадратичный конгруэнтный генератор; генератор Эйхенауэра – Лёна [91] с обращением; конгруэнтный генератор, использующий умножение с переносом; квадратичный генератор Ковэю; генератор «середины квадрата». Наибольшее распространение получили первые три подхода, описание которых приводится ниже.

10.4.1. Квадратичный конгруэнтный генератор

Этот алгоритм генерации псевдослучайной последовательности $x_t \in \mathcal{A} = \{0, 1, \dots, N-1\}$ определяется квадратичным рекуррентным соотношением

$$x_{t+1} = (dx_t^2 + ax_t + c) \bmod N, \quad t = 0, 1, \dots, \quad (10.3)$$

где $x_0, a, c, d \in \mathcal{A}$ – параметры генератора. Выбор этих параметров осуществляется на основе следующих двух свойств последовательности (10.3).

Свойство С₁. Квадратичная конгруэнтная последовательность (10.3) имеет наибольший период $T_{\max} = N$ тогда и только тогда, когда выполнены следующие условия:

- 1) c, N – взаимно простые числа;
- 2) $d, a - 1$ – кратны p , где p – любой нечетный простой делитель N ;
- 3) d – четное число, причем

$$d = \begin{cases} (a - 1) \bmod 4, & \text{если } N \text{ кратно } 4, \\ (a - 1) \bmod 2, & \text{если } N \text{ кратно } 2; \end{cases}$$

- 4) если N кратно 9, то либо $d \bmod 9 = 0$, либо $d \bmod 9 = 1$ и $cd \bmod 9 = 6$.

Свойство С₂. Если $N = 2^q$, $q \geq 2$, то наибольший период $T_{\max} = 2^q$ тогда и только тогда, когда c – нечетно, d – четно, a – нечетное число, удовлетворяющее соотношению $a = (d + 1) \bmod 4$.

10.4.2. Генератор Эйхенауэра – Лёна с обращением

Псевдослучайная нелинейная конгруэнтная последовательность Эйхенауэр – Лёна с обращением [91] определяется следующим нелинейным рекуррентным соотношением ($t = 0, 1, \dots$):

$$x_{t+1} = \begin{cases} (a \cdot x_t^{-1} + c) \bmod N, & \text{если } x_t \geq 1, \\ c, & \text{если } x_t = 0, \end{cases} \quad (10.4)$$

где $x_t^{-1} \in \mathcal{A}$ – обратный к x_t элемент по модулю N , т. е. $x_t x_t^{-1} \equiv 1 \pmod{N}$; $x_0, a, c \in \mathcal{A}$ – параметры генератора.

Свойство С₃. Если $N = 2^q$, a , x_0 – нечетны, c – четно, то генератор (10.4) имеет максимально возможный период $T_{\max} = 2^{q-1}$ тогда и только тогда, когда $a \equiv 1 \pmod{4}$, $c = 2 \pmod{4}$.

10.4.3. Конгруэнтный генератор, использующий умножение с переносом

При этом нелинейная конгруэнтная псевдослучайная последовательность определяется рекуррентным соотношением

$$x_{t+1} = (ax_t + c_t) \bmod N, \quad t = 0, 1, \dots, \quad (10.5)$$

где, в отличие от (10.3), «приращение» $c_t = c(x_{t-1}, x_{t-2}, \dots, x_0)$ изменяется во времени и зависит от указанных аргументов нелинейно:

$$c_t = \left\lfloor \frac{ax_{t-1} + c_{t-1}}{N} \right\rfloor. \quad (10.6)$$

Параметрами нелинейного конгруэнтного генератора (10.5), (10.6) являются x_0, c_0, a, N . Рекомендации по выбору этих параметров даны в работе [49].

10.5. ГЕНЕРАТОРЫ НА ОСНОВЕ РЕГИСТРОВ СДВИГА С ЛИНЕЙНОЙ, НЕЛИНЕЙНОЙ И СЛУЧАЙНОЙ ОБРАТНОЙ СВЯЗЬЮ

Регистр сдвига с обратной связью – один из самых распространенных элементов существующих и разрабатываемых криптографических генераторов.

Пусть алфавит \mathcal{A} выходной последовательности $\{x_t\}$ криптографического генератора является некоторым полем, \mathcal{A}^n есть n -мерное линейное векторное пространство над \mathcal{A} . Дискретное функциональное преобразование

$$A_f(\cdot) : \mathcal{A}^n \rightarrow \mathcal{A}^n$$

называется *преобразованием автономного регистра сдвига с одной обратной связью длины n* [8], если оно задается следующим образом:

$$\begin{aligned} A_f(s_0, s_1, \dots, s_{n-1}) = \\ = (s_1, s_2, \dots, s_{n-s} f(s_0, s_1, \dots, s_{n-1})), \quad (s_i) \in \mathcal{A}^n, \end{aligned} \quad (10.7)$$

где дискретная функция $f(\cdot) : \mathcal{A}^n \rightarrow \mathcal{A}$ называется *функцией обратной связи*.

Регистр сдвига A_f согласно 10.7 порождает *нелинейную рекурренту* над \mathcal{A} :

$$x_{t+n} = f(x_t, x_{t+1}, \dots, x_{t+n-1}). \quad (10.8)$$

Если $f(\cdot)$ – линейная функция:

$$f(s_0, s_1, \dots, s_{n-1}) = \sum_{j=0}^{n-1} a_j s_{t+j}, \quad (s_i) \in \mathcal{A}^n, \quad (10.9)$$

где $a_0, a_1, \dots, a_{n-1} \in \mathcal{A}$ – некоторые заданные коэффициенты, то получаем *регистр сдвига с линейной обратной связью*. Более подробно рассмотрим этот случай применительно к двоичному алфавиту $\mathcal{A} = \{0, 1\}$.

Регистром сдвига с линейной обратной связью (Linear Feedback Shift Register, сокращенно LFSR) называется логическое устройство, схема которого изображена на рис. 10.2 [104].

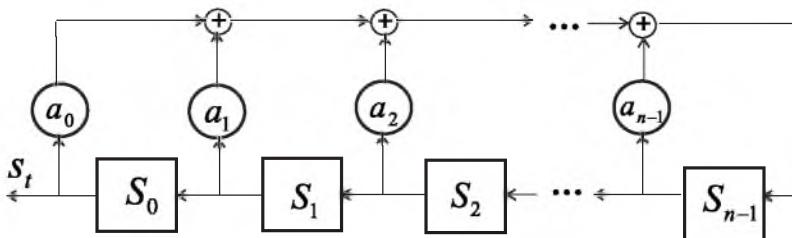


Рис. 10.2. Блок-схема LFSR

LFSR состоит из n ячеек памяти, двоичные состояния которых в момент времени $t = 0, 1, \dots$ характеризуются значениями $S_0(t), S_1(t), \dots, S_{n-1}(t) \in \mathcal{A} = \{0, 1\}$. Выходы ячеек памяти связаны не только последовательно друг с другом, но и с сумматорами \oplus в соответствии с коэффициентами передачи $a_0, a_1, \dots, a_n \in \mathcal{A}$: если $a_i = 1$, то значение $S_i(t)$ i -й ячейки передается на один из входов i -го сумматора; если же $a_i = 0$, то такая передача отсутствует; полагается $a_0 \equiv 1$. Состояние LFSR в момент времени t задается двоичным n -вектором-столбцом $S(t) = (S_{n-1}(t), S_{n-2}(t), \dots, S_0(t))'$.

Содержание ячеек LFSR с течением времени изменяется следующим образом, определяя тем самым динамику состояний LFSR:

$$S_i(t+1) = \begin{cases} S_{i+1}(t), & \text{если } i \in \{0, 1, \dots, n-2\}, \\ \sum_{j=0}^{n-1} a_j S_j(t), & \text{если } i = n-1, \end{cases} \quad (10.10)$$

или в матричном виде:

$$S(t+1) = AS(t), \quad A = \begin{pmatrix} a_0 & a_1 & a_2 & \dots & \vdots & a_{n-1} \\ & & & & \vdots & 0 \\ I_{n-1} & & & & \vdots & \vdots \\ & & & & \vdots & 0 \end{pmatrix}, \quad (10.11)$$

где I_{n-1} – единичная $(n-1) \times (n-1)$ -матрица, $t = 0, 1, \dots$.

Текущие значения нулевой ячейки регистра используются в качестве элементов порождаемой LFSR двоичной псевдослучайной последовательности $s_t = S_0(t)$ (см. рис. 10.2), которая с учетом (10.10) удовлетворяет линейному рекуррентному соотношению, являющемуся частным случаем (10.7), (10.8):

$$s_{t+n} = \sum_{j=0}^{n-1} a_j \times s_{t+j}, \quad t = 0, 1, \dots. \quad (10.12)$$

Модель (10.12) – частный случай модели (10.7) линейной рекурренты над полем \mathbb{F}_2 , поэтому коэффициенты $\{a_j\}$ выбираются согласно методике, приведенной в п. 3.17.

Опишем генератор Таусвортса, основанный на LFSR. Он аналогичен (10.10)–(10.12) и определяется следующими соотношениями:

$$A = (I_n + (L')^p)(I_n + L^q);$$

$$S(t+1) = AS(t), \quad s_t = S_0(t), \quad t = 0, 1, \dots, \quad (10.13)$$

где p, q – некоторые заданные натуральные числа (параметры алгоритма); L – $(n \times n)$ -двоичная матрица, все единицы которой расположены лишь под главной диагональю. Преобразование $\tilde{S} = L^q S$ сдвигает все компоненты вектора S «вниз» на q позиций, заполняя свободившиеся позиции нулями; точно так же $\tilde{S} = (L')^p S'$ осуществляет сдвиг «вверх» на p позиций.

Параметры p, q выбираются таким образом, чтобы матрица A имела порядок $2^n - 1$ в группе невырожденных $(n \times n)$ -матриц; при этом последовательность двоичных векторов $\{S(t)\}$, определяемых (10.13), имеет максимально возможный период $T_{\max} = 2^n - 1$. Это условие выполняется, например, если:

$$\begin{aligned} n &= 31, p = 13, q = 18 \text{ или } p = 18, q = 13; \\ n &= 32, p = 15, q = 17 \text{ или } p = 17, q = 15. \end{aligned} \quad (10.14)$$

В качестве примеров эффективно реализуемых на компьютере алгоритмов генерации вида (10.10) приведем четыре генератора псевдослучайных последовательностей с периодом $T_{\max} \approx 2^{96}$ [124]:

$$\begin{aligned}x_{t+1} &= (1176x_t + 1476x_{t-1} + 1776x_{t-2}) \bmod (2^{32} - 5); \\x_{t+1} &= 2^{13}(x_t + x_{t-1} + x_{t-2}) \bmod (2^{32} - 5); \\x_{t+1} &= (1995x_t + 1998x_{t-1} + 2001x_{t-2}) \bmod (2^{35} - 849); \\x_{t+1} &= 2^{19}(x_t + x_{t-1} + x_{t-2}) \bmod (2^{35} - 1629).\end{aligned}\quad (10.15)$$

Для повышения качества псевдослучайных последовательностей в линейную рекурренту (10.7) вводится нелинейное «приращение» c_t (как и в конгруэнтном генераторе, использующем умножение с переносом, см. п. 10.4):

$$x_{t+1} = (a_{n-1}x_t + a_{n-2}x_{t-1} + \dots + a_0x_{t-n+1} + c_t) \bmod p, \quad (10.16)$$

где «приращение-перенос»

$$c_t = c(x_{t-1}, x_{t-2}, \dots, x_0, x_{-1}, \dots, x_{-n+1}) \in \mathcal{A}$$

зависит от своих аргументов нелинейно согласно рекуррентному соотношению

$$c_t = \lfloor (a_{n-1}x_t + a_{n-2}x_{t-1} + \dots + a_0x_{t-n+1} + c_{t-1})/p \rfloor. \quad (10.17)$$

Примером рекуррентного генератора с переносом (10.16), (10.17) является генератор с периодом $T_{\max} \approx 2^{158}$ [124]:

$$x_{t+1} = (5115x_t + 1776x_{t-1} + 1492x_{t-2} + 2111111111x_{t-3} + c_t) \bmod 2^{32}. \quad (10.18)$$

Соотношения (10.16), (10.17) фактически определяют нелинейное рекуррентное соотношение вида (10.8). Другой пример генератора, основанного на нелинейной рекурренте (10.8) – генератор Фибоначчи.

Общий вид рекуррентного соотношения, определяющего генератор Фибоначчи, задается уравнением

$$x_t = x_{t-r} \diamond x_{t-s}, \quad t = r, r+1, r+2, \dots, \quad (10.19)$$

где $r, s \in \mathbb{N}$ ($r > s$) – параметры генератора; \diamond – символ бинарного отношения:

$$\diamond \in \{+, -, \cdot, \oplus\}.$$

В случае «+» или «-» $\{x_t\}$ – целые числа ($\bmod 2^k$) для некоторого заданного $k \in \mathbb{N}$; в случае «·» $\{x_t\}$ – нечетные числа ($\bmod 2^k$); в случае « \oplus » элемент $x_t \in V_k$ представляет собой двоичный k -вектор и действие \oplus выполняется покомпонентно.

Свойства псевдослучайной последовательности Фибоначчи (10.19) и методика выбора параметров r, s, k излагаются в [49].

Как уже отмечалось, теория регистров сдвига с линейной обратной связью базируется на глубокой теории рекуррент (см. гл. 3). В то же время общей теории регистров сдвига с нелинейной обратной связью пока не создано. В литературе исследуются лишь некоторые специальные классы нелинейностей. Упомянем класс регистров сдвига с «динамическим изменением закона рекурсии» [2]. Пусть имеется $L \geq 2$ вариантов задания вектора коэффициентов $a = (a_0, a_1, \dots, a_{n-1})' \in V_n$ в соотношении (10.10):

$$a^{(1)} = (a_i^{(1)}) \in V_n, \quad a^{(2)} = (a_i^{(2)}) \in V_n, \quad \dots, \quad a^{(L)} = (a_i^{(L)}) \in V_n -$$

и задана некоторая индексная последовательность (детерминированная либо псевдослучайная)

$$l_t \in \{1, 2, \dots, L\}, \quad t = 0, 1, \dots$$

В момент времени t значение индексной последовательности l_t определяет вектор коэффициентов обратной связи в соотношении (10.10):

$$S_i(t+1) = \begin{cases} S_{i+1}(t), & \text{если } i \in \{0, 1, \dots, n-2\}, \\ \sum_{j=0}^{n-1} a_j^{(l_t)} S_j(t), & \text{если } i = n-1. \end{cases}$$

В [2] исследованы свойства такого нелинейного регистра, когда $L = 2$,

$$l_t = \begin{cases} 2, & \text{если } t - \text{нечетно}, \\ 1, & \text{если } t - \text{четно}. \end{cases}$$

10.6. КРИПТОСТОЙКИЕ ГЕНЕРАТОРЫ НА ОСНОВЕ ОДНОСТОРОННИХ ФУНКЦИЙ

Для повышения стойкости алгоритмов генерации псевдослучайных последовательностей к криptoанализу в настоящее время предлагается [133] синтезировать алгоритмы на основе известных в криптографии односторонних функций (см. гл. 8). Характерное свойство односторонних (*one-way*) функций состоит в том, что для вычисления значения функции по заданному значению аргумента существует полиномиально-сложный алгоритм, в то время как для вычисления аргумента по заданному значению функции полиномиально-сложного алгоритма не существует (или он неизвестен). Доказательство свойства односторонности функций является трудной математической задачей, поэтому в настоящее время в криптосистемах часто используются «кандидаты в односторонние функции», для которых пока показано лишь, что в настоящее время не известны полиномиально-сложные

алгоритмы вычисления обратной функции. Примерами таких «кандидатов» являются некоторые известные криптоалгоритмы (например, DES) и хэш-функции (например, SHA-1).

10.6.1. Генератор ANSI X9.17

Он представляет собой национальный стандартный алгоритм США для генерации двоичной псевдослучайной последовательности, входящей в FIPS (USA Federal Information Processing Standard). В этом генераторе в качестве «кандидата» односторонней функции используется так называемый «тройной DES» с двумя ключами $K_1, K_2 \in V_{64}$ – алгоритм шифрования вида

$$F_K = E_{K_1} D_{K_2} E_{K_1},$$

где $K = (K_1 \parallel K_2) \in V_{128}$ – составной 128-битовый ключ; E_{K_1} – алгоритм шифрования DES с ключом K_1 ; D_{K_2} – алгоритм расшифрования DES с ключом $K_2 \neq K_1$.

Входными данными алгоритма ANSI X9.17 являются: некоторое случайное (и конфиденциальное) 64-битовое стартовое слово $s \in V_{64}$; 128-битовый составной ключ $K \in V_{128}$; m – количество 64-битовых двоичных слов, которые необходимо получить на выходе генератора.

Выходными данными являются m 64-битовых двоичных слов $x_1, x_2, \dots, x_m \in V_{64}$.

Алгоритм генерации состоит из пяти шагов.

Шаг 1. Фиксируется 64-битовое представление $d \in V_{64}$ даты и времени при обращении к программе генерации и вычисляется вспомогательное 64-битовое двоичное слово

$$I = F_K(d).$$

Шаг 2. Для $i = \overline{1, m}$ повторяются шаги 3, 4.

Шаг 3. Вычисляется значение i -го выходного слова:

$$x_i = F_K(I \oplus s).$$

Шаг 4. Вычисляется новое значение параметра s :

$$s = F_K(x_i \oplus I).$$

Шаг 5. Формируется выходная последовательность m слов $x_1, x_2, \dots, x_m \in V_{64}$ либо двоичная последовательность из $64m$ битов:

$$X = (x_1 \parallel x_2 \parallel \dots \parallel x_m) \in V_{64m}.$$

10.6.2. Генераторы FIPS-186

Они также являются национальным стандартом США и предназначены для генерации конфиденциальных параметров и ключевой информации для национального стандартного алгоритма электронной цифровой подписи DSA. В качестве «кандидата» односторонней функции используется алгоритм шифрования DES или алгоритм хэширования SHA-1; таким образом, имеются два варианта односторонней функции G_1 и G_2 (алгоритмы их вычисления приведены ниже).

Входными данными алгоритма генерации FIPS-186 являются целое число t и 160-битовое простое число q .

Выходные данные: последовательность m псевдослучайных чисел $x_1, x_2, \dots, x_m \in \{0, 1, \dots, q - 1\}$, которые рекомендовано использовать как ключи в стандарте электронной цифровой подписи DSA.

Алгоритм генерации состоит из девяти шагов.

Шаг 1. Если в алгоритме используется односторонняя функция G_1 , то задать $b = 160$; если же G_2 , то задать произвольное число b : $160 \leq b \leq 512$.

Шаг 2. Сгенерировать неким образом случайное (и конфиденциальное) b -битовое стартовое значение $s \in V_b$.

Шаг 3. Задать 160-битовое «магическое» вспомогательное слово (в шестнадцатеричной записи):

$$t = 67452301 \text{ EFCDAB89 } 98BADCCE \text{ 10325476 C3D2E1F0}_{16}.$$

Шаг 4. Для $i = \overline{1, m}$ выполнить шаги 5–8.

Шаг 5. По усмотрению пользователя: либо произвольно задать b -битовое число $y_i \in V_b$, либо положить его равным нулю ($y_i = 0 \in V_b$).

Шаг 6. Вычислить $z_i = (s + y_i) \bmod 2^b \in V_b$.

Шаг 7. Вычислить $x_i = G(t, z_i) \bmod q \in \mathbb{F}_q$, где G – используемый вариант односторонней функции, т. е. $G = G_1$ либо $G = G_2$; алгоритмы вычисления значений G_1, G_2 приведены ниже.

Шаг 8. Вычислить $s = (1 + s + x_i) \bmod 2^b \in V_b$.

Шаг 9. Формируется псевдослучайная последовательность

$$x_1, x_2, \dots, x_m \in \mathbb{F}_q.$$

Опишем односторонние функции G_1 и G_2 .

Алгоритм вычисления значений функции G_1 . *Входные данные:* два 160-битовых слова $t, c \in V_{160}$.

Выходные данные: 160-битовое слово $G_1(t, c) \in V_{160}$.

Алгоритм вычисления значений G_1 состоит из одиннадцати шагов.

Шаг 1. Разбить слово t на 5 блоков по 32 бита: $t = t_0 \parallel t_1 \parallel t_2 \parallel t_3 \parallel t_4$.

Шаг 2. Аналогично разбить слово: $c = c_0 \parallel c_1 \parallel c_2 \parallel c_3 \parallel c_4$.

Шаг 3. Для $i = \overline{0, 4}$ вычислить $u_i = t_i \oplus c_i$.

Шаг 4. Для $i = \overline{0, 4}$ выполнить шаги 5–9.

Шаг 5. Вычислить $b_1 = c_{(i+4) \bmod 5}$, $b_2 = c_{(i+3) \bmod 5}$.

Шаг 6. Вычислить $a_1 = u_2$, $a_2 = u_{(i+1) \bmod 5} \oplus u_{(i+4) \bmod 5}$.

Шаг 7. Вычислить $A = (a_1 \parallel a_2)$, $B = (\tilde{b}_1 \parallel b_2)$, где $\tilde{b}_1 \in V_{24}$ обозначает 24 младших бита 32-битового слова b_1 .

Шаг 8. Осуществить DES-шифрование слова A с помощью 56-битового ключа B : $y_i = E_B(A)$.

Шаг 9. Разбить полученное после шифрования слово $y_i \in V_{64}$ на два 32-битовых слова: $y_i = L_i \parallel R_i$.

Шаг 10. Для $i = \overline{0, 4}$ вычислить $z_i + L_i \oplus R_{(i+2) \bmod 5} \oplus L_{(i+3) \bmod 5}$.

Шаг 11. Выходное 160-битовое слово является конкатенацией:

$$G(t, c) = z_0 \parallel z_1 \parallel z_2 \parallel z_3 \parallel z_4 \in V_{160}.$$

Алгоритм вычисления значений функции G_2 . Входные данные: 160-битовое слово t и b -битовое слово c , где $160 \leq b \leq 512$.

Выходные данные: 160-битовое слово $G_2(t, c) \in V_{160}$.

Алгоритм вычисления значений функции G_2 состоит из пяти шагов.

Шаг 1. Разбить слово t на 32-битовые слова: $t = H_1 \parallel H_2 \parallel H_3 \parallel H_4 \parallel H_5$.

Шаг 2. Дополнить слово c $512 - b$ нулями, чтобы получить 512-битовое слово: $U = c \parallel \mathbf{0}^{512-b}$.

Шаг 3. Разбить слово U на шестнадцать 32-битовых слов: $U = u_0 \parallel u_1 \parallel \dots \parallel u_{15}$ и задать $m = 1$.

Шаг 4. Выполнить шаг 4 алгоритма SHA-1, изменяющий $\{H_i\}$.

Шаг 5. Выходное слово является конкатенацией:

$$G_2(t, c) = H_1 \parallel H_2 \parallel H_3 \parallel H_4 \parallel H_5.$$

10.6.3. Генератор Yarrow-160

Генератор псевдослучайных двоичных слов Yarrow-160 (по-английски yarrow означает тысячелистник – название растения) предложен Дж. Келси, Б. Шнайером и Н. Фергюсоном [116] как криптографический генератор псевдослучайных чисел. Он использует «тройной DES» с «тройным ключом» и американский стандарт хэш-функции SHA-1.

Очередное псевдослучайное число $x \in V_{64}$ определяется по правилам:

$$C \leftarrow (C + 1) \bmod 2^{64}, \quad x \leftarrow E_{K_3} E_{K_2}^{-1} E_{K_1}(C),$$

где C – 64-битовый счетчик; E_{K_i} – алгоритм DES-шифрования с ключом $K_i \in V_{56}$.

В алгоритме Yarrow предусмотрена специальная схема обновления ключа K и значения счетчика C с использованием хэш-функции SHA-1 и некоторых вспомогательных «аккумуляторов случайности».

10.7. КРИПТОСТОЙКИЕ ГЕНЕРАТОРЫ, ОСНОВАННЫЕ НА ПРОБЛЕМАХ ТЕОРИИ ЧИСЕЛ

Стойкость генераторов псевдослучайных последовательностей, рассматриваемых в данном пункте, основывается на неразрешимости с полиномиальной сложностью (на данный момент) некоторых известных *проблем теории чисел: факторизации больших чисел и дискретного логарифмирования*.

10.7.1. RSA-алгоритм генерации псевдослучайных последовательностей

Этот алгоритм основывается на том факте, что в настоящее время криptoанализ RSA не может быть осуществлен с полиномиальной сложностью.

RSA-алгоритм генерации двоичной псевдослучайной последовательности $x_1, x_2, \dots, x_n \in \mathcal{A}$ состоит из шести шагов.

Шаг 1. Как и в RSA-крипtosистеме, генерируются различные достаточно большие простые числа p, q и вычисляются числа $N = p q, \varphi = (p - 1)(q - 1)$. Выбирается случайное целое число $k, 1 < k < \varphi$, взаимно простое с φ так, что $(k, \varphi) = 1$.

Шаг 2. Выбрать случайное целое – стартовое значение – $u_0 \in \{1, 2, \dots, N - 1\}$.

Шаг 3. Генерируется псевдослучайная последовательность длиной $K \times n$. Для $i = \overline{1, n}$ выполнить шаги 4, 5.

Шаг 4. Вычислить $u_i = u_{i-1}^k \bmod N \in \{0, 1, \dots, N - 1\}$.

Шаг 5. Вычислить $x_i \in \mathcal{A} = \{0, 1\}$ – самый младший бит числа u_i в двоичном представлении.

Шаг 6. Сформировать выходную последовательность x_1, x_2, \dots, x_n .

Недостатком RSA-алгоритма является его невысокое быстродействие при реализации на универсальных компьютерах, вызванное большими затратами машинного времени при выполнении модулярного умножения на шаге 4. Для повышения быстродействия на шаге 5 можно выделять сразу несколько младших битов. Эта идея используется в представленной ниже *модификации Микали – Шнорра RSA-алгоритма*.

Шаг 1. Как и в RSA-крипtosистеме, генерируются различные достаточно большие простые числа p, q и вычисляются $N = p q, \varphi = (p - 1)(q - 1)$. Вычисляется битовая длина $m = \lceil \log_2 N \rceil$ числа N . Выбирается целое $k, 1 < k < \varphi$ так, что $(k, \varphi) = 1, 80k \leq m$. Вычисляются $K = \lfloor m(1 - 2/k) \rfloor, r = m - K$.

Шаг 2. Выбрать стартовое число u_0 , содержащее в двоичном представлении r значимых битов: $u_0 \in \{0, 1, \dots, 2^r - 1\}$.

Шаг 3. Генерируется псевдослучайная последовательность длиной $K \times n$. Для $i = \overline{1, n}$ выполнить шаги 4–6.

Шаг 4. Вычислить $y_i = u_{i-1}^k \bmod N$.

Шаг 5. Вычислить u_i как число, двоичное представление которого образовано r старшими битами двоичного представления числа y_i .

Шаг 6. Вычислить $x_i \in V_K$ как двоичное слово, образованное K младшими битами двоичного представления числа y_i .

Шаг 7. Сформировать выходную двоичную последовательность $x_1, x_2, \dots, x_n \in V_K$.

Отметим, что условие криптостойкости алгоритма Микали – Шнорра является более жестким, чем для RSA-алгоритма, и состоит в следующем: вероятностное распределение случайной величины $y = x^k \bmod N$ для r -битовой случайной величины x должно быть неотличимо полиномиально-сложными статистическими тестами от равномерного распределения на множестве $\{0, 1, \dots, N - 1\}$.

10.7.2. BBS (Blum – Blum – Shub)-алгоритм генерации псевдослучайных последовательностей

Этот алгоритм генерации двоичной псевдослучайной последовательности $x_1, x_2, \dots, x_n \in \mathcal{A} = \{0, 1\}$ основывается на проблеме факторизации больших чисел. Он используется в криптосистеме Блюма – Голдвассера с открытым ключом и состоит из шести шагов.

Шаг 1. Генерируются два достаточно больших секретных случайных (и различных) простых числа $p, q \equiv 3 \pmod{4}$ и вычисляется $N = pq$.

Шаг 2. Выбрать случайное стартовое целое число $s \in \{1, 2, \dots, N - 1\}$ так, что $(s, N) = 1$. Вычислить $u_0 = s^2 \bmod N$.

Шаг 3. Для $i = \overline{1, n}$ выполнить шаги 4, 5.

Шаг 4. Вычислить $u_i = u_{i-1}^2 \bmod N$.

Шаг 5. Вычислить $x_i \in \mathcal{A}$ как самый младший бит двоичного представления числа u_i .

Шаг 6. Сформировать выходную последовательность $x_1, x_2, \dots, x_n \in \mathcal{A}$.

Отметим, что BBS-алгоритм допускает модификацию, подобную модификации Микали – Шнорра.

10.8. МЕТОДЫ «УЛУЧШЕНИЯ» СВОЙСТВ ЭЛЕМЕНТАРНЫХ ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Пусть $\xi_1, \xi_2, \dots \in \mathcal{A}$ – некоторая двоичная псевдослучайная последовательность, генерированная одним из простейших методов и называемая поэтому (в данном пункте) элементарной. Для того чтобы построить псевдослучайную последовательность со свойствами, более близкими к свойствам

РРСП, чем элементарная последовательность, осуществим функциональное преобразование $\{\xi_i\}$:

$$x_1 = f_1(\xi_1, \xi_2, \dots), \quad x_2 = f_2(\xi_1, \xi_2, \dots), \quad \dots, \quad (10.20)$$

где $f_1(\cdot), f_2(\cdot), \dots$ – некоторые функционалы, задающие отображение $\mathcal{A}^\infty \rightarrow \mathcal{A}$.

Функционалы $f_1(\cdot), f_2(\cdot), \dots$ следует подбирать так, чтобы преобразованная последовательность $\{x_k\}$ имела вероятностное распределение, более близкое к распределению РРСП, чем распределение $\{\xi_i\}$.

Выбирая различные функционалы $f_1(\cdot), f_2(\cdot), \dots$ и метрики в пространстве вероятностных распределений, можно разработать множество методов и алгоритмов «улучшения» элементарных псевдослучайных последовательностей.

Проиллюстрируем эту идею на простейшем примере (алгоритм симметризации псевдослучайных последовательностей). Пусть в элементарной последовательности $\{\xi_i\}$ нарушено условие равновероятности символов «0» и «1»:

$$\mathbf{P}\{\xi_i = 1\} = p, \quad \mathbf{P}\{\xi_i = 0\} = q = 1 - p, \quad p \neq \frac{1}{2}. \quad (10.21)$$

В целях преодоления этого отклонения от свойств РРСП осуществим следующее функциональное преобразование вида (10.20):

$$\eta_j = \begin{cases} 0, & \text{если } \xi_{2j-1} = 0, \xi_{2j} = 1, \\ 1, & \text{если } \xi_{2j-1} = 1, \xi_{2j} = 0, \quad j = 1, 2, \dots . \\ 2 & \text{в противном случае,} \end{cases} \quad (10.22)$$

Затем из этой последовательности $\{\eta_j\}$ выходная двоичная последовательность $\{x_k\}$ получается прореживанием: отбрасываются значения $\eta_j \notin \mathcal{A} = \{0, 1\}$, т. е. те значения, которые совпадают с символом «2».

Из (10.21) и (10.22) следует, что

$$\begin{aligned} \pi_0 &= \mathbf{P}\{\eta_j = 0\} = \mathbf{P}\{\xi_{2j-1} = 0, \xi_{2j} = 1\} = p(1-p); \\ \pi_1 &= \mathbf{P}\{\eta_j = 1\} = \mathbf{P}\{\xi_{2j-1} = 1, \xi_{2j} = 0\} = p(1-p); \\ \mathbf{P}\{\eta_j = 2\} &= 1 - 2p(1-p). \end{aligned} \quad (10.23)$$

Из (10.23) видно, что $\pi_0 = \pi_1$, поэтому после прореживания получаем двоичную псевдослучайную последовательность с равновероятными значениями:

$$\mathbf{P}\{x_k = 0\} = \mathbf{P}\{x_k = 1\} = \frac{1}{2}. \quad (10.24)$$

Заметим, что при получении (10.23), (10.24) использовалось предположение о попарной независимости элементов исходной последовательности $\{\xi_i\}$.

Отметим, что доля (в процентах) отброшенных значений в последовательности $\{\eta_j\}$ равна

$$k = (1 - 2p(1 - p)) 100$$

и тем больше, чем больше p отличается от $1/2$. Это позволяет вычислить средний коэффициент использования исходной последовательности для получения одного элемента выходной последовательности.

Проиллюстрированная идея «улучшения» свойств псевдослучайных последовательностей активно используется на практике при конструировании криптографических генераторов. При этом в качестве генераторов элементарных псевдослучайных последовательностей применяются регистры сдвига с линейной обратной связью, а при конструировании используются два основных метода: 1) метод фильтрующих преобразований, порождающий *фильтрующие генераторы*; 2) метод комбинирования, порождающий *комбинирующие генераторы*. Выделяются еще два специальных класса криптографических генераторов: *генераторы с неравномерным движением* (см. п. 10.11.2 и обзор [37]); *генераторы с дополнительной памятью* (см. п. 10.4.3, п. 10.5).

10.9. ФИЛЬТРУЮЩИЕ ГЕНЕРАТОРЫ

Фильтрующий генератор состоит из регистра сдвига некоторой длины n с линейной обратной связью, порождающего линейную рекуррентную последовательность x_t над полем $A = GF(q)$ из q элементов и нелинейного фильтрующего преобразования $y = f(\cdot): A^n \rightarrow A$ (см. рис. 10.3).

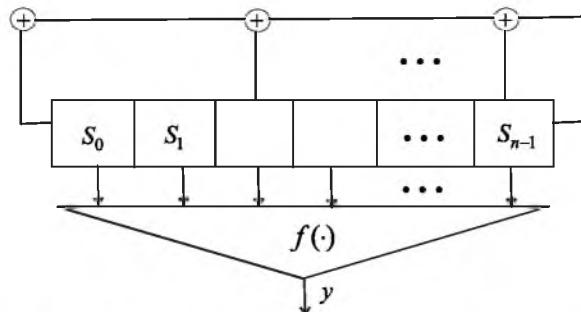


Рис. 10.3. Блок-схема фильтрующего генератора

Параметрами фильтрующего генератора (которые могут использоваться в качестве ключа) являются: примитивный многочлен $F(\cdot)$ степени n над полем $GF(q)$; начальное состояние (заполнение) регистра сдвига; функция $f(\cdot)$.

Фильтрующая функция $f(\cdot)$ над $GF(q)$ задается некоторым полиномом с коэффициентами из этого поля [2]:

$$f(x_1, \dots, x_n) = \sum_{\substack{1 \leq i_1 < \dots < i_s \leq n, \\ k_1, \dots, k_s \in \{0, 1, \dots, q-1\}, \\ 0 \leq s \leq n}} C_{i_1, \dots, i_s}^{k_1, \dots, k_s} x_{i_1}^{k_1} \cdot \dots \cdot x_{i_s}^{k_s}, \quad x_1, \dots, x_n \in A.$$

Тогда в силу преобразования, схема которого приведена на рис. 10.3, выходная последовательность фильтрующего генератора имеет вид

$$\begin{aligned} y_t &= f(x_t, x_{t+1}, \dots, x_{t+n-1}) = \\ &= \sum_{\substack{1 \leq i_1 < \dots < i_s \leq n, \\ k_1, \dots, k_s \in \{0, 1, \dots, q-1\}, \\ 0 \leq s \leq n}} C_{i_1, \dots, i_s}^{k_1, \dots, k_s} x_{t+i_1-1}^{k_1} \cdot \dots \cdot x_{t+i_s-1}^{k_s}, \quad t = 1, 2, \dots. \end{aligned} \quad (10.25)$$

Линейная сложность выходной последовательности (10.25) допускает следующую оценку сверху [2]:

$$\Lambda(\{y_t\}) \leq \Lambda_+ = \sum_{k=0}^{\min(n, \lfloor r/q \rfloor)} (-1)^k C_n^k C_{n+r-kq-1}^{n-1},$$

где r – степень фильтрующего полинома $f(\cdot)$. Из этого соотношения видно, что для увеличения верхней границы сложности Λ_+ необходимо увеличивать степень нелинейности r . В [43] для линейной сложности получены оценки снизу для некоторых нелинейностей $f(\cdot)$ специального вида. Заметим еще, что если $F(\cdot)$ – примитивный многочлен, а $f(\cdot)$ – сбалансированная функция, то для любого $b \in \mathcal{A}$ мощность

$$|x = (x_i) \in \mathcal{A}^f(x) = b| = q^{n-1},$$

то $\{y_t\}$ имеет период $q^n - 1$, такой же, как у $\{x_t\}$. Исследованию фильтрующих генераторов посвящены работы [8, 30].

10.10. КОМБИНИРОВАНИЕ АЛГОРИТМОВ ГЕНЕРАЦИИ МЕТОДОМ МАКЛАРЕНА – МАРСАЛЬИ

Перейдем теперь к рассмотрению метода комбинирования простейших генераторов.

Пусть имеется два простейших генератора псевдослучайных последовательностей: G_1 и G_2 . Генератор G_1 порождает «элементарную» последовательность над алфавитом мощности N :

$$x_0, x_1, x_2, \dots \in \mathcal{A}(N) = \{0, 1, \dots, N-1\},$$

а генератор G_2 – над алфавитом мощности K :

$$y_0, y_1, y_2, \dots \in \mathcal{A}(K) = \{0, 1, \dots, K - 1\}.$$

Пусть имеется вспомогательная таблица

$$T = \{T(0), T(1), \dots, T(K - 1)\}$$

из K целых чисел (память из K ячеек).

Метод Макларена – Марсалы комбинирования последовательностей $\{x_i\}$, $\{y_i\}$ для получения выходной псевдослучайной последовательности $\{z_k\}$ состоит в следующем [122, 123, 125, 126, 127, 128]. Сначала T -таблица заполняется K первыми членами последовательности $\{x_i\}$:

$$T(i) \leftarrow x_i, \quad i = \overline{0, K - 1}.$$

Элементы выходной последовательности вычисляются следующим образом:

$$s \leftarrow y_k, \quad z_k = T(s), \quad T(s) \leftarrow x_{K+k}, \quad k = 0, 1, \dots .$$

Таким образом, генератор G_2 осуществляет «случайный» выбор из T -таблицы, а также ее «случайное» заполнение «случайными» числами, порождаемыми генератором G_1 .

Метод комбинирования Макларена – Марсалы позволяет ослабить зависимость между членами $\{z_k\}$ и увеличить период псевдослучайной последовательности.

10.11. КОМБИНИРОВАНИЕ LFSR-ГЕНЕРАТОРОВ

LFSR-генераторы часто используются в качестве генераторов элементарных псевдослучайных последовательностей и применяются для комбинирования генераторов [150]. Прежде всего отметим, что LFSR-генераторы можно использовать в качестве G_1 , G_2 в генераторе Макларена – Марсалы. Рассмотрим еще ряд способов комбинирования LFSR-генераторов.

10.11.1. Полиномиальное комбинирование элементарных последовательностей

Рассмотрим общую модель комбинирования $M \geq 2$ элементарных псевдослучайных последовательностей, порожденных M генераторами LFSR (см. п. 10.5): $\text{LFSR}_1, \dots, \text{LFSR}_M$ (рис. 10.4). Здесь $x_{lt} \in \mathcal{A} = \{0, 1\}$ – элемент двоичной псевдослучайной последовательности, порождаемой l -м LFSR-генератором в момент времени $t = 1, 2, \dots$ ($l = 1, M$), $y = F(x_1, \dots, x_M) : \mathcal{A}^M \rightarrow \mathcal{A}$ – заданная двоичная функция M двоичных переменных, $y_t \in \mathcal{A}$ – элемент выходной двоичной псевдослучайной последовательности в момент времени t .

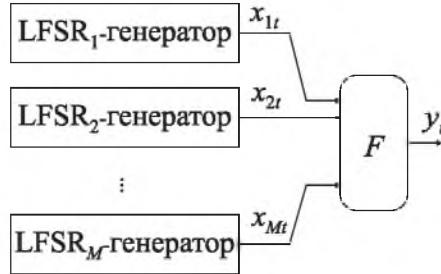


Рис. 10.4. Общая модель комбинирования LFSR-генераторов

Будем рассматривать функцию $F(\cdot)$ общего полиномиального вида:

$$\begin{aligned} y = F(x) = & \left(a_0 + \sum_{1 \leq i \leq M} a_i x_i + \sum_{1 \leq i < j \leq M} a_{ij} x_i x_j + \dots + \right. \\ & \left. + a_{12\dots M} x_1 \dots x_M \right) \bmod 2, \end{aligned} \quad (10.26)$$

где $a_0, \{a_i\}, \{a_{ij}\}, \dots, a_{12\dots M} \in \mathcal{A}$ – заданные двоичные коэффициенты.

Определим аналогично (10.26) полином степени M над полем действительных чисел \mathbb{R} :

$$\begin{aligned} F^*(x) = & a_0 + \sum_{1 \leq i \leq M} a_i x_i + \sum_{1 \leq i < j \leq M} a_{ij} x_i x_j + \dots + \\ & + a_{12\dots M} x_1 \dots x_M. \end{aligned} \quad (10.27)$$

Исследуем влияние функции комбинирования $F(\cdot)$ на свойства выходной последовательности. Поскольку каждая из последовательностей $\{x_{1t}\}, \{x_{2t}\}, \dots, \{x_{Mt}\}$, порождаемая LFSR-генератором, периодична, то и выходная двоичная последовательность

$$y_t = F(x_{1t}, x_{2t}, \dots, x_{Mt}), \quad t = 1, 2, \dots, \quad (10.28)$$

также периодична. Известно, что всякую периодическую последовательность $\{y_t\}$ можно представить как порожденную некоторым LFSR-генератором. Наименьший порядок этого LFSR принято называть *линейной сложностью* последовательности $\{y_t\}$ и обозначать $\Lambda(\{y_t\})$. Справедливо следующее утверждение, связывающее линейные сложности входных последовательностей $\{x_{1t}\}, \dots, \{x_{Mt}\}$ и выходной последовательности $\{y_t\}$ [150].

Теорема 10.1. Пусть M двоичных LFSR-последовательностей максимальной длины (соответствующих примитивным многочленам) комбинируются согласно (10.26). Если все стартовые значения LFSR – ненулевые, все M порядков примитивных многочленов L_1, \dots, L_M различны и

$L_j > 2$ ($j = \overline{1, M}$), то выходная последовательность (10.28) имеет линейную сложность

$$\Lambda(\{y_t\}) = F^*(L_1, L_2, \dots, L_M), \quad (10.29)$$

где $F^*(\cdot)$ определяется полиномом (10.27).

Следствие 10.1. Если комбинирование LFSR-генераторов осуществляется сложением по модулю 2 их выходных последовательностей, то

$$\Lambda(\{y_t\}) = \sum_{i=1}^M L_i.$$

Чем выше линейная сложность $L\{y_t\}$, тем более криптостойким является генератор псевдослучайной последовательности. Отметим, однако, что при подборе функции комбинирования (10.26) необходимо учитывать еще требование сбалансированности функции $F(\cdot)$:

$$W(F) = \sum_{x_1, \dots, x_M \in \{0, 1\}} F(x_1, \dots, x_M) = 2^{M-1}. \quad (10.30)$$

Например, свойству сбалансированности (10.30) при $M = 4$ удовлетворяет многочлен

$$F(x) = (x_1 + x_2 + x_3 + x_4 + x_1x_2 + x_3x_4 + x_1x_2x_3) \bmod 2.$$

В силу отмеченных фактов подбор «оптимальной» функции комбинирования $F_0(\cdot)$ сводится к решению экстремальной задачи:

$$F^*(L_1, L_2, \dots, L_M) \longrightarrow \max_{\{a_i\}, \{a_{ij}\}, \dots, \{a_{12\dots M}\}} \quad (10.31)$$

при ограничении (10.30).

10.11.2. Комбинирование с помощью псевдослучайного прореживания

Рассмотрим еще один способ комбинирования двух LFSR-генераторов G_1 , G_2 . Пусть LFSR-генератор G_1 порождает элементарную двоичную последовательность $\{a_t\}$, а LFSR-генератор G_2 – двоичную селектирующую последовательность $\{s_t\}$. С помощью этих двух последовательностей $\{a_t\}$, $\{s_t\}$ строится выходная последовательность $\{x_t\}$, включающая те биты x_t , для которых соответствующее значение селектора $s_t = 1$; если $s_t = 0$, то значение a_t игнорируется (отбрасывается). Такой генератор двоичной псевдослучайной последовательности предложен в работе [86] и назван *SG-генератором* (*Shrinking Generator*).

Свойства SG-генератора выражаются следующими утверждениями.

Теорема 10.2. Пусть T_a, T_s – соответственно периоды последовательностей $\{a_t\}$ и $\{s_t\}$. Если генераторы G_1, G_2 используют примитивные порождающие многочлены степеней L_1 и L_2 соответственно, а периоды T_a, T_s – взаимно простые числа, то выходная последовательность $\{x_t\}$ имеет период

$$T = (2^{L_1} - 1) \times 2^{L_2 - 1}.$$

Теорема 10.3. В условиях теоремы 10.2 линейная сложность выходной последовательности $\{x_t\}$ удовлетворяет неравенствам

$$L_1 \times 2^{L_2 - 2} \leq \Lambda(\{x_t\}) \leq L_1 \times 2^{L_2 - 1}.$$

10.12. КОНГРУЭНТНЫЙ ГЕНЕРАТОР СО СЛУЧАЙНЫМИ ПАРАМЕТРАМИ

Еще один способ комбинирования двух генераторов G_1, G_2 заключается в том, что G_2 изменяет параметры генератора G_1 с течением времени. Продемонстрируем это на случай, когда G_1 – линейный конгруэнтный генератор:

$$x_t = (a_t x_{t-1} + b_t) \bmod N, \quad t = 1, 2, \dots, \quad (10.32)$$

где $x_0 \in \mathcal{A} = \{0, 1, \dots, N - 1\}$ – некоторое стартовое значение, а

$$A_t = \begin{pmatrix} a_t \\ b_t \end{pmatrix} \in B, \quad t = 1, 2, \dots, \quad (10.33)$$

есть некоторая псевдослучайная последовательность векторов, равномерно распределенных в B .

В работе [83] доказано, что если $|B| = 3$, то наибольшее приближение распределения $\{x_t\}$ к равномерному достигается, если множество параметров B имеет следующий вид:

$$B = \left\{ \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix} \right\}. \quad (10.34)$$

10.13. ЗАДАНИЯ

1. Разработать программные реализации всех генераторов псевдослучайных последовательностей, представленных в указанном подпункте, для значений параметров, заданных преподавателем (см. п. 10.3).
2. Аналитически и с помощью компьютерного моделирования оценить вычислительную сложность (быстродействие) алгоритмов генерации, представленных в п. 10.3.
3. Осуществить статистическое тестирование алгоритмов генерации с помощью батареи тестов ППП «КриптоЛаборатория» и тестов, разработанных при выполнении заданий гл. 6 (см. п. 10.3).

4. Осуществить подбор параметров алгоритмов генерации из условия минимума уклонения статистических характеристик генерируемой последовательности от характеристик РРСП (см. п. 10.3).
5. Выявить «слабости» генераторов, представленных в указанном подпункте, которые могут использоваться для криptoанализа генераторов (см. п. 10.3).
6. Разработать алгоритм статистической (или алгебраической) оценки параметров генератора по наблюдаемой выходной последовательности; оценить точность и вычислительную сложность этого алгоритма (см. п. 10.3).
7. Провести сравнительный анализ генераторов, представленных в п. 10.3, и сформулировать рекомендации по условиям их применения.
8. Осуществить графическую иллюстрацию свойства С₉ (см. п. 10.3).
9. Выполнить задания 1–7 применительно к нелинейным конгруэнтным генераторам (10.3)–(10.6) (см. п. 10.4).
10. Разработать и исследовать модификацию квадратичного конгруэнтного генератора, основанную на введении приращения c_t аналогично (10.5), (10.6) (см. п. 10.4).
11. Разработать и исследовать модификацию генератора Эйхенауэра – Лена, основанную на введении приращения c_t аналогично (10.5), (10.6).
12. Выполнить задания 1–7 применительно к LFSR и генератору Таусворта (10.10)–(10.14) (см. п. 10.5).
13. Разработать и исследовать модификацию генератора Таусворта (см. п. 10.5):

$$A = \prod_{k=1}^m (I_n + (L')^{p_k})(I_n + L^{q_k}).$$
14. Разработать и исследовать модификацию генератора Таусворта (см. п. 10.5) $s_t = b' S(t)$, где b – некоторый заданный двоичный n -вектор (параметр генератора).
15. Выполнить задания 1–7 применительно к генераторам Фибоначчи вида (10.19) (см. п. 10.5).
16. Установить связь между генераторами Фибоначчи и рекуррентами в конечном поле (см. п. 10.5).
17. Выполнить задания 1–7 применительно к генератору ANSI X9.17 (см. п. 10.6).
18. Выполнить задания 1–7 применительно к генератору FIPS-186 с использованием функции G_1 (см. п. 10.6).
19. Выполнить задания 1–7 применительно к генератору FIPS-186 с использованием функции G_2 (см. п. 10.6).
20. Осуществить программную реализацию генератора Yarrow-160 и исследовать его свойства (см. п. 10.6).

21. Выполнить задания 1–7 применительно к RSA-генератору (см. п. 10.7)
22. Выполнить задания 1–7 применительно к генератору Микали – Шнорра (см. п. 10.7).
23. Выполнить задания 1–7 применительно к BBS-генератору (см. п. 10.7)
24. Аналогично пп. 10.7.1 построить модификацию BBS-алгоритма, используя подход Микали – Шнорра.
25. Выполнить задания 1–7 применительно к алгоритму симметризации элементарной двоичной последовательности. В качестве элементарных последовательностей использовать данные из АДП (прил. 1) (см. п. 10.8).
26. Оценить коэффициент использования к [55] для алгоритма симметризации (10.20)–(10.24) (см. п. 10.8).
27. Предполагая, что элементарная последовательность $\{\xi_i\}$ является двоичной однородной цепью Маркова, исследовать изменение результатов (10.23), (10.24) (см. п. 10.8).
28. В предположениях предыдущего задания подобрать преобразования (10.20) так, чтобы $\{x_k\}$ стала РРСП (см. п. 10.8).
29. Построить обобщение алгоритма в задании 28 для случая, когда $\{\xi_i\}$ – s -связная однородная цепь Маркова (см. п. 10.8).
30. Выполнить задания 1–7 применительно к генератору Макларена – Марсальи, используя в качестве G_1 , G_2 мультипликативные конгруэнтные генераторы (см. п. 10.10).
31. Исследовать влияние параметра K на качество генерируемой псевдослучайной последовательности (см. п. 10.10).
32. Методом из раздела 6.8 книги [49] провести теоретическое исследование свойств генератора Макларена – Марсальи (см. п. 10.10).
33. Выполнить задания 1–7 применительно к генератору (10.26)–(10.28) с полиномиальным комбинированием элементарных последовательностей.
34. Разработать алгоритм нахождения сбалансированных полиномов $F(\cdot)$, удовлетворяющих условию (10.30) (см. п. 10.11).
35. Разработать алгоритм нахождения оптимальной функции комбинирования $F_0(\cdot)$ согласно (10.30), (10.31) (см. п. 10.11).
36. Выполнить задания 1–7 применительно к SG-генератору, основанному на псевдослучайном прореживании LFSR-последовательности (см. п. 10.11).
37. Выполнить задания 1–7 применительно к линейному конгруэнтному генератору со случайными параметрами (10.32)–(10.34) (см. п. 10.12).
38. Разработать и исследовать алгоритм генерации со случайными параметрами, в котором в качестве G_1 , G_2 используются LFSR-генераторы (см. п. 10.12).

Г л а в а 11

ПОТОЧНЫЕ КРИПТОСИСТЕМЫ

11.1. ОСНОВНЫЕ ПОНЯТИЯ И КЛАССИФИКАЦИЯ ПОТОЧНЫХ КРИПТОСИСТЕМ

Как уже отмечалось ранее, криптографические системы шифрования делятся на два больших класса: 1) симметричные криптосистемы, или криптосистемы с секретным ключом (*secret-key cryptosystems*); 2) асимметричные криптосистемы, или криптосистемы с открытым ключом (*public-key cryptosystems*). Классификация симметричных криптосистем приведена на рис. 11.1.

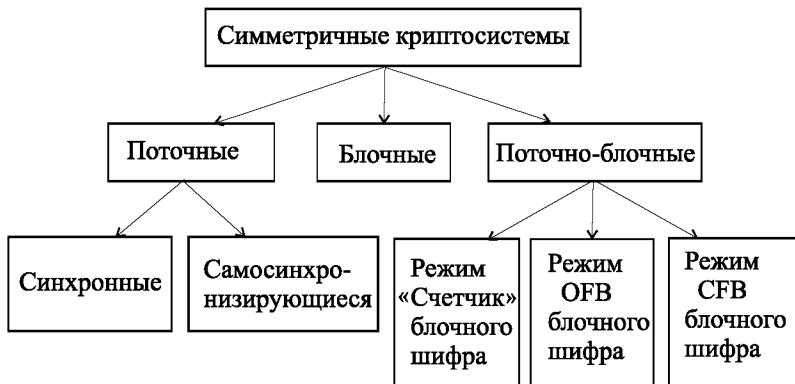


Рис. 11.1. Классификация симметричных криптосистем

Приведем сначала содержательное описание каждого из классов на рис. 11.1 [149]. Блочные криптосистемы предполагают разбиение открытого текста (*plaintext*) на блоки фиксированного размера (например, 64 бита в DES и ГОСТ 28147-89 и 128 бит в AES) и обработку каждого блока одним и тем же преобразованием замены независимо для каждого блока. Поэтому блочный шифр, по существу, – простая подстановка, требующая большой мощности алфавита блоков для предохранения от «атаки со словарем» методом «грубой силы».

В поточных криптосистемах открытый текст также разбивается на блоки, однако их размер значительно меньше, чем в блочных криптосистемах: обычно это биты или байты открытого текста. Основное отличие поточного

шифра от блочного заключается в том, что функциональное преобразование блоков текста в поточной криптосистеме зависит от номера блока, т. е. имеется поток функциональных преобразований, а в блочной криптосистеме функциональное преобразование блоков не зависит от номера блока.

К классу *поточно-блочных криптосистем* (см. рис. 11.1) относятся получившие широкое распространение в «открытой криптографии» блочные криптосистемы (ГОСТ 28147-89, AES и др.), которые снабжены специальными режимами поточного шифрования:

- режим шифрования CFB (Ciphertext FeedBack);
- режим шифрования OFB (Output FeedBack);
- режим Счетчика (Counter).

Представим теперь математическое описание каждого из классов на рис. 11.1.

Пусть определены следующие математические объекты: \mathcal{A} – алфавит открытого текста, совпадающий с алфавитом шифртекста; K – пространство ключей; $\Phi = \{\varphi_\gamma : \gamma \in \Gamma\}$ – параметрическое семейство биекций (подстановок) $\varphi_\gamma : \mathcal{A} \leftrightarrow \mathcal{A}$ с параметром γ (который принято называть управляющим параметром); Γ – множество значений управляющего параметра; $x_t \in \mathcal{A}$ ($t = 1, 2, \dots$) – последовательность символов из \mathcal{A} , определяющая открытый текст; $y_t \in \mathcal{A}$ – шифртекст; $\gamma_t \in \Gamma$ – управляющая последовательность, которую принято называть *гаммой* (ключевым потоком).

Поточной криптосистемой называется криптосистема, в которой:

а) по заданному (сеансовому) ключу $k \in K$ у отправителя и получателя сообщения формируется гамма:

$$\gamma_t = g(k; t), \quad t = 1, 2, \dots; \quad (11.1)$$

б) зашифрование осуществляется отправителем с помощью взаимно однозначного преобразования символа x_t при фиксированном значении символа γ_t ключевого потока:

$$y_t = \varphi_{\gamma_t}(x_t), \quad t = 1, 2, \dots; \quad (11.2)$$

в) расшифрование осуществляется получателем с помощью единственного обратного преобразования при том же самом значении символа γ_t :

$$x_t = \varphi_{\gamma_t}^{-1}(y_t), \quad t = 1, 2, \dots. \quad (11.3)$$

Функция $g(\cdot)$ в соотношении (11.1) по заданному ключу порождает псевдослучайную последовательность $\gamma_t \in \Gamma$ и по своей сути является некоторым криптографическим генератором псевдослучайной последовательности (см. гл. 10).

Согласно (11.1)–(11.3) поточная криптосистема (рис. 11.2) состоит из двух блоков: управляющего блока, генерирующего гамму (11.1) и шифрующего блока, реализующего (11.2), (11.3). Отметим, что для обеспечения конфиденциальности передачи сообщения ключ k является общим секретом отправителя и получателя и должен доставляться им соответствующим образом.

Криптосистемы типа поточной криптосистемы на рис. 11.2, в которых ключ k используется симметрично отправителем и получателем, принято называть *симметричными* (в отличие от *асимметричных криптосистем*, в которых используются два ключа: личный секретный ключ и общедоступный открытый ключ, см. гл. 14).

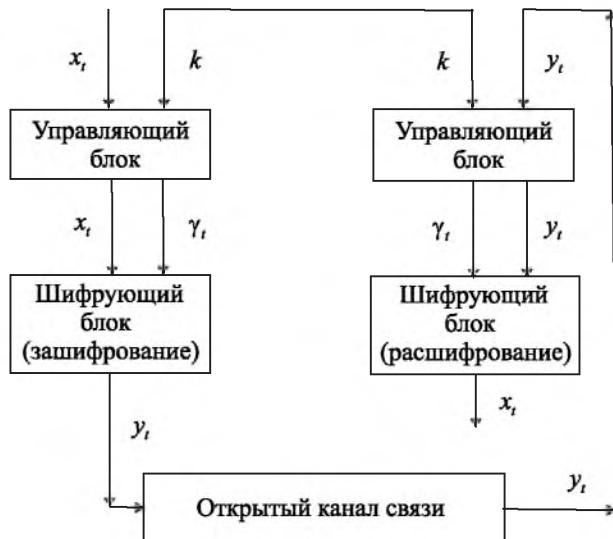


Рис. 11.2. Схема передачи сообщения с использованием поточной криптосистемы

Рассмотрим теперь классификацию поточных криптосистем относительно различных параметров и свойств.

В предельном случае, не применяемом на практике, когда $|\Gamma| = 1$ и (или) функция $g(\cdot)$ в (11.1) не зависит от времени (номера символа) t , то $\gamma_t \equiv \gamma = \text{const}$ и имеем шифр простой замены, который рассматривался в гл. 1.

Если $|\Gamma| = L > 1$ и $g(\cdot)$ зависит от t , определяя L различных подстановок $\varphi_\gamma(\cdot) \in \Phi$, то имеем L -алфавитный поточный шифр.

Если $|\Gamma| = |\mathcal{A}| = L > 1$ и в качестве L подстановок $\{\varphi_\gamma\}$ выбраны такие подстановки, вторые строки которых образуют $(L \times L)$ -латинский квадрат, то получаем поточный шифр, который называется *шифром табличного гаммирования* (при этом вычисления (11.2), (11.3) сводятся к считыванию соответствующих элементов из латинского квадрата).

Если $\mathcal{A} = \Gamma = Z_N = \{0, 1, \dots, N - 1\}$ – кольцо вычетов по модулю N , а $\varphi_\gamma(x) = (x + \gamma) \bmod N$, то (11.2), (11.3) принимают вид

$$y_t = (x_t + \gamma_t) \bmod N, \quad x_t = (y_t - \gamma_t) \bmod N, \quad (11.4)$$

и имеем *шифр модульного гаммирования*. В частности, при $N = 2$ имеем поточный шифр

$$y_t = x_t \oplus \gamma_t, \quad x_t = y_t \oplus \gamma_t, \quad t = 1, 2, \dots, \quad (11.5)$$

который широко используется в Интернете при создании IP-шифраторов. Из (11.5) видно, что в этом случае шифрующие блоки для зашифрования и расшифрования идентичны.

Заметим, что если передаваемое сообщение x_1, \dots, x_n имеет конечную длину n , $K = \mathcal{A}^n$ и ключ k представляет собой фрагмент длины n из равномерно распределенной случайной последовательности k_1, \dots, k_n , то, выбирая $\gamma_t \equiv k_t$ в (11.5), получим совершенно стойкий поточный шифр Вернама (см. гл. 6).

Поточная криптосистема называется *синхронной*, если генерируемая гамма, как указано в (11.1), не зависит от открытого текста $\{x_t\}$, т. е. криптографический генератор гаммы работает автономно. Если счетчики времени у отправителя и получателя работают синхронно, то нарушений в расшифровании (11.3) не происходит. При появлении сбоев (или при вмешательстве противника) в последовательности $\{y_t\}$ некоторые символы могут теряться, что приводит к невозможности правильного расшифрования той части шифртекста, которая идет после потери символа. Для восстановления синхронности работы могут использоваться специальные маркеры, включаемые в передаваемый текст, либо реинициализация состояний (включая замену ключа) шифраторов у отправителя и получателя. Однако с указанным недостатком синхронные поточные криптосистемы обладают существенным положительным качеством: они *не размножают искажений знаков текста*, которые довольно часто имеют место при передаче по реальным каналам связи. Если в канале связи был искажен символ y_t , то этот факт не повлияет на правильность расшифрования всех других символов, кроме y_t .

Самосинхронизирующаяся поточная криптосистема также удовлетворяет соотношениям (11.2), (11.3), однако уравнение (11.1) формирования γ имеет обратную связь по шифртексту (режим CFB, для сравнения см. гл. 12):

$$\gamma_t = g(k; y_{t-1}, \dots, y_{t-n}), \quad t > n, \quad (11.6)$$

где $n \geq 1$ – глубина обратной связи. Из (11.6) следует, что если символ y_{t-n-1} потерян, но последующие n символов y_{t-n}, \dots, y_{t-1} не искажены, то γ_t «выработается» правильно и синхронизация с момента t восстановится. Слабой стороной самосинхронизирующейся поточной криптосистемы является *размножение ошибок*: появление единичной ошибки в шифртексте порождает n ошибок в расшифрованном открытом тексте.

Дадим теперь краткое математическое описание функционирования поточно-блочных криптосистем. Обозначим: n – размер блока открытого текста и шифртекста в блочной криптосистеме;

$$y = E_k(x), \quad x, y \in V_n, -$$

преобразование блочного шифрования на ключе k ; $s_t \in V_n$ – состояние вспомогательного регистра памяти длины n ($t = 1, 2, \dots$); $s_0 \in V_n$ – начальное состояние регистра, называемое синхропосылкой.

Во всех трех вышеуказанных режимах («Счетчик», OFB, CFB) поточно-блочных криптосистем ключевой поток формируется одинаково с помощью шифрования текущего состояния регистра памяти:

$$\bar{\gamma}_t = (\gamma_{tn+1}, \dots, \gamma_{(t+1)n}) = E_k(s_t); \quad (11.7)$$

при этом получается сразу n символов ключевого потока для использования в (11.2). Режимы «Счетчик», OFB, CFB различаются уравнениями динамики состояния s_t регистра памяти.

Для режима «Счетчик» это уравнение динамики есть уравнение рекурсии:

$$s_t = F(s_{t-1}), \quad (11.8)$$

где $F(\cdot) : V_n \rightarrow V_n$ – произвольная дискретная функция, обеспечивающая достаточно большой период последовательности s_t . Часто используется частный случай (11.8):

$$s_t = s_{t-1} + (0, 0, \dots, 0, 1),$$

описывающий функционирование n -разрядного счетчика; отсюда и происходит название данного режима.

В режиме OFB состояние s_t зависит не только от s_{t-1} , но и от $\bar{\gamma}_{t-1}$:

$$s_t = F(s_{t-1}, \bar{\gamma}_{t-1}). \quad (11.9)$$

Для режима CFB состояние s_t зависит от s_{t-1} и ранее выработанного шифртекста $Y_{t-1} = \{y_\tau : \tau \leq t-1\}$:

$$s_t = F(s_{t-1}, Y_{t-1}). \quad (11.10)$$

Функция $F(\cdot)$ в (11.8)–(11.10) выбирается таким образом, чтобы увеличить криптостойкость шифра.

Атаки при известном, выбранном и выбираемом открытом текстах x_1, \dots, x_T для поточных криптосистем эквивалентны. В условиях этих атак криптоаналитику становится известным фрагмент $\gamma_1, \dots, \gamma_T$ гаммы и требуется решить одну из следующих криптоаналитических задач:

S1) определить ключ k , при котором получена эта гамма;

S2) не определяя k , построить алгоритм вычисления (прогнозирования) последующих символов гаммы $\gamma_{T+1}, \gamma_{T+2}, \dots$;

S3) удостовериться, что наблюдаемый отрезок гаммы порожден данным криптографическим генератором G (эта задача называется задачей распознавания криптографического генератора).

В заключение сформулируем ряд требований, на практике предъявляемых к блокам поточной криптосистемы, несоблюдение которых приводит к появлению аналитических или статистических слабостей алгоритма шифрования, снижающих его стойкость [2].

Требования к управляющему блоку (см. рис. 11.2):

- период управляющей гаммы $T \geq T_{max}$, где T_{max} – максимально возможная длина открытого сообщения, подлежащего шифрованию;
- статистические свойства управляющей гаммы γ_t должны быть близки к свойствам равномерно распределенной случайной последовательности;
- в управляющей гамме должны отсутствовать простые аналитические зависимости между близко расположенным символами;
- криптографический алгоритм генерации гаммы γ_t должен обеспечивать заданную высокую сложность определения секретного ключа k .

Требования к шифрующему блоку (см. рис. 11.2):

- применение алгоритма шифрования должно носить универсальный характер и не зависеть от вида шифруемой информации;
- желательно, чтобы шифрующий блок обеспечивал криптографическую стойкость шифра при перекрытии управляющей гаммы, в частности, при повторном использовании ключей.

Выполнение перечисленных требований – необходимо, но не достаточное условие криптографической стойкости поточной криптосистемы.

11.2. РЕКУРРЕНТНЫЕ ПОСЛЕДОВАТЕЛЬНОСТИ И РЕГИСТРЫ СДВИГА

Пусть задана функция $f(x_1, \dots, x_n): \mathcal{A}^n \rightarrow \mathcal{A}$, которая порождает *рекуррентную последовательность* s порядка n с элементами

$$\gamma_{t+n} = f(\gamma_{t+n-1}, \dots, \gamma_t), \quad t = 0, 1, \dots,$$

при заданных начальных значениях $\gamma_0, \dots, \gamma_{n-1}$.

Всякая рекуррентная последовательность является периодической – существуют числа $t_0 \in \mathbb{N}_0$ и $r \in \mathbb{N}$ такие, что $\gamma_{t+r} = \gamma_t$ для всех $t \geq t_0$. Минимальное r с данным свойством называется *минимальным периодом*, а минимальное t_0 – *предпериодом*. Если $t_0 = 0$, то γ_t – чисто периодическая последовательность.

Очевидно, что $r \leq |\mathcal{A}|^n$. Если $r = |\mathcal{A}|^n$, то γ_t называется *последовательностью де Брейна*. Векторы состояния $\Gamma_0 = (\gamma_0, \dots, \gamma_{n-1})$, $\Gamma_1 = (\gamma_1, \dots, \gamma_n), \dots, \Gamma_{r-1} = (\gamma_{r-1}, \dots, \gamma_{r+n-2})$ такой последовательности проходят все множество \mathcal{A}^n , $\Gamma_r = \Gamma_0$.

Далее в главе будем рассматривать бинарный алфавит $\mathcal{A} = \mathbb{F}_2$ и считать, что $\odot = \oplus$, $f \in \mathcal{F}_n$.

Линейная булева функция $f \in \mathcal{F}_n$ порождает *линейную рекуррентную последовательность* (ЛРП), в которой

$$\gamma_{t+n} = \alpha_{n-1}\gamma_{t+n-1} \oplus \dots \oplus \alpha_0\gamma_t, \quad t = 0, 1, \dots,$$

или

$$\gamma_t = \Gamma_{t,1}, \quad \Gamma_{t+1} = \Gamma_t A,$$

где $\Gamma_{t,1}$ – первая компонента вектора $\Gamma_t \in V_n$,

$$A = \begin{pmatrix} 0 & 0 & \dots & 0 & \alpha_0 \\ 1 & 0 & \dots & 0 & \alpha_1 \\ 0 & 1 & \dots & 0 & \alpha_2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & \alpha_{n-1} \end{pmatrix}.$$

Далее полагаем, что вектор Γ_0 начального состояния ЛРП является ненулевым.

Характеристический многочлен

$$p(\lambda) = \lambda^n + \alpha_{n-1}\lambda^{n-1} + \dots + \alpha_0$$

матрицы A называется *характеристическим многочленом* ЛРП $\gamma_0, \gamma_1, \dots, \gamma_n$, A при этом называется *сопровождающей матрицей* $p(\lambda)$.

Последовательность $\gamma_0, \gamma_1, \dots$ удовлетворяет целому семейству линейных рекуррентных соотношений, например всякому соотношению, заданному характеристическим многочленом $p(\lambda)q(\lambda)$, $q(\lambda) \in \mathbb{F}_2[\lambda]$, $q(0) = 1$. При этом однозначно определен *минимальный многочлен* $m(\lambda)$ такой, что всякий характеристический многочлен последовательности $\gamma_0, \gamma_1, \dots$ делится на $m(\lambda)$. Известно, что минимальный период линейной рекуррентной последовательности совпадает с порядком ее минимального многочлена.

Для криптографических приложений важность представляют ЛРП с примитивными характеристическими многочленами, называемые *последовательностями максимального периода*, или *m-последовательностями*. В таблице приведен список всех примитивных триномов степени $n \leq 100$.

Использование триномов удобно при генерации ЛРП, так как для вычисления нового элемента последовательности требуется выполнить всего одно сложение \oplus . В таблице приведем один элемент пары «многочлен $p(\lambda)$ – *возвратный многочлен* $p^*(\lambda) = \lambda^n p(1/\lambda)$ », поскольку $p(\lambda)$ и $p^*(\lambda)$ являются примитивными одновременно.

Примитивные триномы
 $\lambda^n + \lambda^m + 1, 1 \leq n \leq 100, 1 \leq m \leq n/2$

n	m	n	m	n	m	n	m
2	1	20	3	41	3, 20	73	25, 28, 31
3	1	21	2	47	5, 14, 20, 21	79	9, 19
4	1	22	1	49	9, 12, 15, 22	81	4, 16, 35
5	2	23	5, 9	52	3, 7, 19, 21	84	13
6	1	25	3, 7	55	24	87	13
7	1, 3	28	3, 9, 13	57	7, 22	89	38
9	4	29	2	58	19	93	2
10	3	31	3, 6, 7, 13	60	1, 11	94	21
11	2	33	13	63	1, 5, 31	95	11, 17
15	1, 4, 7	35	2	65	18, 32	97	6, 12, 33, 34
17	3, 5, 6	36	11	68	9, 33	98	11, 27
18	3, 7, 9	39	4, 8, 14	71	6, 9, 18, 20, 35	100	37

Схема устройства для вычисления элементов линейной рекуррентной последовательности с характеристическим многочленом

$$\lambda^8 + \lambda^4 + \lambda^3 + \lambda^2 + 1,$$

называемого *регистром сдвига с обратной связью*, изображена на рис. 11.3, а. В ячейках 0, 1, ..., 7 такого регистра в момент времени t содержатся соответственно компоненты $\gamma_t, \gamma_{t+1}, \dots, \gamma_{t+7}$ вектора состояния Γ_t .

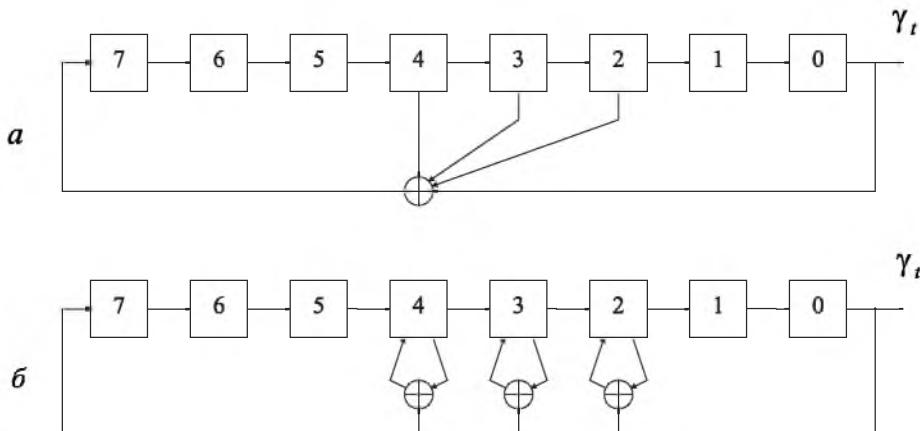


Рис. 11.3. Линейные регистры сдвига

При $\alpha_0 \neq 0$ характеристический многочлен $\tilde{p}(\lambda)$ матрицы

$$\tilde{A} = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_{n-1} & \alpha_0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix}$$

является возвратным к характеристическому многочлену $p(\lambda)$ матрицы A . Регистр сдвига, вырабатывающий последовательность

$$\tilde{\gamma}_t = \tilde{\Gamma}_{t,1}, \quad \tilde{\Gamma}_{t+1} = \tilde{\Gamma}_t \tilde{A}, \quad t = 0, 1, \dots,$$

при выборе $\tilde{p}(\lambda) = \lambda^8 + \lambda^5 + \lambda^4 + \lambda^3 + 1$, представлен на рис. 11.3, б. Программная реализация такого регистра может оказаться эффективнее стандартной реализации (рис. 11.3, а).

11.3. АЛГОРИТМЫ БЕРЛЕКЭМПА – МЕССИ И ЛИНЕЙНАЯ СЛОЖНОСТЬ

Если $\gamma_0, \gamma_1, \dots$ – линейная рекуррентная последовательность с минимальным многочленом $m(\lambda)$ степени n , то ганкелев определитель

$$D_t^{(k)} = \begin{vmatrix} \gamma_t & \gamma_{t+1} & \dots & \gamma_{t+k-1} \\ \gamma_{t+1} & \gamma_{t+2} & \dots & \gamma_{t+k} \\ \dots & \dots & \dots & \dots \\ \gamma_{t+k-1} & \gamma_{t+k} & \dots & \gamma_{t+2k-2} \end{vmatrix}$$

равен 0 для всех $k \geq n + 1$ и всех $t \geq 0$. С другой стороны, если $\gamma_0, \gamma_1, \dots$ – последовательность независимых случайных величин, $0 < \mathbf{P}\{\gamma_t = 0\} < 1$, то

$$\mathbf{P}\left\{D_0^{(k)} = \dots = D_{T-1}^{(k)} = 0\right\} \rightarrow 0$$

при $T \rightarrow \infty$ для любого фиксированного натурального k . Данное свойство можно использовать для построения алгоритмов распознавания ЛРП.

Алгоритм Берлекэмпа – Месси позволяет по $2k$ -отрезку $\gamma_0, \dots, \gamma_{2k-1}$ линейной рекуррентной последовательности определить минимальный многочлен $m(\lambda)$, если степень последнего не превосходит k .

При выполнении алгоритма формируются последовательности многочленов $m_0^*(\lambda) = 1, m_1^*(\lambda), \dots, m_{2k}^*(\lambda)$ и целых чисел $l_0 = 0, l_1, \dots, l_{2k}$, а именно для $t = 0, \dots, 2k - 1$:

1) вычисляется невязка

$$e_t = \gamma_t \oplus \alpha_{t,1}\gamma_{t-1} \oplus \dots \oplus \alpha_{t,n_t}\gamma_{t-l_t},$$

где $\alpha_{t,i}$ – коэффициент при степени λ^i в $m_t^*(\lambda)$;

2) вычисляется многочлен

$$m_{t+1}^*(\lambda) = \begin{cases} m_t^*(\lambda) + m_\tau^*(\lambda)\lambda^{t-\tau}, & \text{если } e_t = 1, \\ m_t^*(\lambda) & \text{в противном случае,} \end{cases}$$

где τ – максимальное число из множества $\{-1, 0, 1, \dots, t-1\}$ такое, что $l_\tau < l_t$ (l_{-1} считается равным -1 , $m_{-1}^*(\lambda) = 1$);

3) вычисляется

$$l_{t+1} = \begin{cases} t + 1 - l_t, & \text{если } e_t = 1 \text{ и } l_t \leq t/2, \\ l_t & \text{в противном случае.} \end{cases}$$

Искомый минимальный многочлен

$$m(\lambda) = \lambda^{l_{2k}} m_{2k}^*(1/\lambda).$$

Пусть теперь γ_t – произвольная последовательность $\gamma_0, \gamma_1, \dots$ элементов поля \mathbb{F}_2 . Линейной сложностью $l_t(\gamma)$ начального t -отрезка последовательности γ называется минимальное целое n такое, что $\gamma_0, \dots, \gamma_{t-1}$ – первые t элементов линейной рекуррентной последовательности порядка n (если $\gamma_0 = \dots = \gamma_{t-1} = 0$, то $l_t(\gamma) = 0$). Линейная сложность $l(\gamma)$ всей последовательности γ определяется как $l_\infty(\gamma)$. Если линейная сложность $l(G)$ последовательностей криптосистемы G невелика, то, применив алгоритм Берлекэмпа – Месси, можно эффективно решить задачу криптоанализа G .

Известно [149], что для последовательности $\gamma_0, \gamma_1, \dots$ независимых случайных величин, принимающих значения 0 и 1 с равной вероятностью, выполняется

$$\mathbf{E}\{l_t(\gamma)\} = \frac{t}{2} + \frac{4+b_t}{18} - \frac{1}{2^t} \left(\frac{1}{3}t + \frac{2}{9} \right),$$

$$\mathbf{D}\{l_t(\gamma)\} = \frac{86}{81} - \frac{1}{2^t} \left(\frac{14-b_t}{27}t + \frac{82-2b_t}{81} \right) - \frac{1}{2^{2t}} \left(\frac{1}{9}t^2 + \frac{4}{27}t + \frac{4}{81} \right),$$

где $b_t = t \bmod 2$. Значения моментов статистик $l_t(\gamma)$ используются в teste для проверки гипотезы о независимости и равновероятности символов $\gamma_0, \gamma_1, \dots$ (см. п. 6.15).

Последовательность $l_1(\gamma), l_2(\gamma), \dots$ называется профилем линейной сложности γ . Такой профиль совпадает с формируемой при выполнении алгоритма Берлекэмпа – Месси последовательностью l_1, l_2, \dots .

11.4. КОМБИНИРОВАНИЕ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Опишем, следуя п. 10.11, основные способы комбинирования выходных рекуррентных последовательностей $\gamma_0^{(i)}, \gamma_1^{(i)}, \dots$ генераторов $G^{(i)}$, $i = 1, \dots, d$, для получения последовательности $\gamma_0, \gamma_1, \dots$ поточной криптосистемы G . Стартовые значения $\Gamma_0^{(i)}$ генераторов $G^{(i)}$ определяются ключом θ криптосистемы.

При *параллельном подключении* (рис. 11.4) выходные символы $G^{(i)}$ обрабатываются функцией $g \in \mathcal{F}_d$:

$$\gamma_t = g(\gamma_t^{(1)}, \dots, \gamma_t^{(d)}), \quad t = 0, 1, \dots.$$

Например, для генератора Геффе [99] $d = 3$ и $g(x_1, x_2, x_3) = x_1x_2 \oplus (x_1 \oplus 1)x_3$.

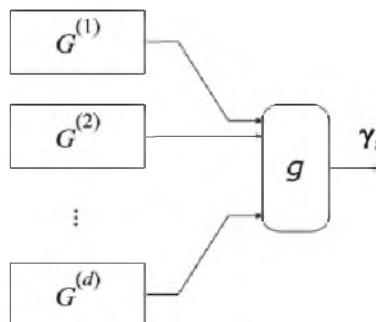


Рис. 11.4. Параллельное подключение генераторов

Выходной символ $\gamma_t^{(1)}$ генератора $G^{(1)}$ определяет выбор между выходными символами $\gamma_t^{(2)}$ и $\gamma_t^{(3)}$ двух других генераторов.

Если $G^{(i)}$ – генератор m -последовательностей порядка n_i , $i = 1, \dots, d$, то линейную сложность последовательности $\gamma_0, \gamma_1, \dots$ можно определить, вычислив значение многочлена Жегалкина $g(x_1, \dots, x_d)$ при значениях n_i переменных x_i , выполняя умножение и сложение в кольце \mathbb{Z} , а не в поле \mathbb{F}_2 . Например, линейная сложность выходных последовательностей генератора Геффе равна $n_1n_2 + n_1n_3 + n_3$.

В *пороговом генераторе* [80] $d = 2k + 1$ – нечетное число. Используется *пороговая функция* g : $g(x) = 1$ – только если $w(x) \geq k + 1$, где $w(x)$ – вес Хэмминга (количество ненулевых координат) вектора x . Таким образом, $\gamma_t = 1$, только если среди выходных символов $\gamma_t^{(1)}, \dots, \gamma_t^{(d)}$ единиц больше, чем нулей.

При *последовательном подключении* выходные символы генератора $G^{(i)}$ управляют выполнением преобразований на генераторе $G^{(i+1)}$. Он может либо выполнять стандартное рекуррентное преобразование (умножение вектора состояния $\Gamma_t^{(i+1)}$ на сопровождающую матрицу характеристического многочлена $A^{(i+1)}$, *шаг*), либо проставливать ($\Gamma_{t+1}^{(i)} = \Gamma_t^{(i)}$, *стоп*). Например, в *каскаде Голлманна* [82] (рис. 11.5) генератор $G^{(1)}$ вырабатывает обычную линейную рекуррентную последовательность. Функционирование генераторов $G^{(i)}$, $i = 2, \dots, d$, осуществляется по правилам:

$$\gamma_t^{(i)} = \gamma_t^{(i-1)} \oplus \Gamma_{t,1}^{(i)}, \quad \Gamma_{t+1}^{(i)} = \begin{cases} \Gamma_t^{(i)} A^{(i)}, & \text{если } \gamma_t^{(i-1)} = 1, \\ \Gamma_t^{(i)}, & \text{если } \gamma_t^{(i-1)} = 0 \end{cases}$$

и $\gamma_t = \gamma_t^{(d)}$, $t = 0, 1, \dots$

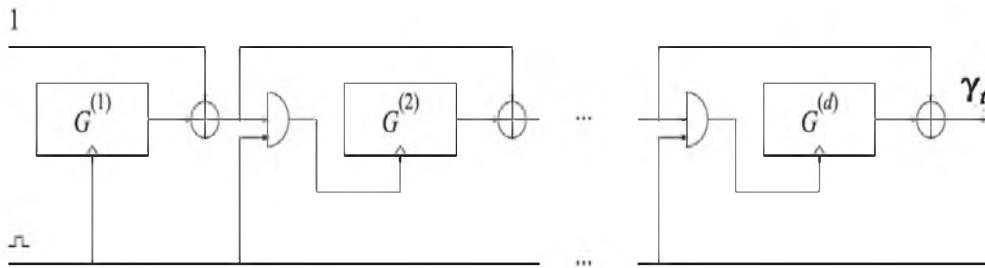


Рис. 11.5. Каскад Голлманна

Переключающий шаг-стоп генератор [105] (рис. 11.6) использует параллельное и последовательные подключения.

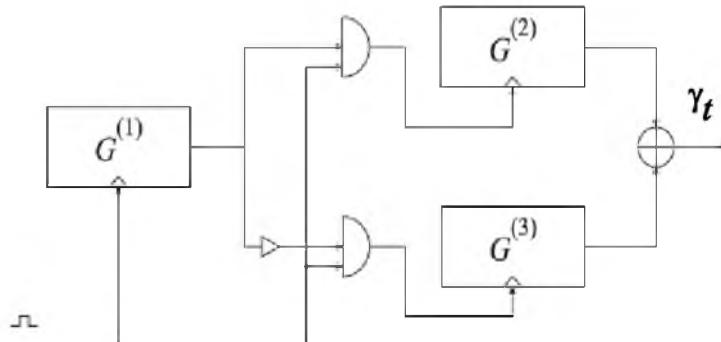


Рис. 11.6. Переключающий шаг-стоп генератор

Здесь $d = 3$,

$$\Gamma_{t+1}^{(2)} = \begin{cases} \Gamma_t^{(2)} A^{(2)}, & \gamma_t^{(1)} = 1, \\ \Gamma_t^{(2)}, & \gamma_t^{(1)} = 0, \end{cases} \quad \Gamma_{t+1}^{(3)} = \begin{cases} \Gamma_t^{(3)}, & \gamma_t^{(1)} = 1, \\ \Gamma_t^{(3)} A^{(3)}, & \gamma_t^{(1)} = 0 \end{cases}$$

и $\gamma_t = \gamma_t^{(2)} \oplus \gamma_t^{(3)}$.

Иной способ последовательного подключения – выходная последовательность одного генератора управляет выбором выходных символов другого генератора. Например, для *прореживающего генератора* [86] $d = 2$,

$$\gamma_t = \gamma_{\tau_t}^{(2)},$$

где τ_t – номер t -й единицы в последовательности $\gamma_0^{(1)}, \gamma_1^{(1)}, \dots$

11.5. СТАТИСТИЧЕСКОЕ ОЦЕНИВАНИЕ НАЧАЛЬНОГО СОСТОЯНИЯ ЛРП

Пусть ЛРП $\gamma_0, \gamma_1, \dots$ с характеристическим многочленом

$$p(\lambda) = \lambda^n + \alpha_{n-1}\lambda^{n-1} + \dots + \alpha_1\lambda + 1,$$

содержащим $(q + 1)$ ненулевых коэффициентов, непосредственно не наблюдается. Криптоаналитику становится известной *искаженная последовательность* [155]

$$z_t = \gamma_t \oplus \xi_t, \quad t = 0, \dots, T - 1.$$

Здесь ξ_0, \dots, ξ_{T-1} – реализация схемы независимых испытаний, в которой событие $\{\xi_0 = 0\}$ наступает с вероятностью $(1 + \varepsilon)/2$, $0 < \varepsilon < 1$. Требуется определить вектор начального состояния $\Gamma_0 \in V_n$.

Криптоаналитик выбирает всевозможные векторы начального состояния $\hat{\Gamma}_0 \in V_n \setminus \{0\}$ и строит ЛРП $\hat{\gamma}_0, \hat{\gamma}_1, \dots, \hat{\gamma}_{T-1}$. В качестве статистической оценки Γ_0 выбирается вектор $\hat{\Gamma}_0$, доставляющий минимум сумме:

$$\sum_{t=0}^{T-1} \mathbf{1}\{\hat{\gamma}_t = z_t\} \rightarrow \min_{\Gamma_0}.$$

Вероятность ошибочного решения близка к нулю, если

$$\frac{n}{T} < 1 - h_2\left(\frac{1 + \varepsilon}{2}\right),$$

где $h_2(\cdot)$ – *двоичная энтропия* (см. гл. 7):

$$h_2(x) = -x \log_2(x) - (1 - x) \log_2(1 - x), \quad 0 \leq x \leq 1, \quad 0 \times \log_2(0) = 0.$$

Эффективность атаки можно повысить, если q мало. Используя результат из задания 11, для чисел $t, k = 0, 1, \dots, t + 2^k n < T$, запишем линейные равенства:

$$\gamma_t \oplus \alpha_1 \gamma_{t+2^k} \oplus \dots \oplus \alpha_{n-1} \gamma_{t+2^k(n-1)} \oplus \gamma_{t+2^k n} = 0,$$

связывающие элементы ЛРП. Если $z_t = \gamma_t$, т. е. $\xi_t = 0$, то соотношение

$$z_t \oplus \alpha_1 z_{t+2^k} \oplus \dots \oplus \alpha_{n-1} z_{t+2^k(n-1)} \oplus z_{t+2^k n} = 0$$

выполняется с вероятностью

$$\mathbf{P} \left\{ \alpha_1 \xi_{t+2^k} \oplus \dots \oplus \alpha_{n-1} \xi_{t+2^k(n-1)} \oplus \xi_{t+2^k n} = 0 \right\} = \frac{1}{2} (1 + \varepsilon^q).$$

Если же $z_t \neq \gamma_t$, то такое соотношение выполняется с вероятностью

$$(1 - \varepsilon^q)/2.$$

Пусть наблюдение z_t входит в R соотношений, r из которых выполняются. При таком условии вероятность β_t совпадения символов z_t и γ_t равна

$$\beta_t = \frac{(1 + \varepsilon)(1 + \varepsilon^q)^r(1 - \varepsilon^q)^{R-r}}{(1 + \varepsilon)(1 + \varepsilon^q)^r(1 - \varepsilon^q)^{R-r} + (1 - \varepsilon)(1 - \varepsilon^q)^r(1 + \varepsilon^q)^{R-r}}.$$

В ходе описанной в работе [132] атаки выбираются элементы z_{t_1}, \dots, z_{t_m} , $m \geq n$, наблюдаемой последовательности, соответствующие максимальным значениям вероятностей $\beta_{t_1}, \dots, \beta_{t_m}$. Задается порог Δ , $0 < \Delta < 1$, определяются всевозможные векторы невязок (e_1, \dots, e_m) , для которых

$$\prod_{i=1}^m \beta_{t_i}^{1-e_i} (1 - \beta_{t_i})^{e_i} \geq \Delta,$$

и составляется система линейных уравнений

$$\hat{\gamma}_{t_i} = \hat{\gamma}_{t_i} \left(\hat{\Gamma}_0 \right) = z_{t_i} \oplus e_i, \quad i = 1, \dots, m,$$

относительно неизвестного вектора $\hat{\Gamma}_0$. Искомая оценка начального состояния Γ_0 выбирается среди решений получаемых систем.

11.6. КРИПТОАНАЛИЗ ПОТОЧНЫХ ШИФРОВ

11.6.1. Корреляционный криptoанализ

Пусть в поточной криптосистеме G используется d генераторов $G^{(1)}, \dots, G^{(d)}$. Если при случайном выборе ключа k символы $\gamma_0, \gamma_1, \dots$ наблюдаемой последовательности G_θ коррелируют с символами ненаблюдаемой выходной последовательности $\gamma_0^{(1)}, \gamma_1^{(1)}, \dots$ генератора $G^{(1)}$, то возможно определение части ключа k – вектора $\Gamma_0^{(1)}$ начального состояния генератора $G^{(1)}$ [155]. Действительно, пусть для простоты $G^{(1)}$ – генератор ЛРП и

$$\mathbf{P}\{\xi_t = 0\} = \frac{1 + \varepsilon}{2}, \quad \text{где } \xi_t = \gamma_t \oplus \gamma_t^{(1)}, \quad 0 < \varepsilon < 1.$$

Применяя описанные в п. 11.5 методы, определяем вектор $\Gamma_0^{(1)}$ начального состояния искаженной ЛРП $\gamma_0^{(1)}, \gamma_1^{(1)}, \dots$. Если

$$\mathbf{P}\{\gamma_t = \gamma_t^{(2)}\} \neq \frac{1}{2},$$

то можно определить вектор $\Gamma_0^{(2)}$ начального состояния генератора $G^{(2)}$ и т. д. Такая процедура иллюстрирует важный принцип криптоанализа – «разделяй и властвуй».

Рассмотрим применение корреляционных методов для криптоанализа генератора Беса – Пайпера [77] (рис. 11.7), в котором:

- а) $d = 3$;
- б) $G^{(1)}, G^{(3)}$ – генераторы m -последовательностей с известными примитивными характеристическими многочленами $p_1(\lambda)$ и $p_3(\lambda)$;
- в) многочлен $p_3(\lambda)$ содержит нечетное количество ненулевых коэффициентов;
- г) $\gamma_t^{(2)} = \Gamma_{t,1}^{(2)}, \Gamma_{t+1}^{(2)} = \Gamma_t^{(2)} A^{(2)}$, если $\gamma_t^{(1)} = 1$; $\Gamma_{t+1}^{(2)} = \Gamma_t^{(2)}$, если $\gamma_t^{(1)} = 0$;
- д) $A^{(2)}$ – сопровождающая матрица примитивного многочлена $p_2(\lambda)$;
- е) $\gamma_t = \gamma_t^{(2)} \oplus \gamma_t^{(3)}$.

Описанная в работе [163] корреляционная атака позволяет по последовательности $\gamma_0, \gamma_1, \dots$ генератора Беса – Пайпера определить векторы $\Gamma_0^{(1)}, \Gamma_0^{(2)}, \Gamma_0^{(3)}$ и многочлен $p_2(\lambda)$. Атака проводится следующим образом (для упрощения выкладок считаем, что в m -последовательностях символы 0 и 1 встречаются одинаково часто).

1. По наблюдениям $\gamma_0 \oplus \gamma_1, \gamma_1 \oplus \gamma_2, \dots$ определяется линейная рекуррентная последовательность $\gamma_0^{(3)} \oplus \gamma_1^{(3)}, \gamma_1^{(3)} \oplus \gamma_2^{(3)}, \dots$. Это возможно, так как

$$\mathbf{P}\{\gamma_t^{(3)} \oplus \gamma_{t+1}^{(3)} = \gamma_t \oplus \gamma_{t+1}\} = \mathbf{P}\{\gamma_t^{(2)} = \gamma_{t+1}^{(2)}\} = \frac{3}{4} \neq \frac{1}{2}.$$

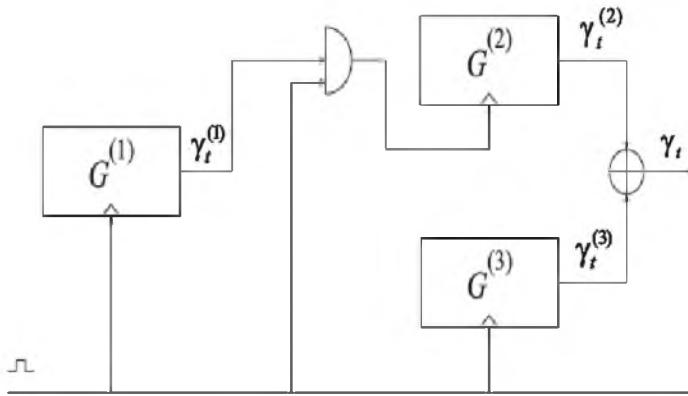


Рис. 11.7. Генератор Беса – Пайпера

2. По последовательности $\gamma_0^{(3)} \oplus \gamma_1^{(3)}, \gamma_1^{(3)} \oplus \gamma_2^{(3)}, \dots$ определяется последовательность $\gamma_0^{(3)}, \gamma_1^{(3)}, \dots$ (см. задание 16).

3. Определяется последовательность $\gamma_t^{(2)} = \gamma_t \oplus \gamma_t^{(3)}, t = 0, 1, \dots$.

4. Строится последовательность

$$\hat{\gamma}_t^{(1)} = \gamma_t \oplus \gamma_t^{(3)} \oplus \gamma_{t+1} \oplus \gamma_{t+1}^{(3)}, \quad t = 0, 1, \dots.$$

5. По последовательности $\hat{\gamma}_0^{(1)}, \hat{\gamma}_1^{(1)}, \dots$ определяется линейная рекуррентная последовательность $\gamma_0^{(1)}, \gamma_1^{(1)}, \dots$ генератора $G^{(1)}$. Это возможно, поскольку

$$\begin{aligned} \mathbf{P}\left\{\gamma_t^{(1)} \neq \hat{\gamma}_t^{(1)}\right\} &= \mathbf{P}\left\{\gamma_t^{(1)} \neq \gamma_t^{(2)} \oplus \gamma_{t+1}^{(2)}\right\} = \\ &= \mathbf{P}\left\{\gamma_t^{(1)} = 1, \gamma_t^{(2)} = \gamma_{t+1}^{(2)}\right\} = \frac{1}{4} \neq \frac{1}{2}. \end{aligned}$$

6. В последовательности $\gamma_0^{(2)}, \gamma_1^{(2)}, \dots$ удаляются повторяющиеся при $\gamma_t^{(1)} = 0$ символы $\gamma_{t+1}^{(2)}$. К полученной последовательности применяется алгоритм Берлекэмпа – Месси и определяется многочлен $p_2(\lambda)$.

11.6.2. Криптоанализ на основе повторного использования гаммы

Пусть рассматривается некоторая поточная криптосистема, использующая один из самых распространенных шифров – шифр модульного гаммирования, рассмотренный в п. 11.1:

$$y_t = (x_t + \gamma_t) \bmod N, \quad t = 1, 2, \dots, \quad (11.11)$$

где $\{\gamma_t\}$ – ключевой поток, или гамма, порождаемая некоторым криптографическим генератором, $\gamma_t \in \mathcal{A} = \{0, 1, \dots, N - 1\}$; $\{x_t\}$ – последовательность символов из алфавита \mathcal{A} , подлежащая зашифрованию; N – мощность алфавита; $\{y_t\}$ – криптограмма, являющаяся результатом зашифрования.

Рассмотрим криптоатаку, заключающуюся в том, что криptoаналитик имеет две криптограммы некоторой длины T :

$$Y^{(1)} = (y_1^{(1)}, \dots, y_T^{(1)}) \in \mathcal{A}^T, \quad Y^{(2)} = (y_1^{(2)}, \dots, y_T^{(2)}) \in \mathcal{A}^T,$$

полученные зашифрованием двух различных сообщений:

$$X^{(1)} = (x_1^{(1)}, \dots, x_T^{(1)}) \in \mathcal{A}^T, \quad X^{(2)} = (x_1^{(2)}, \dots, x_T^{(2)}) \in \mathcal{A}^T -$$

с использованием одной и той же гаммы

$$\Gamma = (\gamma_1, \dots, \gamma_T) \in \mathcal{A}^T.$$

Задача криптоаналитика заключается в том, чтобы по имеющимся криптограммам $Y^{(1)}, Y^{(2)}$ восстановить сообщения $X^{(1)}, X^{(2)}$.

В силу (11.11) имеем соотношения зашифрования:

$$y_t^{(i)} = (x_t^{(i)} + \gamma_t), \quad t = 1, \dots, T; \quad i = 1, 2. \quad (11.12)$$

Используя (11.12), построим поэлементную разность криптограмм $Y^{(1)}, Y^{(2)}$:

$$z_t = (y_t^{(1)} - y_t^{(2)}) \bmod N = (x_t^{(1)} - x_t^{(2)}) \bmod N, \quad t = 1, \dots, T. \quad (11.13)$$

Именно благодаря использованию одной и той же гаммы, как следует из (11.13), криптоаналитик получает точное значение разности $Z = (z_1, \dots, z_T)$ двух сообщений $X^{(1)}, X^{(2)}$, по которой необходимо восстановить эти сообщения. Заметим, что если восстановлено одно из этих сообщений, например $X^{(1)}$ (без потери общности):

$$x_1^{(1)} = j_1, \dots, x_T^{(1)} = j_T,$$

то в силу (11.13) другое сообщение $X^{(2)}$ определяется однозначно:

$$x_1^{(2)} = (j_1 - z_1) \bmod N, \dots, x_T^{(2)} = (j_T - z_T) \bmod N. \quad (11.14)$$

Для восстановления $X^{(1)}, X^{(2)}$ применим критерий максимума правдоподобия, используя вероятностную модель сообщений.

Теорема 11.1. Если $X^{(i)}$ – последовательность независимых в совокупности одинаково распределенных символов открытого текста:

$$\mathbf{P} \left\{ x_t^{(i)} = j \right\} = \pi_j, \quad j \in \mathcal{A}; \quad i = 1, 2, \quad (11.15)$$

причем $X^{(1)}$ и $X^{(2)}$ независимы и одинаково распределены, то по критерию максимума правдоподобия наилучшие оценки $\widehat{X}^{(1)}, \widehat{X}^{(2)}$ для текстов $X^{(1)}, X^{(2)}$ имеют следующий вид:

$$\widehat{X}^{(1)} = \left(\widehat{x}_1^{(1)}, \dots, \widehat{x}_T^{(1)} \right), \quad \widehat{x}_t^{(1)} = \arg \max_{j_t \in \mathcal{A}} (\pi_{j_t} \cdot \pi_{(j_t - z_t) \bmod N}), \quad t = 1, \dots, T;$$

$$\widehat{X}^{(2)} = \left(\widehat{x}_1^{(2)}, \dots, \widehat{x}_T^{(2)} \right), \quad \widehat{x}_t^{(2)} = \left(\widehat{x}_T^{(1)} - z_t \right) \bmod N, \quad t = 1, \dots, T. \quad (11.16)$$

Доказательство. Построим функцию правдоподобия в силу (11.13)–(11.15):

$$\begin{aligned} L(j_1, \dots, j_T) &= \mathbf{P} \left\{ x_1^{(1)} = j_1, \dots, x_T^{(1)} = j_T, \right. \\ &\quad \left. x_1^{(2)} = (j_1 - z_1) \bmod N, \dots, x_T^{(2)} = (j_T - z_T) \bmod N \right\} = \\ &= \prod_{t=1}^T \mathbf{P} \left\{ x_t^{(1)} = j_t \right\} \mathbf{P} \left\{ x_t^{(2)} = (j_t - z_t) \bmod N \right\} = \prod_{t=1}^T (\pi_{j_t} \cdot \pi_{(j_t - z_t) \bmod N}), \\ &\quad j_1, \dots, j_T \in \mathcal{A}. \end{aligned} \quad (11.17)$$

Из (11.17) видно, что функция правдоподобия $L(\cdot)$ сепарабельна и задача ее максимизации

$$L(j_1, \dots, j_T) \rightarrow \max_{j_1, \dots, j_T \in \mathcal{A}}$$

распадается на T независимых задач, и это с учетом (11.14) приводит к решению (11.16). \square

Теорема 11.1 при восстановлении текстов $X^{(1)}, X^{(2)}$ использует в качестве модели текста вероятностную схему независимых испытаний и учитывает лишь статистику частот символов $\{\pi_j\}$, не учитывая статистики зависимости символов открытого текста. Следующая теорема учитывает не только $\{\pi_j\}$, но и статистику марковской зависимости символов открытого текста (см. гл. 5, 7).

Теорема 11.2. Если $X^{(i)}$ – стационарная цепь Маркова с пространством состояний \mathcal{A} и матрицей вероятностей одношаговых переходов $P = (p_{jk})$:

$$p_{jk} = \mathbf{P} \left\{ x_{t+1}^{(i)} = k \mid x_t^{(i)} = j \right\}, \quad t = 1, 2, \dots, \quad (11.18)$$

причем $X^{(1)}$ и $X^{(2)}$ независимы и одинаково распределены, то по критерию максимума правдоподобия наилучшие оценки $\widehat{X}^{(1)}, \widehat{X}^{(2)}$ для текстов $X^{(1)}, X^{(2)}$ имеют следующий вид:

$$\begin{aligned}\widehat{X}^{(1)} &= \left(\widehat{x}_1^{(1)}, \dots, \widehat{x}_T^{(1)}\right) = \arg \max_{j_1, \dots, j_T \in \mathcal{A}} \left(\ln \pi_{j_1} + \ln \pi_{(j_1 - z_1) \bmod N} + \right. \\ &\quad \left. + \sum_{t=1}^{T-1} \left(\ln p_{j_t, j_{t+1}} + \ln p_{(j_t - z_t) \bmod N, (j_{t+1} - z_{t+1}) \bmod N} \right) \right),\end{aligned}$$

$$\widehat{X}^{(2)} = \left(\widehat{x}_1^{(2)}, \dots, \widehat{x}_T^{(2)}\right), \quad \widehat{x}_t^{(2)} = \left(\widehat{x}_t^{(1)} - z_t\right) \bmod N, \quad t = 1, \dots, T. \quad (11.19)$$

Доказательство. В силу условия стационарности распределение вероятностей $\pi = (\pi_0, \dots, \pi_{N-1})'$ является решением системы уравнений

$$\begin{cases} P' \pi = \pi, \\ \sum_{j=0}^{N-1} \pi_j = 1. \end{cases} \quad (11.20)$$

Учитывая этот факт, марковское свойство $X^{(1)}, X^{(2)}$ и соотношение (11.18), построим логарифмическую функцию правдоподобия:

$$\begin{aligned}l(j_1, \dots, j_T) &= \ln L(j_1, \dots, j_T) = \ln \mathbf{P} \left\{ x_1^{(1)} = j_1, \dots, x_T^{(1)} = j_T, \right. \\ &\quad \left. x_1^{(2)} = (j_1 - z_1) \bmod N, \dots, x_T^{(2)} = (j_T - z_T) \bmod N \right\} = \\ &= \ln \left(\left(\pi_{j_1} \prod_{t=1}^{T-1} p_{j_t, j_{t+1}} \right) \left(\pi_{(j_1 - z_1) \bmod N} \prod_{t=1}^{T-1} p_{(j_t - z_t) \bmod N, (j_{t+1} - z_{t+1}) \bmod N} \right) \right) = \\ &= \ln \pi_{j_1} + \ln \pi_{(j_1 - z_1) \bmod N} + \\ &\quad + \sum_{t=1}^{T-1} \left(\ln p_{j_t, j_{t+1}} + \ln p_{(j_t - z_t) \bmod N, (j_{t+1} - z_{t+1}) \bmod N} \right). \quad (11.21)\end{aligned}$$

Выбирая оценку для X_1 по критерию максимума логарифмической функции правдоподобия

$$l(j_1, \dots, j_T) \rightarrow \max_{j_1, \dots, j_T \in \mathcal{A}}, \quad (11.22)$$

а также учитывая (11.14), приходим к (11.19). \square

Заметим, что логарифмическая функция правдоподобия (11.21) имеет следующий сепарабельный вид:

$$l(j_1, \dots, j_T) = \sum_{j=1}^{T-1} f(j_t, j_{t+1}).$$

Как известно [11], в такой ситуации вместо перебора N^T вариантов экстремальная задача (11.22) эффективно решается методом динамического программирования с использованием функции Беллмана одной переменной. Для решения этой задачи также может использоваться алгоритм Баума [67].

Теорема 11.2 допускает расширение для ситуации, когда в качестве модели сообщения $X^{(i)}$ используется цепь Маркова высокого порядка (см. гл. 5). Отметим, что точность восстановления $X^{(1)}, X^{(2)}$ может быть повышена, если криптоаналитик располагает некоторой дополнительной априорной информацией о некоторых символах $X^{(1)}, X^{(2)}$; такие ситуации порождаются наличием некоторых обязательных служебных символов, особенно в начале и в конце сообщений.

Рассмотрим еще специальный частный случай криптоатаки на основе повторного использования гаммы-атаки с помощью вставки символа (insertion attack), представленный в [43]. Пусть сообщение $X^{(1)}$ было отправлено повторно с той же гаммой, но в нем был вставлен некоторый символ x^* , например, на 2-й позиции:

$$X^{(1)} = (x_1, x_2, x_3, x_4, \dots), \quad X^{(2)} = (x_1, x^*, x_2, x_3, x_4, \dots).$$

Такая ситуация может возникнуть естественно или с умыслом, если передающая сторона по ошибке пропустила некоторый символ. При этом простыми алгебраическими выкладками криптоаналитик начиная со 2-й позиции последовательно восстанавливает точные значения не только $X^{(1)}, X^{(2)}$, но и гамму при известном значении x^* :

$$\begin{aligned}\gamma_2 &= (y_2^{(2)} - x^*) \bmod N, & x_2 &= (y_2^{(1)} - \gamma_2) \bmod N, \\ \gamma_3 &= (y_3^{(2)} - x_2) \bmod N, & x_3 &= (y_3^{(1)} - \gamma_3) \bmod N, \\ \gamma_4 &= (y_4^{(2)} - x_3) \bmod N, & x_4 &= (y_4^{(1)} - \gamma_4) \bmod N\end{aligned}$$

и так далее.

Криптоаналитик может эффективно использовать и другие ошибки в текстах, допускаемые передающей стороной, при повторном использовании гаммы. Поэтому в поточных криптосистемах повторное использование гаммы недопустимо.

11.6.3. Другие методы криптоанализа поточных криптосистем

Остановимся кратко на других известных в литературе методах криптоанализа поточных криптосистем [115, 117, 157].

Методы, основанные на стохастических аналогах [2, 8, 43, 119].

Суть этих методов заключается в замене криптографического генератора либо его узлов некоторыми упрощенными стохастическими моделями и использовании этих моделей для восстановления сообщения, ключа или для прогнозирования будущих значений гаммы с некоторой достаточно большой вероятностью успеха.

Алгебраические методы [2]. Эти методы используют атаку при известном открытом тексте небольшого размера T . На основании доступных значений ключевого потока $\gamma_1, \dots, \gamma_T \in \mathcal{A}$ с учетом уравнения (11.1) составляется система T уравнений относительно неизвестного сеансового ключа:

$$\begin{cases} g(k, 1) = \gamma_1, \\ \vdots \\ g(k, T) = \gamma_T. \end{cases}$$

При решении этой системы находится значение ключа k , которое используется для восстановления $\{\gamma_{T+1}, \gamma_{T+2}, \dots\}$, и согласно (11.3) восстанавливается фрагмент последующего текста $\{x_{T+1}, x_{T+2}, \dots\}$.

11.7. ЗАДАНИЯ

1. Пусть функция $f(x_1, \dots, x_n)$ порождает последовательность де Брейна. Доказать:

1) $f(x_1, \dots, x_n) = g(x_1, \dots, x_{n-1}) \oplus x_n$, где $g \in \mathcal{F}_{n-1}$, причем $g(0, \dots, 0) = g(1, \dots, 1) = 1$;

2) функции

$$\begin{aligned} &g(x_{n-1}, \dots, x_1) \oplus x_n; \\ &g(x_1 \oplus 1, \dots, x_{n-1} \oplus 1) \oplus x_n; \\ &g(x_1, \dots, x_{n-1}) \oplus x_1 \dots x_{n-2} \oplus x_2 \dots x_{n-1} \oplus x_n; \\ &g(x_1, \dots, x_{n-1}) \oplus (x_1 \oplus x_{n-1})(x_2 \oplus 1) \dots (x_{n-2} \oplus 1) \oplus x_n \end{aligned}$$

также порождают последовательности де Брейна;

3) функция $f(x_1, \dots, x_n) \oplus (x_1 \oplus 1) \dots (x_{n-1} \oplus 1)$ порождает последовательность, векторы состояния которой пробегают все множество $V_n \setminus \{0\}$.

2. Найти все функции $f \in \mathcal{F}_n$, порождающие последовательности де Брейна порядка $n = 2, 3, 4$. Проверить, что количество таких функций равно $2^{2^{n-1}-n}$.

3. Доказать, что при $\alpha_0 \neq 0$:

- 1) ЛРП $\gamma_0, \gamma_1, \dots$ является чисто периодической последовательностью;
- 2) минимальный период ЛРП $\gamma_0, \gamma_1, \dots$ делит порядок матрицы A как элемента полной линейной группы $GL_n(\mathbb{F}_2)$ (мультиликативной группы всех невырожденных $n \times n$ матриц над полем \mathbb{F}_2);
- 3) если $n > 1$ и $\gamma_0 = \dots = \gamma_{n-2} = 0, \gamma_{n-1} = 1$, то минимальный период ЛРП $\gamma_0, \gamma_1, \dots$ совпадает с порядком матрицы A в группе $GL_n(\mathbb{F}_2)$.

4. Доказать, что:

- 1) многочлен $p(\lambda) \in \mathbb{F}_2[\lambda]$ делит многочлен $\lambda^r - 1$ тогда и только тогда, когда $\text{ord } p(\lambda) \mid r$;
- 2) если $p(\lambda)$ – неприводимый многочлен степени n , то $\text{ord } p(\lambda) \mid (2^n - 1)$ (просуммировать периоды всех ЛРП с характеристическим многочленом $p(\lambda)$);
- 3) если $2^n - 1$ – простое число Мерсенна, то всякий неприводимый многочлен $p(\lambda)$ степени n является примитивным.

5. Написать программу, которая по заданным характеристическому многочлену $p(\lambda)$ и вектору начального состояния Γ_0 строит ЛРП заданной длины T .

6. Разработать алгоритм и написать программу оценки снизу порядка заданной линейной рекуррентной последовательности. Использовать ганкелевы определители.

7. Написать программу определения минимального многочлена заданной линейной рекуррентной последовательности.

8. Доказать:

- 1) $l_t(\gamma) \leq t$, причем $l_t(\gamma) = t$, только если γ_{t-1} – единственный ненулевой элемент в начальном t -отрезке γ ;
- 2) если γ – периодическая последовательность с периодом r , то $l(\gamma) \leq r$;
- 3) $l_{t+1}(\gamma) > l_t(\gamma)$, только если $l_t(\gamma) \leq t/2$, при этом $l_{t+1}(\gamma) + l_t(\gamma) = t + 1$.

9. Написать программу вычисления профиля линейной сложности заданной последовательности γ . Вычислить профиль последовательности

$$\gamma_t = \begin{cases} 1, & \text{если } t = 0, 1, 2^2 - 1, 2^3 - 1, \dots, \\ 0 & \text{в противном случае.} \end{cases}$$

Сравнить с теоретиче

10. Известно [98], $\gamma_t = \begin{cases} \text{результатом [149]: } l_t(\gamma) = \lfloor (t+1)/2 \rfloor, \\ \text{если } r = 2^n - \text{период } \gamma, n > 0, \end{cases}$

$$L(\gamma) = (\gamma_0, \dots, \gamma_{r/2-1}, \gamma_0, \dots), \quad R(\gamma) = (\gamma_{r/2}, \dots, \gamma_{r-1}, \gamma_{r/2+1}, \dots),$$

то

$$l(\gamma) = \begin{cases} l(L(\gamma)), & L(\gamma) = R(\gamma), \\ 2^{n-1} + l(L(\gamma) \oplus R(\gamma)), & L(\gamma) \neq R(\gamma). \end{cases}$$

Используя данный факт, разработать алгоритм и программу вычисления линейной сложности последовательностей де Брейна порядка n . Найти последовательности де Брейна с максимальной и минимальной линейной сложностью для $n = 2, 3, 4$. Проверить, что при $n \geq 3$ справедлива оценка

$$2^{n-1} + n \leq l(\gamma) \leq 2^n - 1.$$

11. Доказать, что $(p(\lambda))^{2^k} = p(\lambda^{2^k})$ и

$$\gamma_{t+2^k n} = \alpha_{n-1} \gamma_{t+2^k(n-1)} \oplus \dots \oplus \alpha_1 \gamma_{t+2^k} \oplus \alpha_0 \gamma_t$$

для любых $t, k = 0, 1, 2, \dots$.

12. Написать программу определения начального состояния искаженной ЛРП. Положить $n = 20$, $p(\lambda) = \lambda^{20} + \lambda^3 + 1$, $T = 2000$, $\varepsilon = \frac{3}{8}$.

13. Пороговая функция $g(x_1, \dots, x_d)$ симметрическая, ее значение не изменяется при перестановке значений переменных. Найти вид многочлена Жегалкина для g при $d = 3, 5, 7$. Доказать, что

$$\mathbf{P}\{g(x_1, \dots, x_d) = x_1\} = \frac{1}{2} + \frac{1}{2^d} \left(\begin{array}{c} d-1 \\ \frac{d-1}{2} \end{array} \right)$$

для случайного вектора (x_1, \dots, x_d) с равномерным на V_d распределением.

14. Пусть в G используется d генераторов $G^{(i)}$ m -последовательностей порядка n_i с минимальным периодом $r_i = 2^{n_i} - 1$, $i = 1, \dots, d$. Пусть r – минимальный период внутреннего состояния G , т. е. минимальное натуральное число, для которого

$$\left(\Gamma_r^{(1)}, \dots, \Gamma_r^{(d)} \right) = \left(\Gamma_0^{(1)}, \dots, \Gamma_0^{(d)} \right).$$

Доказать:

- 1) $r = \text{НОК } r_1, \dots, r_d$ при параллельном подключении генераторов;
- 2) $r = r_1 \text{НОК } r_2, r_3 / (2^m - 1)$, $m = \text{НОД } n_1 - 1, n_3$, для переключающего шаг-стоп генератора;
- 3) $r = \text{НОК } (r_1 + 1)/2, r_2$ для прореживающего генератора.

15. Разработать и реализовать программно пороговую криптосистему G , используя комбинацию из пяти генераторов ЛРП. Оценить период и линейную сложность последовательностей G .

16. Доказать, что на втором этапе атаки на генератор Беса – Пайпера последовательность $\gamma_0^{(3)}, \gamma_1^{(3)}, \dots$ определяется однозначно. Использовать тот факт, что многочлен $p_3(\lambda)$ содержит нечетное количество q ненулевых коэффициентов. Предложить модификацию атаки для случая четного q .

17. Используя результат задания 13, провести корреляционную атаку на разработанную в задании 15 пороговую криптосистему.

Г л а в а 12

БЛОЧНЫЕ КРИПТОСИСТЕМЫ

12.1. БЛОЧНОЕ ШИФРОВАНИЕ

Пусть Алиса и Боб обмениваются сообщениями по открытому каналу связи и пусть сообщениями являются слова в алфавите A . Алиса и Боб опасаются, что канал прослушивается противником Виктором и для обеспечения конфиденциальности своей переписки решают использовать шифрование с помощью блочных криптосистем. Термин «блочные» объясняется тем, что шифрование сообщения выполняется *блоками* – словами в алфавите A , которые имеют фиксированную длину n_b .

Прежде чем определить блочные криптосистемы, введем несколько понятий, связанных с подстановками. *Подстановкой* на множестве B называют биективное преобразование этого множества. Обозначим через $S(B)$ множество всех таких преобразований. *Композицией* $\sigma_1, \sigma_2 \in S(B)$ называется подстановка, которая обозначается $\sigma_2\sigma_1$ и действует по правилу

$$\sigma_2\sigma_1(x) = \sigma_2(\sigma_1(x)), \quad x \in B.$$

Множество $S(B)$ с операцией композиции является группой, которая называется *симметрической*. Единицей этой группы является тождественная подстановка $id: id(x) = x$. Если $\sigma_2\sigma_1 = id$, то подстановка σ_2 является *обратной* к σ_1 , что записывается как $\sigma_2 = \sigma_1^{-1}$.

Определение 12.1. *Блочной криптосистемой* называется семейство ключезависимых подстановок

$$F = \{F_\theta: \theta \in \Theta\} \subseteq S(A^{n_b}).$$

Здесь Θ – множество ключей, A^{n_b} – множество блоков сообщений, F_θ – подстановка зашифрования, действие которой определяется ключом θ и которой соответствует подстановка расшифрования F_θ^{-1} .

Как правило, $|\Theta| \ll |S(A^{n_b})|$, т. е. криптосистема представляет собой лишь малое подмножество допустимых подстановок. Желательно, чтобы элементы подмножества были «разбросаны» по $S(A^{n_b})$ как можно более хаотично. Это цель, которая ставится при проектировании современных блочных криптосистем.

Действие подстановок F_θ, F_θ^{-1} задается алгоритмически. Высокое быстродействие алгоритмов зашифрования и расшифрования является еще одной целью при проектировании F .

Алиса и Боб используют криптосистему F в следующем протоколе (интерактивном алгоритме). Кроме открытого канала связи, по которому передаются сообщения, в протоколе используется также секретный канал, по которому доставляются ключи. Алисе и Бобу помогает третья доверенная сторона – Трент. Тренту доверяются генерация и доставка ключей.

ПРОТОКОЛ ЗАЩИЩЕННАЯ ПЕРЕДАЧА ДАННЫХ

Предназначен для конфиденциальной передачи сообщений $X \in A^*$

Стороны: Алиса, Боб, Трент.

Каналы: открытый канал связи (ОКС), секретный канал связи (СКС).

Генерация ключей:

1. Трент: $\theta \xleftarrow{R} \Theta$.
2. Трент $\xrightarrow{\text{СКС}}$ Алиса: θ .
3. Трент $\xrightarrow{\text{СКС}}$ Боб: θ .

Передача X (длина X кратна n_b):

1. Алиса:
 1. Разбивает X на блоки $X_1, \dots, X_T \in A^{n_b}$;
 2. Выполняет зашифрование блоков: $Y_t \leftarrow F_\theta(X_t)$, $t = 1, \dots, T$;
 3. Формирует шифртекст $Y \leftarrow Y_1 \parallel \dots \parallel Y_T$.
 2. Алиса $\xrightarrow{\text{ОКС}}$ Боб: Y .
 3. Боб:
 1. Разбивает Y на блоки $Y_1, \dots, Y_T \in A^{n_b}$;
 2. Выполняет расшифрование блоков: $X_t \leftarrow F_\theta^{-1}(Y_t)$, $t = 1, \dots, T$;
 3. Собирает открытый текст $X \leftarrow X_1 \parallel \dots \parallel X_T$.
-

Здесь и далее символ \parallel обозначает конкатенацию (объединение) слов.

Если длина передаваемого сообщения X не обязательно кратна n_b , то Алиса и Боб должны предусмотреть обработку последнего (возможно, неполного) блока $X_T \in A^n$, $n \leq n_b$.

Простейшая схема обработки последнего блока состоит в следующем. Алиса дополняет X_T произвольными символами до слова длины n_b , зашифровывает X_T и дополнительно зашифровывает блок X_{T+1} , содержащий представление числа n словом из A^{n_b} . Боб расшифровывает последний блок, получает n , расшифровывает предпоследний блок и отбрасывает в нем $n_b - n$ заключительных символов.

Можно организовать обработку последнего блока так, чтобы дополнительный блок формировался не всегда. Пусть α и β – различные символы A . Алиса дописывает к X_T символ α (маркер), а затем минимальное число сим-

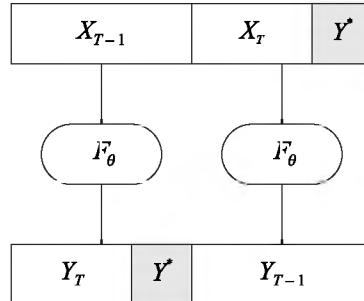


Рис. 12.1. Кража блока

волов β до получения слова, длина которого кратна n . Алиса зашифровывает X_T и, если блок X_T был полным, дополнительный блок $X_{T+1} = \alpha\beta\beta\dots\beta$. Боб после расшифрования и формирования открытого текста отбрасывает в нем заключительные символы β вплоть до маркера α , который также отбрасывается.

Третья схема обработки последнего блока может применяться при $T \geq 2$. Обработка организована так, что длина шифртекста совпадает с длиной открытого текста. Алиса зашифровывает два последних блока следующим образом (рис. 12.1):

$$Y_T \parallel Y^* \leftarrow F_\theta(X_{T-1}), \quad Y_{T-1} \leftarrow F_\theta(X_T \parallel Y^*),$$

а Боб следующим образом выполняет их расшифрование:

$$X_T \parallel Y^* \leftarrow F_\theta^{-1}(Y_{T-1}), \quad X_{T-1} \leftarrow F_\theta(Y_T \parallel Y^*).$$

Данную схему обработки называют *кражей блока*, имея в виду использование («похищение») Y^* для дополнения X_T .

12.2. ЗАДАЧИ КРИПТОАНАЛИЗА

Посмотрим на предыдущий протокол глазами противника Виктора. При оценке надежности крипtosистем предполагают, что противник обладает максимально возможным потенциалом. Поэтому, во-первых, считают, что Виктор точно знает устройство F , ему неизвестен только ключ θ , который доставляется по секретному каналу связи. Данное предположение есть известный в криптологии *принцип Керкгоффса* – надежность крипtosистемы определяется лишь секретностью ключа. Во-вторых, Виктор прослушивает открытый канал связи и таким образом перехватывает все блоки шифртекста Y_t . В-третьих, Виктор знает полностью или частично некоторые блоки открытого текста X_t или даже имеет возможность выбирать эти блоки.

По доступным данным, которые называются *шифрматериалом*, Виктору требуется решить задачу

C1: определить ключ θ подстановки F_θ .

Зная θ , Виктор может определить блок открытого текста $X_{T+1} = F_\theta^{-1}(Y_{T+1})$ по любому перехваченному блоку шифртекста $Y_{T+1} = F_\theta(X_{T+1})$. Вообще говоря, для определения X_{T+1} Виктору не обязательно находить ключ, ему достаточно решить задачу

C2: построить алгоритм нахождения X_{T+1} по заданному $Y_{T+1} = F_\theta(X_{T+1})$.

Чтобы решить поставленные задачи, Виктору требуется найти слабые места криптосистемы. Ими могут быть свойства, которыми, как правило, обладают подстановки F и, как правило, не обладают подстановки $S(A^{n_b})$. Всякое такое свойство позволяет отличить подстановку криптосистемы от других подстановок, и поэтому поиск свойств может быть сформулирован в виде задачи

C3: считая, что шифрматериал получен с помощью подстановки $\sigma \in S(A^{n_b})$, определить, является эта подстановка элементом F или нет.

Задачу **C3** можно интерпретировать как поддержку принципа Керкгоффса. Действительно, решив задачу Виктор узнает, используют ли Алиса и Боб криптосистему F или нет.

Задача **C1** является самой сложной (и практически значимой), задача **C3** – самой простой. Как правило, если найден способ решения одной задачи, то появляются подходы к решению всех остальных.

Алгоритмы решения задач криптоанализа называются *атаками*. Сложность атаки, как и любого другого алгоритма, характеризуется временем и памятью (см. п. 8.5). Атака может быть вероятностной, и тогда появляется еще одна характеристика сложности – вероятность успеха (см. п. 8.6). Криптографическую специфику атаки характеризует *объем шифрматериала* – количество T блоков шифртекста (и, возможно, открытого текста), требуемых для проведения атаки.

В зависимости от качества шифрматериала выделяют следующие типы атак:

- 1) известны $\{Y_t\}$ и свойства открытого текста $\{X_t\}$ (*атака при известном шифртексте*);
- 2) известны $\{X_t\}$ и $\{Y_t\}$ (*атака при известном открытом тексте*);
- 3) можно выбрать $\{X_t\}$ и получить $\{Y_t\}$ (*атака при выбранном открытом тексте*);
- 4) можно выбирать X_t , зная Y_1, \dots, Y_{t-1} (*атака при выбираемом открытом тексте*).

Условия атак последних типов кажутся искусственными. Это действительно так, если считать, что атаки применяются к протоколу из предыдущего пункта. Однако, как демонстрируют следующие примеры, атаки имеют практическое значение, если рассмотреть расширения этого протокола.

Пример 12.1 (GSM). В сетях связи GSM второго поколения речевые данные оцифровываются. Каждым 18,4 мс разговора соответствует двоичное слово длиной 184. Для противодействия помехам в канале связи слово (как вектор-строка) умножается на двоичную матрицу размера 184×456 . В результате получается кодовое слово X , которое обладает структурными особенностями: имеется $456 - 184$ независимых линейных комбинаций символов X , которые обязательно обращаются в $0 \bmod 2$. Кодовое слово X разбивается на 4 фрейма – слова длиной 114. Каждый фрейм зашифровывается перед отправкой в канал связи. Виктор, который перехватывает зашифрованные фреймы, знает о структурных особенностях соответствующего открытого текста X (хотя не располагает информацией о самих речевых данных).

Пример 12.2 (формат). Сообщение X представляет собой файл определенного формата. В частности, X всегда начинается фиксированным заголовком X_1 , известным Виктору.

Пример 12.3 («Энигма»). Во время Второй мировой войны британская специальная служба обрабатывала шифрматериал немецкой шифровальной машины «Энигма». Для организации атаки при выбранном открытом тексте англичане по агентурным каналам доводили в немецкие подразделения информацию (ложную или правдивую) о наличии мин в тех или иных районах. Последующие сообщения немцев обязательно содержали слово «Minen» (мины, нем.).

Пример 12.4 (пограничные шифраторы). Имеются два сегмента корпоративной сети, соединенные открытым каналом. На границах сегментов установлены шифраторы «Алиса» и «Боб», которые выполняют шифрование данных обмена. Виктор является пользователем сети, может выбрать любой открытый текст X_t для передачи в другой сегмент и перехватить соответствующий шифртекст Y_t .

12.3. БЛОЧНО-ИТЕРАЦИОННЫЕ КРИПТОСИСТЕМЫ

К. Шенон предложил строить подстановки F_θ как многократные композиции преобразований усложнения и перемешивания. Преобразования усложнения отвечают за установление сложных зависимостей между отдельными символами шифруемых данных и ключа, преобразования перемешивания распространяют эти зависимости по всему блоку шифруемых данных. Принцип Шеннона реализован в блочно-итерационных криптосистемах. Принцип оказался чрезвычайно плодотворным – все современные блочные криптосистемы являются блочно-итерационными.

Блочно-итерационная криптосистема F задается следующими элементами:

- 1) число *тактов* d ;
- 2) множество *тактовых ключей* K ;

3) алгоритм KS (от англ. Key Schedule, расписание ключей), который по ключу $\theta \in \Theta$ строит тактовые ключи $\kappa_1, \kappa_2, \dots, \kappa_d \in K$;

4) семейство тактовых подстановок $\Sigma = \{\Sigma_\kappa : \kappa \in K\} \subseteq S(A^{nb})$.

Подстановки зашифрования и расшифрования определяются по правилам:

$$F_\theta = \Sigma_{\kappa_d} \dots \Sigma_{\kappa_2} \Sigma_{\kappa_1}, \quad F_\theta^{-1} = \Sigma_{\kappa_1}^{-1} \dots \Sigma_{\kappa_{d-1}}^{-1} \Sigma_{\kappa_d}^{-1}.$$

Правила означают, что зашифрование открытого текста $X \in A^{nb}$ состоит в последовательном применении тактовых подстановок $\Sigma_{\kappa_1}, \Sigma_{\kappa_2}, \dots, \Sigma_{\kappa_d}$. Наоборот, расшифрование шифртекста $Y \in A^{nb}$ состоит в последовательном применении обратных тактовых подстановок $\Sigma_{\kappa_d}^{-1}, \Sigma_{\kappa_{d-1}}^{-1}, \dots, \Sigma_{\kappa_1}^{-1}$.

Правила можно задать алгоритмически.

АЛГОРИТМ БЛОЧНО-ИТЕРАЦИОННОЕ ЗАШИФРОВАНИЕ

Вход: $X \in A^{nb}$ – открытый текст, $\theta \in \Theta$ – ключ.

Выход: $Y \in A^{nb}$ – шифртекст.

Шаги:

1. $(\kappa_1, \dots, \kappa_d) \leftarrow \text{KS}(\theta)$.
 2. $Y \leftarrow X$.
 3. Для $i = 1, \dots, d$: $Y \leftarrow \Sigma_{\kappa_i}(Y)$.
 4. Возвратить Y .
-

АЛГОРИТМ БЛОЧНО-ИТЕРАЦИОННОЕ РАСШИФРОВАНИЕ

Вход: $Y \in A^{nb}$ – шифртекст, $\theta \in \Theta$ – ключ.

Выход: $X \in A^{nb}$ – открытый текст.

Шаги:

1. $(\kappa_1, \dots, \kappa_d) \leftarrow \text{KS}(\theta)$.
 2. $X \leftarrow Y$.
 3. Для $i = 1, \dots, d$: $X \leftarrow \Sigma_{\kappa_i}^{-1}(X)$.
 4. Возвратить X .
-

Тактовые подстановки Σ_κ имеют, как правило, простое строение и состоят в замене и перестановке символов подлежащего преобразованию слова. Однако многократная композиция таких подстановок определяет сложную зависимость между открытым текстом, шифртекстом и ключом.

Существует множество модификаций описанной блочно-итерационной конструкции. Приведем некоторые из них.

Неоднородные такты. Действие тактовых подстановок определяется не только ключом, но и номером такта: $F_\theta = \Sigma_{d, \kappa_d} \dots \Sigma_{2, \kappa_2} \Sigma_{1, \kappa_1}$.

Дополнительные бесключевые подстановки. Перед первым тактом и после последнего применяются дополнительные бесключевые подстановки $\tau_1, \tau_2 \in S(A^{n_b})$: $F_\theta = \tau_2 \Sigma_{\kappa_d} \dots \Sigma_{\kappa_1} \tau_1$. Например, в криптосистемах Фейстеля $\tau_1 = id$, а τ_2 состоит в перестановке половинок блоков из A^{n_b} (n_b – четное).

Отбеливание. На A^{n_b} вводится групповая операция $+$, по θ строится дополнительный ключ $\kappa_{d+1} \in A^{n_b}$ и на последнем шаге алгоритма зашифрования вместо Y возвращается $Y + \kappa_{d+1}$.

При атаках на блочно-итерационные криптосистемы у Виктора имеется несколько перспективных возможностей. Во-первых, Виктору не обязательно определять ключ θ , достаточно найти тактовые ключи $\kappa_1, \kappa_2, \dots, \kappa_d$, т. е. вместо задачи **C1** решить задачу **C2**. Во-вторых, тактовые ключи можно определить последовательно: сначала κ_d , затем κ_{d-1} и т. д. При этом реализуется важный криптоаналитический принцип: *divide et impera* (разделяй и властвуй, лат.). Определение каждого следующего тактового ключа является более простой задачей, чем предыдущего. Труднее всего определить κ_d . Для этого Виктор может решить задачу **C3** для подстановки $F'_\theta = \Sigma_{\kappa_{d-1}} \dots \Sigma_{\kappa_2} \Sigma_{\kappa_1}$, т. е. найти некоторое отличительное свойство этой подстановки. Затем Виктор проверяет всевозможные ключи-кандидаты $\hat{\kappa}_d$ и в качестве искомой оценки κ_d выбирает тот из них, на котором для подстановки $\hat{F}'_\theta = \Sigma_{\hat{\kappa}_d}^{-1} F_\theta$ найденное свойство проявляется в наибольшей степени.

12.4. ОПЕРАЦИИ НАД ДВОИЧНЫМИ СЛОВАМИ

В современных блочно-итерационных криптосистемах тексты и ключи являются двоичными словами. Откажемся от рассмотрения неиспользуемых на практике вариантов и случаев, и везде далее будем считать, что $A = \{0, 1\}$.

Построение криптосистемы сводится к организации преобразований усложнения и перемешивания над двоичными словами. Слова можно объединять, разбивать на блоки, блоки слов можно менять местами или заменять на другие блоки. Однако перечисленных операций может быть недостаточно. Для расширения набора преобразований удобно считать, что слова представляют элементы некоторых алгебраических структур, и задействовать операции этих структур. Рассмотрим распространенные представления.

Слова как векторы. Слово $a_1 a_2 \dots a_n \in \{0, 1\}^n$ представляет вектор

$$a = (a_1, a_2, \dots, a_n) \in \mathbb{F}_2^n.$$

Векторы одинаковой размерности можно складывать: если $b = (b_1, b_2, \dots, b_n)$, то

$$a + b = (a_1 + b_1, a_2 + b_2, \dots, a_n + b_n).$$

Сложение выполняется в поле \mathbb{F}_2 , т. е. по модулю 2. Такое сложение в криптографии часто обозначается знаком \oplus («круглый плюс»).

Кроме этого, векторы можно умножать на матрицы над полем \mathbb{F}_2 . Умножение на перестановочную матрицу задает перестановку координат вектора. Особый вид перестановки – циклический сдвиг. Через $a \lll r$ будем обозначать результат циклического сдвига вектора a на r позиций влево.

Через $a \cdot b$ будем обозначать скалярное произведение векторов a и b :

$$a \cdot b = a_1 b_1 + a_2 b_2 + \dots + a_n b_n,$$

через $w(a)$ – вес Хэмминга (число ненулевых координат) вектора a . Пусть $\mathbf{0}$ – нулевой вектор.

Слова как числа. Слово $a_1 a_2 \dots a_n$ представляет число

$$a = a_1 2^{n-1} + a_2 2^{n-2} + \dots + a_n.$$

Наоборот, всякое число $a \in \{0, 1, \dots, 2^n - 1\}$ представляется двоичным словом длины n . Это слово будем обозначать через $\langle a \rangle_n$.

Число a интерпретируется как элемент кольца \mathbb{Z}_{2^n} . Это кольцо составлено из целых от 0 до $2^n - 1$, их сложение и умножение выполняется по модулю 2^n . Умножение в \mathbb{Z}_{2^n} редко используется при построении блочных криптосистем, а вот сложение – достаточно часто. Операцию сложения принято обозначать знаком \boxplus («квадратный плюс»), операцию вычитания – знаком \boxminus («квадратный минус»).

Слова как элементы поля характеристики 2. Слово $a_1 a_2 \dots a_n$ представляет многочлен

$$a(\lambda) = a_1 \lambda^{n-1} + a_2 \lambda^{n-2} + \dots + a_n \in \mathbb{F}_2[\lambda].$$

Этот многочлен интерпретируется как элемент факторкольца $\mathbb{F}_2[\lambda]/(f(\lambda))$, где $f(\lambda) \in \mathbb{F}_2[\lambda]$ – неприводимый многочлен степени n . Элементами факторкольца являются многочлены, степени которых меньше n . Эти элементы складываются и умножаются как обычные многочлены, произведение дополнительно приводится по модулю f . Поскольку f – неприводим, факторкольцо $\mathbb{F}_2[\lambda]/(f(\lambda))$ является полем. Это поле состоит из 2^n элементов. Все такие поля изоморфны друг другу и представляют одно и то же поле \mathbb{F}_{2^n} : $\mathbb{F}_2[\lambda]/(f(\lambda)) \cong \mathbb{F}_{2^n}$ (см. п. 3.14).

В поле появляется дополнительная операция умножения, которая обозначается знаком $*$ или, как обычно, опускается. Сложение слов как элементов \mathbb{F}_{2^n} эквивалентно сложению слов как элементов \mathbb{F}_2^n .

Элементы \mathbb{F}_{2^n} – это в общем случае абстрактные объекты, не обязательно многочлены. Тем не менее всякий элемент $a \in \mathbb{F}_{2^n}$ может быть представлен вектором $(a_1, a_2, \dots, a_n) \in \mathbb{F}_2^n$ и далее словом $a_1 a_2 \dots a_n \in \{0, 1\}^n$. Для этого можно выбрать базис $\alpha_1, \alpha_2, \dots, \alpha_n$ поля \mathbb{F}_{2^n} как векторного пространства над \mathbb{F}_2 и записать a в виде

$$a = a_1 \alpha_1 + a_2 \alpha_2 + \dots + a_n \alpha_n.$$

Координаты a_i можно найти по формуле

$$a_i = \text{Tr}(\alpha_i^* a), \quad i = 1, 2, \dots, n.$$

Здесь $\text{Tr}: \mathbb{F}_{2^n} \rightarrow \mathbb{F}_2$, $x \mapsto x + x^2 + x^{2^2} + \dots + x^{2^{n-1}}$ – функция абсолютного следа, а $\{\alpha_i^*\}$ – базис \mathbb{F}_{2^n} над \mathbb{F}_2 , *дualъный* к $\{\alpha_i\}$:

$$\text{Tr}(\alpha_i \alpha_j^*) = \begin{cases} 1, & i = j, \\ 0, & \text{в противном случае.} \end{cases}$$

Функция Tr является линейным отображением $\mathbb{F}_{2^n} \rightarrow \mathbb{F}_2$, где \mathbb{F}_{2^n} и \mathbb{F}_2 рассматриваются как векторные пространства над \mathbb{F}_2 . Более того, любое линейное отображение $L: \mathbb{F}_{2^n} \rightarrow \mathbb{F}_2$ имеет вид

$$L(x) = \text{Tr}(bx), \quad b \in \mathbb{F}_{2^n}.$$

В частности, если $a \in \mathbb{F}_2^n$, а x интерпретируется и как элемент поля, и как вектор, то

$$a \cdot x = \text{Tr}(bx)$$

при подходящем выборе $b \in \mathbb{F}_{2^n}$.

Слова как элементы простого поля. Если $p = 2^n + 1$ – простое, то слова из $\{0, 1\}^n$ представляют элементы мультиликативной группы простого поля $\mathbb{F}_p = \{1, 2, \dots, p - 1\}$. Ненулевые слова-как-числа представляют элементы $1, 2, \dots, p - 2$ этой группы, а нулевое слово – элемент $p - 1 = 2^n$. Умножение в \mathbb{F}_p^* обозначается знаком \odot («круглая точка») или опускается.

Числа p указанного выше вида называются *простыми Ферма*. На сегодняшний день известно 5 таких простых: 3, 5, 17, 257, 65537.

Обозначения для алгебраических структур, элементы которых представляют двоичные слова, будем переносить на сами слова, и от них распространять на все остальные структуры. Можно сказать, что множество двоичных является общей платформой для всех структур. Будем подчеркивать связь между структурами знаком \sim : $\{0, 1\}^n \sim \mathbb{F}_2^n \sim \mathbb{Z}_{2^n} \sim \mathbb{F}_{2^n} \sim \mathbb{F}_{2^n+1}^*$.

Пример 12.5. В крипtosистеме AES *октеты* (слова длины 8) отождествляются с векторами \mathbb{F}_2^8 и элементами поля

$$\mathbb{F}_{2^8} \cong \mathbb{F}_2[\lambda]/(\lambda^8 + \lambda^4 + \lambda^3 + \lambda + 1).$$

Используемый здесь неприводимый многочлен даже получил именное название: *многочлен AES*. Октеты кодируются числами из \mathbb{Z}_{2^8} , представленными в шестнадцатеричной системе счисления. Например,

$$\begin{aligned} \text{слово } 01010111 &\sim \text{вектор } (0, 1, 0, 1, 0, 1, 1, 1) \sim \text{число } 57_{16} \sim \\ &\sim \text{элемент поля } \lambda^6 + \lambda^4 + \lambda^2 + \lambda + 1. \end{aligned}$$

12.5. БУЛЕВЫ ФУНКЦИИ И ОТОБРАЖЕНИЯ

12.5.1. *S*-блоки

Интересующие нас криптографические преобразования в конце концов оказываются отображениями $\{0, 1\}^n \rightarrow \{0, 1\}^m$ или $\mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$. Обозначим через $\mathcal{F}_{n,m}$ множество всех таких отображений. Множество $\mathcal{F}_n = \mathcal{F}_{n,1}$ является множеством булевых функций $\mathbb{F}_2^n \rightarrow \mathbb{F}_2$, а всякое отображение $\sigma \in \mathcal{F}_{n,m}$ можно задать m координатными булевыми функциями $\sigma_1, \dots, \sigma_m \in \mathcal{F}_n$ так, что

$$\sigma(x) = (\sigma_1(x), \dots, \sigma_m(x)), \quad x \in \mathbb{F}_2^n.$$

Для $a = (a_1, \dots, a_m) \in \mathbb{F}_2^m$ через $a \cdot \sigma$ будем обозначать линейную комбинацию координатных функций σ :

$$a \cdot \sigma(x) = a_1\sigma_1(x) + \dots + a_m\sigma_m(x).$$

При малых n и m отображения $\sigma \in \mathcal{F}_{n,m}$, которые используются для усложнения при построении блочных криптосистем, принято называть *S*-блоками. *S*-блоки, как правило, задаются таблично. Вычисление значения $\sigma(x)$ сводится к обращению к ячейке таблицы по индексу x .

S-блоки могут быть фиксированными (DES), могут быть частью ключа Θ (GOST) либо определяться по этому ключу (Blowfish). *S*-блоки могут быть сжимающими ($n > m$, DES), расширяющими ($n < m$, Blowfish) или сохраняющими размерность ($n = m$, GOST). Особый интерес представляют биективные *S*-блоки $\sigma \in S(\mathbb{F}_2^n) \subset \mathcal{F}_{n,n}$.

S-блоки должны обладать свойствами, затрудняющими применение тех или иных методов криптоанализа. Основными критериями выбора *S*-блоков являются: высокая нелинейность, большие степени координатных функций и их линейных комбинаций, малые значения в таблицах разностей. Эти критерии мы опишем в следующих пунктах, предварительно введя необходимый технический аппарат.

Существует три основных подхода к построению *S*-блоков:

1. *Случайная генерация*. Таблица значений *S*-блока заполняется случайно или псевдослучайно. Если криптографические характеристики полученного *S*-блока не являются удовлетворительными, то таблица генерируется заново. Как правило, с увеличением размерностей время генерации качественного *S*-блока быстро растет, что затрудняет применение данного подхода.

2. *Алгоритмические конструкции*. Задается алгоритм вычисления образов $\sigma(x)$, в котором используются эффективно реализуемые аппаратно и программно арифметические и логические операции, а также *S*-блоки меньшей размерности. Как правило, криптографические характеристики алгоритмических *S*-блоков не являются оптимальными.

3. Алгебраические конструкции. При построении S -блоков используются операции алгебраических структур, описанных в п. 12.4. Правила, определяющие действие S -блока, являются достаточно простыми, что позволяет провести теоретическое исследование криптографических характеристик. Правила выбирают так, чтобы характеристики были близки к оптимальным.

В пп. 12.5.9, 12.5.10 мы подробно рассмотрим две алгебраические конструкции S -блоков. Кроме этого, в следующем примере определяются модельные S -блоки, которые также построены с помощью алгебраических операций. Сразу скажем, что модельные S -блоки не являются оптимальными, это позволит нам впоследствии провести несколько атак на модельную крипосистему, в которых эти S -блоки используются.

Пример 12.6 (модельные S -блоки). S -блоки S_1 и S_2 действуют на $\{0, 1\}^4 \sim \mathbb{F}_2^4 \sim \mathbb{Z}_{16} \sim \mathbb{F}_{17}^*$ по правилам:

$$\begin{aligned} S_1(x) &= ((3^x \bmod 17) + 2) \bmod 16; \\ S_2(x) &= ((5^x \bmod 17) + 7) \bmod 16, \quad x = 0, 1, \dots, 15. \end{aligned}$$

В правых частях выражений неявно используются операции \odot и \boxplus . Поскольку 3 и 5 – примитивные элементы \mathbb{F}_{17} , модельные S -блоки являются биективными.

12.5.2. Аддитивный характер

Введем в рассмотрение функцию $\chi: \mathbb{F}_2 \rightarrow \mathbb{R}$, $c \mapsto (-1)^c$. В теории конечных полей эту функцию принято называть *аддитивным характером*, имея в виду, что

$$\chi(c_1 + c_2) = \chi(c_1)\chi(c_2), \quad c_i \in \mathbb{F}_2.$$

Лемма 12.1 (тождество для характера). Для $a \in \mathbb{F}_2^n$ выполняется

$$\frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} \chi(a \cdot x) = \begin{cases} 1, & a = \mathbf{0}, \\ 0 & a \neq \mathbf{0}. \end{cases}$$

Доказательство. Для нулевого a равенство очевидно. При $a \neq \mathbf{0}$ имеем

$$\begin{aligned} \sum_{x \in \mathbb{F}_2^n} \chi(a \cdot x) &= \sum_{x_1 \in \mathbb{F}_2} \dots \sum_{x_n \in \mathbb{F}_2} \chi(a_1 x_1) \dots \chi(a_n x_n) = \\ &= \left(\sum_{x_1 \in \mathbb{F}_2} \chi(a_1 x_1) \right) \dots \left(\sum_{x_n \in \mathbb{F}_2} \chi(a_n x_n) \right) = 0, \end{aligned}$$

поскольку $\chi(a_i \cdot 0) + \chi(a_i \cdot 1) = 1 - 1 = 0$, если $a_i = 1$. \square

Выражение в правой части доказанного тождества есть индикатор наступления события $\mathcal{E} = \{a = 0\}$. Этот индикатор будем обозначать через $\mathbf{I}\{\mathcal{E}\}$, распространяя обозначение на произвольные события \mathcal{E} .

12.5.3. Преобразование Уолша – Адамара

Преобразование Уолша – Адамара ставит в соответствие функции $f \in \mathcal{F}_n$ вещественнозначную функцию

$$\hat{f}(u) = \sum_{x \in \mathbb{F}_2^n} \chi(f(x))\chi(u \cdot x), \quad u \in \mathbb{F}_2^n.$$

Функцию f можно восстановить по \hat{f} :

$$\begin{aligned} \frac{1}{2^n} \sum_{u \in \mathbb{F}_2^n} \hat{f}(u)\chi(u \cdot x) &= \frac{1}{2^n} \sum_{u \in \mathbb{F}_2^n} \chi(u \cdot x) \sum_{y \in \mathbb{F}_2^n} \chi(f(y) + u \cdot y) = \\ &= \frac{1}{2^n} \sum_{u \in \mathbb{F}_2^n} \sum_{y \in \mathbb{F}_2^n} \chi(f(y) + u \cdot (x + y)) = \\ &= \frac{1}{2^n} \sum_{y \in \mathbb{F}_2^n} \chi(f(y)) \sum_{u \in \mathbb{F}_2^n} \chi(u \cdot (x + y)) = \chi(f(x)). \end{aligned}$$

Значения $\hat{f}(u)$ называются *коэффициентами Уолша – Адамара*. Коэффициенты удовлетворяют *равенству Парсеваля*:

$$\begin{aligned} \sum_{u \in \mathbb{F}_2^n} \hat{f}(u)^2 &= \sum_{u \in \mathbb{F}_2^n} \sum_{x, y \in \mathbb{F}_2^n} \chi(f(x) + f(y))\chi(u \cdot (x + y)) = \\ &= \sum_{x, y \in \mathbb{F}_2^n} \chi(f(x) + f(y)) \sum_{u \in \mathbb{F}_2^n} \chi(u \cdot (x + y)) = 2^n \sum_{x \in \mathbb{F}_2^n} \chi(0) = 2^{2n}. \end{aligned}$$

Из этого равенства следует, что $\max_{u \in \mathbb{F}_2^n} |\hat{f}(u)| \geq 2^{n/2}$.

12.5.4. Ортогональные системы

Функция $f \in \mathcal{F}_n$ называется *уравновешенной*, если она принимает значения 0 и 1 одинаковое количество раз. Другими словами, f – уравновешенная, если $\hat{f}(\mathbf{0}) = 0$.

Система булевых функций $\sigma_1, \dots, \sigma_m \in \mathcal{F}_n$ называется *ортогональной*, если для каждого вектора $(b_1, \dots, b_m) \in \mathbb{F}_2^m$ система уравнений

$$\sigma_1(x) = b_1, \dots, \sigma_m(x) = b_m$$

имеет ровно 2^{n-m} решений относительно x .

Очевидно, ортогональные системы существуют только при $m \leq n$. При $m = n$ ортогональной системе соответствует подстановка

$$\sigma = (\sigma_1, \dots, \sigma_n) \in S(\mathbb{F}_2^n).$$

Теорема 12.1 (критерий ортогональности). Система $\sigma_1, \dots, \sigma_m \in \mathcal{F}_n$ является ортогональной тогда и только тогда, когда для каждого ненулевого $a \in \mathbb{F}_2^m$ функция $a \cdot \sigma$ является уравновешенной.

Доказательство. Для $b = (b_1, \dots, b_m) \in \mathbb{F}_2^m$ через $N(b)$ обозначим число решений системы уравнений $\sigma_i(x) = b_i$, $i = 1, \dots, m$.

Если система $\{\sigma_i\}$ ортогональна, то для ненулевого $a \in \mathbb{F}_2^m$ выполняется

$$\begin{aligned} \widehat{a \cdot \sigma}(0) &= \sum_{x \in \mathbb{F}_2^n} \chi(a_1 \sigma_1(x)) \dots \chi(a_m \sigma_m(x)) = \sum_{b \in \mathbb{F}_2^m} N(b) \chi(a_1 b_1) \dots \chi(a_m b_m) = \\ &= 2^{n-m} \sum_{b \in \mathbb{F}_2^m} \chi(a \cdot b) = 0 \end{aligned}$$

и функция $a \cdot \sigma$ сбалансирована.

Обратно, если функция $a \cdot \sigma$ сбалансирована и $\widehat{a \cdot \sigma}(0) = 0$ для всех ненулевых a , то для любого b выполняется:

$$\begin{aligned} N(b) &= \sum_{x \in \mathbb{F}_2^n} \prod_{i=1}^m \left(\frac{1}{2} \sum_{a_i \in \mathbb{F}_2} \chi(a_i (\sigma_i(x) + b_i)) \right) = \\ &= \frac{1}{2^m} \sum_{a \in \mathbb{F}_2^m} \chi(a \cdot b) \sum_{x \in \mathbb{F}_2^n} \chi(a_1 \sigma_1(x) + \dots + a_m \sigma_m(x)) = 2^{n-m}. \quad \square \end{aligned}$$

Теорема показывает, что при изучении биективных S -блоков важно исследовать свойства уравновешенных булевых функций, поскольку координатные функции S -блока и их невырожденные линейные комбинации являются таковыми.

12.5.5. Нелинейность

Расстоянием Хэмминга между функциями $f, g \in \mathcal{F}_n$ называется число несовпадений их значений:

$$\rho(f, g) = \sum_{x \in \mathbb{F}_2^n} \mathbf{1}\{f(x) \neq g(x)\}.$$

Пусть $\mathcal{A}_n = \{f \in \mathcal{F}_n : \deg f \leq 1\}$ – множество аффинных функций от n переменных. Всякая функция $l \in \mathcal{A}_n$ имеет вид $l(x) = a \cdot x + b$, где $a \in \mathbb{F}_2^n$, $b \in \mathbb{F}_2$.

Нелинейностью функции $f \in \mathcal{F}_n$ называется расстояние от нее до множества аффинных функций:

$$\text{nl}(f) = \rho(f, \mathcal{A}_n) = \min_{l \in \mathcal{A}_n} \rho(f, l).$$

Расстояние между f и l определяется через коэффициент Уолша – Адамара:

$$\begin{aligned}\rho(f, l) &= 2^n - \sum_{x \in \mathbb{F}_2^n} \mathbf{1}\{f(x) = l(x)\} = 2^n - \frac{1}{2} \sum_{x \in \mathbb{F}_2^n} \sum_{c \in \mathbb{F}_2} \chi(c(f(x) + l(x))) = \\ &= 2^{n-1} - \chi(b) \frac{1}{2} \sum_{x \in \mathbb{F}_2^n} \chi(f(x) + a \cdot x) = 2^{n-1} - \chi(b) \frac{1}{2} \hat{f}(a).\end{aligned}$$

Поэтому

$$\text{nl}(f) = 2^{n-1} - \frac{1}{2} \max_{u \in \mathbb{F}_2^n} |\hat{f}(u)| \quad \text{и} \quad \text{nl}(f) \leq 2^{n-1} - 2^{n/2-1}.$$

Функции f , для которых достигается последняя оценка, существуют при четных n и называются *бент-функциями*. Достижимые оценки сверху для $\text{nl}(f)$ при нечетных $n \geq 9$ на сегодняшний день неизвестны. Неизвестны также максимальные значения нелинейности уравновешенных функций от $n \geq 11$ переменных. В табл. 12.1 представлены известные результаты, касающиеся максимальной нелинейности функций от малого числа переменных.

Таблица 12.1

Максимальные значения $\text{nl}(f)$

Функция f	Число переменных n						
	4	5	6	7	8	9	10
Произвольная	6	12	28	56	120	от 242 до 244	496
Уравновешенная	4	12	24	56	112	240	480

Линейной аппроксимацией для $\sigma \in \mathcal{F}_{n,m}$ называется пара векторов $a \in \mathbb{F}_2^n$ и $b \in \mathbb{F}_2^m$. При случайному равновероятном выборе x из \mathbb{F}_2^n определяется вероятность аппроксимации

$$\eta_{ab}(\sigma) = \mathbf{P}\{a \cdot x = b \cdot \sigma(x)\}.$$

Чем выше эта вероятность, тем с большей точностью можно предсказать линейную комбинацию $b \cdot \sigma(x)$ по линейной комбинации $a \cdot x$. Если вероятность близка к нулю, то можно изменить прогноз $b \cdot \sigma(x)$ на $b \cdot \sigma(x) + 1$, сделав вероятность прогнозирования близкой к 1. Таким образом, качество аппроксимации характеризует абсолютное значение величины $\varepsilon_{ab}(\sigma) = \eta_{ab}(\sigma) - 1/2$, которую принято называть *преобладанием*.

Преобладание связано с коэффициентами Уолша – Адамара:

$$\varepsilon_{ab}(\sigma) = \frac{1}{2^n} \rho(a \cdot x, b \cdot \sigma(x)) - \frac{1}{2} = \frac{1}{2^n} \left(2^{n-1} - \frac{1}{2} \widehat{b \cdot \sigma}(a) \right) - \frac{1}{2} = \frac{1}{2^{n+1}} \widehat{b \cdot \sigma}(a).$$

Отсюда

$$\max_{a, b \neq 0} |\varepsilon_{ab}(\sigma)| = \frac{1}{2} - \frac{1}{2^n} \text{nl}(\sigma),$$

где $\text{nl}(\sigma)$ – нелинейность σ :

$$\text{nl}(\sigma) = \min_{b \in \mathbb{F}_2^n \setminus \{0\}} \text{nl}(b \cdot \sigma).$$

При построении блочных крипtosистем отображения σ выбирают так, чтобы их нелинейность была максимально большой. При этом преобладания всевозможных нетривиальных линейных аппроксимаций для σ будут близки к 0.

Пример 12.7 (нелинейность модельных S -блоков). Для модельных S -блоков, описанных в примере 12.6, выполняется: $\text{nl}(S_1) = 2$, $\text{nl}(S_2) = 0$. Задекомпонуем следующие расстояния, которые нам пригодятся в дальнейшем:

$$\rho(0011 \cdot S_1(x), 1110 \cdot x + 1) = 2, \quad \rho(1100 \cdot S_2(x), 0001 \cdot x + 1) = 4.$$

12.5.6. Многочлены Жегалкина

Для $a, b \in \mathbb{F}_2^n$ обозначим $a^b = \prod_{i=1}^n a_i^{b_i}$. Здесь предполагается, что $0^0 = 1^0 = 1^1 = 1$ и $0^1 = 0$. Ясно, что $a^b = 1$ тогда и только тогда, когда $b_i \leq a_i$ для всех $i = 1, \dots, n$ (сравнение в обычном арифметическом смысле, т. е. $0 \leq 0$, $0 \leq 1$, $1 \leq 1$). Систему неравенств $b_i \leq a_i$ будем записывать просто как $b \leq a$. Если дополнительно $b \neq a$, то пишем $b < a$.

Функцию $f \in \mathcal{F}_n$ можно представить многочленом Жегалкина (в англоязычной литературе он называется алгебраической нормальной формой). Это многочлен от переменных $x = (x_1, \dots, x_n)$ над \mathbb{F}_2 , каждый его моном содержит любую из переменных x_i в степени не выше первой:

$$f(x) = \sum_{u \in \mathbb{F}_2^n} c_u x^u \in \mathbb{F}_2[x], \quad c_u \in \mathbb{F}_2.$$

Теорема 12.2. Каждая булева функция представляется единственным многочленом Жегалкина.

Доказательство. При подстановке в многочлене Жегалкина на места переменных x_i всевозможных значений из \mathbb{F}_2 получается некоторая функция $\mathbb{F}_2^n \rightarrow \mathbb{F}_2$. Имеется 2^{2^n} различных многочленов и столько же различных функций. Остается показать, что различные многочлены представляют различные функции.

Пусть, от противного, различные многочлены p_1 и p_2 представляют одну и ту же функцию f . Тогда ненулевой многочлен $p = p_1 - p_2$ представляет нулевую функцию. Пусть p содержит моном с k переменными и не содержит мономов с большим числом переменных. Считаем, не нарушая общности, что p содержит моном $x_1 \dots x_k$. Тогда

$$p(\underbrace{1, \dots, 1}_{k \text{ раз}}, 0, \dots, 0) = 1,$$

и p не может представлять нулевую функцию. \square

Коэффициенты многочлена Жегалкина можно найти по следующей формуле:

$$c_u = \sum_{x \leq u} f(x), \quad u \in \mathbb{F}_2^n.$$

Действительно,

$$\sum_{x \leq u} f(x) = \sum_{x \leq u} \sum_{v \leq x} c_v = \sum_{v \leq u} \sum_{x: v \leq x \leq u} c_v = c_u + \sum_{v < u} \underbrace{\sum_{x: v \leq x \leq u} c_v}_{\text{четное число слагаемых}} = c_u.$$

12.5.7. Алгебраическая степень

Алгебраической степенью функции f называется степень ее многочлена Жегалкина:

$$\deg(f) = \max_{u \in \mathbb{F}_2^n : c_u=1} w(u).$$

Если $f = 0$, то $\deg(f)$ полагается равной -1 .

Если многочлен Жегалкина для f содержит моном $x_1 x_2 \dots x_m$, то функция $g(x_1, x_2, \dots, x_m) = f(x_1, x_2, \dots, x_m, 0, \dots, 0)$ принимает значение 1 нечетное число раз. Функция g называется *сужением* f на подпространство $L = \{(x_1, x_2, \dots, x_m, 0, \dots, 0)\}$ линейного пространства \mathbb{F}_2^n . Рассмотрим все подпространства, сужения f на которые принимают значения 1 нечетное число раз, и пусть r – максимальная размерность этих подпространств ($r = -1$, если f – нулевая функция). Тогда $\deg(f) = r$, что является альтернативным определением алгебраической степени.

Алгебраическая степень не изменяется при обратном аффинном преобразовании переменных, т. е. является *аффинным инвариантом*: если $g \in \mathcal{F}_n$ получена из f по правилу

$$g(x) = f(xA + b), \quad A \in \mathrm{GL}_n(\mathbb{F}_2), \quad b \in \mathbb{F}_2^n,$$

то $\deg(f) = \deg(g)$. Действительно, при замене $x \mapsto y = xA + b$ степень многочлена $f(x)$ не увеличивается и $\deg(g) \leq \deg(f)$. Аналогично при замене $y \mapsto x = yA^{-1} - bA^{-1}$ не увеличивается степень $g(y)$ и $\deg(f) \leq \deg(g)$.

Криптографическое отображение $\sigma \in \mathcal{F}_{n,m}$ выбирают так, чтобы степени координатных функций были велики. Иногда требуют, чтобы величина

$$\min_{b \in \mathbb{F}_2^m \setminus \{\mathbf{0}\}} \deg(b \cdot \sigma)$$

также была большой. Эта величина называется *минимальной степенью* σ и обозначается через $\deg(\sigma)$.

Если $n \geq 2$ и f – уравновешенная, то $\deg(f) \leq n - 1$. Действительно, если неравенство нарушается и $\deg(f) = n$, то f принимает значение 1 нечетное число раз и не может быть уравновешенной.

По теореме 12.1 все невырожденные линейные комбинации координатных функций подстановки $\sigma \in S(\mathbb{F}_2^n)$ уравновешены. Поэтому при $n \geq 2$ справедлива оценка: $\deg(\sigma) \leq n - 1$.

Пример 12.8 (степени модельных S -блоков). Для модельных S -блоков, описанных в примере 12.6, выполняется: $\deg(S_1) = 2$, $\deg(S_2) = 1$. В частности,

$$0010 \cdot S_1(x) = x_2x_3 + x_2x_4 + x_4 + 1,$$

$$1111 \cdot S_2(x) = x_3 + x_4 + 1.$$

12.5.8. Таблицы разностей

Пусть $\sigma \in \mathcal{F}_{n,m}$ и пусть на \mathbb{F}_2^n и \mathbb{F}_2^m заданы групповые операции $+$ и $+'$ соответственно. Событие $\{\sigma(x + a) = \sigma(x) +' b\}$ означает, что прообразам x и $x + a$ с *разностью* (сдвигом друг относительно друга) a соответствуют образы $\sigma(x)$ и $\sigma(x + a)$ с разностью b .

Таблица разностей для σ – это матрица размера $2^n \times 2^m$ с элементами

$$\mu_{ab}(\sigma) = \sum_{x \in \mathbb{F}_2^n} \mathbf{1}_{\{\sigma(x + a) = \sigma(x) +' b\}}, \quad a \in \mathbb{F}_2^m, \quad b \in \mathbb{F}_2^m.$$

Элементам таблицы разностей можно придать вероятностный смысл. Пусть x, \tilde{x} – случайные независимые векторы с равномерным распределением на \mathbb{F}_2^n . Тогда величина $2^{-n} \mu_{ab}(\sigma)$ есть вероятность перехода от пары (x, \tilde{x}) с разностью a к паре $(\sigma(x), \sigma(\tilde{x}))$ с разностью b :

$$\frac{1}{2^n} \mu_{ab}(\sigma) = \mathbf{P} \{\sigma(x + a) = \sigma(x) +' b\} = \mathbf{P} \{\sigma(\tilde{x}) = \sigma(x) +' b \mid \tilde{x} = x + a\}.$$

Понятно, как устроена первая строка таблицы разностей: $\mu_{00}(\sigma) = 2^n$ и $\mu_{0b}(\sigma) = 0$ для всех $b \in \mathbb{F}_2^m \setminus \{0\}$. Вычертим эту строку и обозначим через $R(\sigma)$ максимальный элемент в оставшейся части таблицы:

$$R(\sigma) = \max_{\substack{a \in \mathbb{F}_2^n \\ a \neq 0}} \max_{b \in \mathbb{F}_2^m} \mu_{ab}(\sigma).$$

Криптографические отображения σ строят так, чтобы их разностные характеристики относительно определенных операций были невелики. Если $R(\sigma) \leq r$, то отображение σ называют *r-равномерным*.

В качестве + и +'', как правило, выбирают операции \oplus и \boxplus , которые часто используются при построении блочных криптосистем. При этом получают разностные характеристики четырех типов: $R_{\oplus\oplus}$, $R_{\oplus\boxplus}$, $R_{\boxplus\oplus}$ и $R_{\boxplus\boxplus}$. Здесь в индексе указывается сначала операция, выбранная в качестве +, а затем операция, выбранная в качестве +'.

Пусть $n = m$ и $\sigma \in S(\mathbb{F}_2^n)$. Построение *r*-равномерных подстановок с минимальным значением *r* является важной и не до конца решенной задачей. Ни одна из подстановок не может быть 1-равномерной. Действительно, если, например, $R_{\oplus\boxplus}(\sigma) = 1$, то уравнение $\sigma(x \oplus a) = \sigma(x) \boxplus b$ имеет ровно одно решение относительно x для любых ненулевых a и b . Это значит, что при фиксированном a отображение $x \mapsto \sigma(x \oplus a) \boxplus \sigma(x)$ является биекцией и, в частности, принимает нулевое значение, что невозможно. Подстановки с $r = 2$ известны при любом сочетании операций \oplus и \boxplus , но для определенных n . Например, неизвестно, существуют ли при четных n подстановки σ , для которых $R_{\oplus\oplus}(\sigma) = 2$.

Пример 12.9 (таблицы разностей модельных *S*-блоков). Для модельных *S*-блоков, описанных в примере 12.6, выполняется:

$$R_{\oplus\oplus}(S_1) = R_{\oplus\boxplus}(S_1) = 8, \quad R_{\boxplus\oplus}(S_1) = 4, \quad R_{\boxplus\boxplus}(S_1) = 2;$$

$$R_{\oplus\oplus}(S_2) = R_{\oplus\boxplus}(S_2) = 16, \quad R_{\boxplus\oplus}(S_2) = 4, \quad R_{\boxplus\boxplus}(S_2) = 2.$$

Зафиксируем следующий факт, который нам потребуется в дальнейшем:

$$\mathbf{P}\{S_1(x \oplus 1000) = S_1(x) \oplus 0001\} = \frac{1}{4}.$$

12.5.9. Инверсные *S*-блоки

В 1993 г. К. Нюберг и, независимо от нее, Т. Бес и К. Динг предложили строить биективные *S*-блоки, используя обращение (инверсию) в поле \mathbb{F}_{2^n} , $n \geq 2$. *Инверсный S-блок* $s \in S(\mathbb{F}_{2^n})$ определяется следующим образом:

$$s(x) = x^{2^n - 2} = \begin{cases} x^{-1}, & x \neq 0, \\ 0, & x = 0. \end{cases}$$

Элементы \mathbb{F}_{2^n} , как обычно, отождествляются с двоичными векторами или словами, при этом действие s естественным образом переносится на \mathbb{F}_2^n или $\{0, 1\}^n$.

Инверсные S -блоки обладают рядом замечательных свойств, и в связи с этим часто используются в блочных криптосистемах. Например, S -блок AES – это модификация инверсного S -блока при $n = 8$. Модификация состоит в применении дополнительных обратимых аффинных преобразований к прообразам и образам. Эти преобразования не изменяют криптографические характеристики $R_{\oplus\oplus}(s)$, $\deg(s)$ и $\text{nl}(s)$, которые мы рассмотрим ниже.

Теорема 12.3. Пусть $s \in S(\mathbb{F}_{2^n})$ – инверсный S -блок. Тогда $R_{\oplus\oplus}(s) = 4$ при четном n и $R_{\oplus\oplus}(s) = 2$ при нечетном n .

Доказательство. Пусть $a, b \in \mathbb{F}_{2^n}$ и $a \neq 0$. Рассмотрим уравнение

$$s(x + a) = s(x) + b \quad (12.1)$$

относительно x . Характеристика $R_{\oplus\oplus}(s)$ есть максимальное число решений данного уравнения при всевозможных фиксированных a, b . Подсчитаем максимальное число решений. Для этого рассмотрим два случая.

1. Пусть $b \neq a^{-1}$. Тогда $x = 0$ и $x = a$ не могут являться решениями (12.1). Считая, что $x \notin \{0, a\}$, можно записать (12.1) в следующем виде:

$$\frac{1}{x+a} + \frac{1}{x} + b = \frac{bx^2 + abx + a}{(x+a)x} = 0.$$

Последнее равенство равносильно уравнению $bx^2 + abx + a = 0$, которое имеет не более двух решений в поле \mathbb{F}_{2^n} .

2. Пусть $b = a^{-1}$. В этом случае решениями (12.1) являются $x = 0$ и $x = a$, а также, дополнительно, решения уравнения

$$a^{-1}x^2 + x + a = 0.$$

Умножим обе его части на a , заменим x на ay и получим уравнение $y^2 + y + 1 = 0$, все корни которого лежат в поле \mathbb{F}_{2^2} . (Действительно, корнями являются элементы $\lambda, \lambda + 1 \in \mathbb{F}_2[\lambda]/(\lambda^2 + \lambda + 1) \cong \mathbb{F}_{2^2}$.)

При четном n поле \mathbb{F}_{2^2} является подполем \mathbb{F}_{2^n} , а при нечетном n – нет. Таким образом, у исходного уравнения (12.1) имеется не более 4 решений при четном n и не более 2 решений при нечетном, причем верхние границы достижимы. \square

Теорема 12.4. Если $s \in S(\mathbb{F}_{2^n})$ – инверсный S -блок, то $\deg(s) = n - 1$.

Доказательство. 1. Рассмотрим функцию $f \in \mathcal{F}_n$, которая определяется по правилу $f(x) = \text{Tr}(bx^d)$, $b \in \mathbb{F}_{2^n}^*$, $d \in \{1, 2, \dots, 2^n - 1\}$. Пусть число d представлено двоичным словом и $r = w(d)$ – вес Хэмминга этого слова. Докажем, что $\deg(f) \leq r$.

Пусть $d = 2^{k_1} + 2^{k_2} + \dots + 2^{k_r}$, где $0 \leq k_1 < k_2 < \dots < k_r \leq n - 1$, и пусть $\alpha_1, \alpha_2, \dots, \alpha_n$ – базис \mathbb{F}_{2^n} над \mathbb{F}_2 , который отвечает за представление элементов \mathbb{F}_{2^n} векторами. Тогда

$$\begin{aligned} f(x) &= \text{Tr} \left(b \left(\sum_{i=1}^n x_i \alpha_i \right)^d \right) = \text{Tr} \left(b \prod_{j=1}^r \left(\sum_{i=1}^n x_i \alpha_i \right)^{2^{k_j}} \right) = \\ &= \text{Tr} \left(b \prod_{j=1}^r \left(\sum_{i=1}^n x_i \alpha_i^{2^{k_j}} \right) \right) = \sum_{1 \leq i_1, i_2, \dots, i_r \leq n} x_{i_1} x_{i_2} \dots x_{i_r} \text{Tr} \left(b \prod_{j=1}^r \alpha_{i_j}^{2^{k_j}} \right) \end{aligned}$$

и, следовательно, $\deg(f) \leq r$.

2. Любую функцию $g \in \mathcal{F}_n$ можно представить в виде $g(x) = \text{Tr}(P(x))$, где $P(x) \in \mathbb{F}_{2^n}[x]$, $\deg(P) \leq 2^n - 1$. Действительно, можно подобрать преобразование $\sigma: \mathbb{F}_{2^n} \rightarrow \mathbb{F}_{2^n}$ такое, что $g(x) = \text{Tr}(\sigma(x))$, а затем задать действие σ интерполяционным многочленом Лагранжа P .

3. Пусть $\deg(g) = n - 1$. Многочлен $P(x)$ представляет собой сумму членов bx^d , каждому из которых соответствует функция $\text{Tr}(bx^d)$ степени $\leq w(d)$. Найдется член cx^e с $w(e) = n - 1$, которому соответствует функция $\text{Tr}(cx^e)$ степени $n - 1$. Действительно, противное означает, что в многочлене Жегалкина для g мономы степени $n - 1$ индуцируются членом bx^{2^n-1} . Но этому члену соответствует функция

$$\text{Tr} \left(bx^{2^n-1} \right) = \begin{cases} 0, & x = 0, \\ \text{Tr}(b), & x \neq 0, \end{cases}$$

которая либо нулевая, либо имеет степень n . В обоих случаях получаем противоречие.

4. Показатель e представляется словом $11 \dots 1011 \dots 1$ из $n - 1$ единичных и одного нулевого символов. Пусть нулевой символ находится в k -й слева позиции. Умножение e на 2^k по модулю $2^n - 1$ состоит в циклическом сдвиге слова-представления на k позиций влево. После умножения будет получено слово $11 \dots 10$, которому соответствует число $2^n - 2$. Поэтому функция

$$h(x) = \text{Tr} \left(c^{2^k} x^{2^n-2} \right) = \text{Tr} \left((cx^e)^{2^k} \right) = \text{Tr}(cx^e)$$

имеет степень $n - 1$.

5. Невырожденные линейные комбинации координатных функций s имеют вид $s_b(x) = \text{Tr}(bx^{2^n-2})$, $b \in \mathbb{F}_{2^n}^*$. Заменив в любой из таких функций x на $bc^{-2^k}y$, получаем функцию $h(y)$. Замена переменных является обратимой линейной, и поэтому $\deg(s_b) = \deg(h) = n - 1$. Следовательно, $\deg(s) = \min_b \deg(s_b) = n - 1$. \square

Следующая теорема, которую мы приводим без доказательства, касается нелинейности инверсного S -блока. Теорема является следствием оценок для сумм Клостермана, полученных Ж. Лапо и Ж. Волфманом в 1990 г.

Теорема 12.5. *Пусть $s \in S(\mathbb{F}_{2^n})$ – инверсный S -блок. Если n – четное, то $\text{nl}(s) = 2^{n-1} - 2^{n/2}$. Если n – нечетное, то $\text{nl}(s)$ есть минимальное четное $\geq 2^{n-1} - 2^{n/2}$.*

12.5.10. Экспоненциальные S -блоки

Выберем примитивный элемент $\alpha \in \mathbb{F}_{2^n}$ и построим отображение $s: \mathbb{Z}_{2^n} \rightarrow \mathbb{F}_{2^n}$,

$$s(x) = \begin{cases} 0, & x = 0, \\ \alpha^x, & x \neq 0. \end{cases}$$

Поскольку $\alpha, \alpha^2, \alpha^3, \dots, \alpha^{2^n-1}$ суть все ненулевые элементы \mathbb{F}_{2^n} , s является биекцией. Заменяя прообразы и образы s векторами из \mathbb{F}_2^n , получаем экспоненциальный S -блок $s \in S(\mathbb{F}_2^n)$.

Пусть $f(\lambda) \in \mathbb{F}_2[\lambda]$ – минимальный многочлен элемента α . Этот многочлен имеет степень n и является примитивным. Через $L(f)$ обозначим множество всех двоичных последовательностей длиной 2^n , устроенных следующим образом: первый элемент каждой последовательности – 0, остальные элементы – отрезок линейной рекуррентной последовательности с характеристическим многочленом $f(\lambda)$. Используя свойства линейных рекуррентных последовательностей, легко проверить, что $L(f)$ – векторное пространство размерностью n над полем \mathbb{F}_2 , и всякая ненулевая последовательность $L(f)$ является уравновешенной – содержит одинаковое количество 0 и 1.

Невырожденные линейные комбинации координатных функций s имеют вид (см. п. 3.17):

$$s_b(x) = \begin{cases} 0, & x = 0, \\ \text{Tr}(b\alpha^x), & x \neq 0, \end{cases}$$

где $b \in \mathbb{F}_{2^n}^*$; x интерпретируется как вектор в левой части и как число – в правой. Пусть $s_{b,0}, s_{b,1}, \dots, s_{b,2^n-1}$ – таблица истинности s_b , т. е. последовательность значений функции на упорядоченных по возрастанию аргументах x (как числах). Эта последовательность является элементом $L(f)$. В частности, таблица истинности уравновешена, откуда с использованием теоремы 12.1 получаем еще одно доказательство биективности s .

Альтернативный способ построения подстановки s состоит в выборе в качестве таблиц истинности ее координатных функций некоторого базиса векторного пространства $L(f)$. Имеется $\varphi(2^n - 1)/n$ различных примитивных

многочленов степени n над полем \mathbb{F}_2 (здесь φ – функция Эйлера) и, следовательно, столько же различных множеств $L(f)$. Базис $L(f)$ можно выбрать $(2^n - 1)(2^n - 2) \dots (2^n - 2^{n-1})$ способами. Таким образом, имеется

$$\frac{\varphi(2^n - 1)}{n} (2^n - 1)(2^n - 2) \dots (2^n - 2^{n-1})$$

различных экспоненциальных S -блоков, действующих на \mathbb{F}_2^n .

Экспоненциальные S -блоки обладают достаточно высокой нелинейностью, а их характеристики $R_{\boxplus\oplus}(s)$ и $\deg(s)$, как показывают следующие теоремы, близки к оптимальным.

Теорема 12.6. *Пусть $s \in S(\mathbb{F}_2^n)$ – экспоненциальный S -блок, построенный с помощью примитивного элемента α . Если минимальный многочлен $f(\lambda)$ этого элемента не делит ни один из многочленов вида*

$$\lambda^{2^{n-1}} + \lambda^t + 1, \quad t = 1, \dots, 2^{n-1} - 1,$$

то $R_{\boxplus\oplus}(s) \leq 3$. В противном случае $R_{\boxplus\oplus}(s) = 4$.

Доказательство. Пусть a – ненулевой вектор \mathbb{F}_2^n и τ – число, которое соответствует этому вектору. Множество значений $s(x) \oplus s(x \boxplus a)$, $x \in \mathbb{F}_2^n$, является объединением $\{\alpha^\tau\} \cup \{\alpha^{2^n-\tau}\} \cup A \cup B$, где

$$A = \{\alpha^t \oplus \alpha^{t+\tau} : t = 1, \dots, 2^n - 1 - \tau\}, \quad B = \{\alpha^t \oplus \alpha^{2^n-\tau+t} : t = 1, \dots, \tau - 1\}.$$

Все элементы, определяющие каждое из множеств A и B , различны, следовательно, среди значений $s(x) \oplus s(x \boxplus a)$ не может быть более 4 одинаковых. Это означает, что $R_{\boxplus\oplus}(s) \leq 4$.

Более того, $R_{\boxplus\oplus}(s) = 4$ тогда и только тогда, когда для $\tau = 2^{n-1}$ (в этом случае $\alpha^\tau = \alpha^{2^n-\tau}$) и некоторого $t = 1, \dots, 2^{n-1} - 1$ выполняется равенство

$$\alpha^\tau = \alpha^t \oplus \alpha^{t+\tau}.$$

Но это означает, что минимальный многочлен $f(\lambda)$ делит некоторый многочлен, указанный в формулировке теоремы. \square

Теорема 12.7. *Если $s \in S(\mathbb{F}_2^n)$ – экспоненциальный S -блок, то*

$$\deg(s) \geq n - \lceil \log_2(n+1) \rceil.$$

Доказательство. Обозначим $\alpha_i = \alpha^{2^{i-1}}$, $i = 1, 2, \dots$. Пусть s_b – невырожденная линейная комбинация координатных функций s , соответствующая элементу $b \in \mathbb{F}_{2^n}^*$:

$$\begin{aligned} s_b(x_1, \dots, x_n) &= \text{Tr} \left(b \prod_{i=1}^n \alpha_i^{x_i} \right) + \text{Tr}(b) \prod_{i=1}^n (1 + x_i) = \\ &= \begin{cases} 0, & x_1 = \dots = x_n = 0, \\ \text{Tr}(b\alpha^x) & \text{в противном случае.} \end{cases} \end{aligned}$$

Введем оператор Δ_j взятия разности по переменной x_j :

$$\begin{aligned} \Delta_j s_b(x_1, \dots, x_n) &= s_b(x_1, \dots, x_{j-1}, 0, x_{j+1}, \dots, x_n) + \\ &\quad + s_b(x_1, \dots, x_{j-1}, 1, x_{j+1}, \dots, x_n) = \\ &= \text{Tr} \left(b(1 + \alpha_j) \prod_{\substack{1 \leq i \leq n \\ i \neq j}} \alpha_i^{x_i} \right) + \text{Tr}(b) \prod_{\substack{1 \leq i \leq n \\ i \neq j}} (1 + x_i). \end{aligned} \quad (12.2)$$

Многочлен $\Delta_j s_b$, домноженный на x_j , входит в качестве слагаемого в многочлен s_b . Если $\Delta_j s_b$ ненулевой, то

$$\deg(s_b) \geq \deg(\Delta_j s_b) + 1.$$

Пусть $g(x_1, \dots, x_k)$ – функция, полученная после применения к $s(x_1, \dots, x_n)$ последовательно операторов $\Delta_{k+1}, \dots, \Delta_n$, $1 \leq k < n$. Используя (12.2), получаем

$$g(x_1, \dots, x_k) = \text{Tr} \left(bc \prod_{i=1}^k \alpha_i^{x_i} \right) + \text{Tr}(b) \prod_{i=1}^k (1 + x_i),$$

где $c = \prod_{j=k+1}^n (1 + \alpha_j) = \sum_{t=0}^{2^{n-k}-1} \alpha_{k+1}^t = (1 + \alpha)(1 + \alpha_{k+1})^{-1} \neq 0$.

После удаления в таблице истинности функции g первого элемента $\text{Tr}(bc + b)$ остается отрезок

$$\text{Tr}(bc\alpha), \text{Tr}(bc\alpha^2), \dots, \text{Tr}(bc\alpha^{2^k-1})$$

ненулевой линейной рекуррентной последовательности с примитивным характеристическим многочленом степени n . Если $2^k - 1 \geq n$, то данный отрезок не может быть нулевым. Поэтому при $k = \lceil \log_2(n+1) \rceil$

$$\deg(s_b) \geq \deg(g) + n - k \geq n - k,$$

откуда и следует требуемый результат. \square

Следующая теорема дает критерий выбора α , при котором степени всех ненулевых координатных функций s достигают максимального значения.

Теорема 12.8. Пусть $s \in S(\mathbb{F}_{2^n})$ – экспоненциальный S -блок, построенный с помощью примитивного элемента α . Тогда $\deg(s) = n - 1$, только если элементы

$$a = \alpha(1 + \alpha)^{-1}, a^2, \dots, a^{2^n-1}$$

образуют базис \mathbb{F}_{2^n} над \mathbb{F}_2 .

Доказательство. Используем обозначения из доказательства предыдущей теоремы. Рассмотрим функции

$$\begin{aligned} g_j(x_j) &= \Delta_n \dots \Delta_{j+1} \Delta_{j-1} \dots \Delta_1 s_b(x_1, \dots, x_n) = \\ &= \text{Tr} \left(b \alpha_j^{x_j} \prod_{\substack{1 \leq i \leq n \\ i \neq j}} (1 + \alpha_i) \right) + \text{Tr}(b)(1 + x_j) = \\ &= \text{Tr} \left(b \alpha_j^{x_j} (1 + \alpha_j)^{-1} \right) + \text{Tr}(b)(1 + x_j). \end{aligned}$$

Последнее равенство справедливо в силу того, что

$$\prod_{i=1}^n (1 + \alpha_i) = \sum_{t=0}^{2^n - 1} \alpha^t = 1 + \sum_{c \in \mathbb{F}_{2^n}} c = 1.$$

Для функции g_j выполняется

$$\begin{aligned} g_j(0) &= \text{Tr} (b((1 + \alpha_j)^{-1} + 1)) = \\ &= \text{Tr} (b \alpha_j (1 + \alpha_j)^{-1}) = g_j(1). \end{aligned}$$

Обозначим $a_j = \alpha_j (1 + \alpha_j)^{-1} = a^{2^j}$. Теперь $\deg(s_b) = n - 1$ тогда и только тогда, когда $g_j(0) = \text{Tr}(ba_j) \neq 0$ для некоторого j , $1 \leq j \leq n$. Следовательно, $\deg(s) = n - 1$ тогда и только тогда, когда ядро гомоморфизма $\mathbb{F}_{2^n} \rightarrow \mathbb{F}_2^n$, $b \mapsto (\text{Tr}(ba_1), \dots, \text{Tr}(ba_n))$ состоит из единственного (нулевого) элемента. Но последнее означает, что a_1, \dots, a_n является базисом \mathbb{F}_{2^n} над \mathbb{F}_2 . \square

12.6. КРИПТОСИСТЕМЫ ПОДСТАНОВКИ-ПЕРЕСТАНОВКИ

Вернемся к блочно-итерационным криптосистемам и обсудим строение тактовых подстановок $\Sigma_\kappa \in S(\{0, 1\}^{n_b})$, $\kappa \in K$. В *криптосистемах подстановки-перестановки* множество тактовых ключей K совпадает с $\{0, 1\}^{n_b}$, длина блока n_b является произведением rtm , где r и m – натуральные числа, $r \geq 2$. Используются S -блоки $S_1, \dots, S_r \in S(\{0, 1\}^m)$ и преобразование $P \in S(\{0, 1\}^{n_b})$. Действие P состоит в обратимой перестановке символов слов-прообразов:

$$P(x_1 x_2 \dots x_{n_b}) = x_{\pi(1)} x_{\pi(2)} \dots x_{\pi(n_b)}, \quad \pi \in S(\{1, 2, \dots, n_b\}).$$

В соответствии с английскими терминами Substitution (подстановка) и Permutation (перестановка) криптосистемы подстановки-перестановки часто называют *SP-криптосистемами*.

Образы $\Sigma_\kappa(X)$ определяются в три этапа (рис. 12.2).

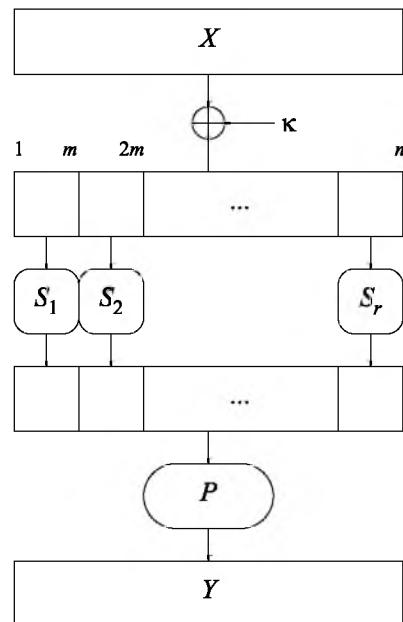


Рис. 12.2. Тактовая подстановка SP-криптосистемы

1. Прообраз X суммируется с тактовым ключом κ по правилу \oplus .
 2. К последовательным m -фрагментам суммы применяются S -блоки S_1, \dots, S_r .
 3. К объединению выходов S -блоков применяется преобразование перестановки P .
- Действие Σ_κ , а также обратной подстановки Σ_κ^{-1} , можно задать алгоритмически.

АЛГОРИТМ ПРИМЕНЕНИЕ Σ_κ

Вход: $X, \kappa \in \{0, 1\}^{n_b}$ ($n_b = rm$).

Выход: $Y = \Sigma_\kappa(X)$.

Шаги:

1. $Y \leftarrow X \oplus \kappa$.
2. Представить Y в виде $Y_1 \parallel \dots \parallel Y_r$.
3. Для $i = 1, \dots, r$: $Y_i \leftarrow S_i(Y_i)$.
4. $Y \leftarrow P(Y)$.
5. Возвратить Y .

АЛГОРИТМ ПРИМЕНЕНИЕ Σ_{κ}^{-1}

Вход: $Y, \kappa \in \{0, 1\}^{n_b}$ ($n_b = rm$).

Выход: $X = \Sigma_{\kappa}^{-1}(Y)$.

Шаги:

1. $X \leftarrow P^{-1}(Y)$.
 2. Представить X в виде $X_1 \parallel \dots \parallel X_r$.
 3. Для $i = 1, \dots, r$: $X_i \leftarrow S_i^{-1}(X_i)$.
 4. $X \leftarrow X \oplus \kappa$.
 5. Возвратить X .
-

Современные подходы к построению SP-криптосистем заключаются в замене перестановки P на преобразование A , состоящее в умножении слова-как-вектора на обратимую матрицу над полем \mathbb{F}_2 (не обязательно перестановочную). Например в криптосистеме *Serpent* преобразование A реализовано с помощью многократных сложений, сдвигов и перестановок частей исходного вектора. Криптосистемы, полученные заменой P на A , принято обозначать аббревиатурой SA.

Частным случаем SA-криптосистем являются *SQUARE-криптосистемы*. Здесь m -фрагменты X записываются в ячейки квадратной матрицы (отсюда *SQUARE* – квадрат, англ.). После сложения с ключом и подстановки на S -блоках выполняются линейные преобразования сначала всех строк матрицы (по отдельности), а затем всех столбцов.

12.7. КРИПТОСИСТЕМА AES

Криптосистема AES была разработана бельгийскими криптографами В. Рэйменом и Й. Дэмемом в рамках проводимого в США конкурса на разработку стандарта шифрования XXI в. Разработка Рэймена и Дэмена, которая первоначально называлась *Rijndael*, победила в конкурсе и была стандартизована в 2002 г. Сегодня это самая распространенная блочная криптосистема в мире.

Криптосистема построена по схеме *SQUARE*, длина блока $n_b = 128$, число тактов $d \in \{10, 12, 14\}$, множество ключей $\Theta \in \{\{0, 1\}^{128}, \{0, 1\}^{192}, \{0, 1\}^{256}\}$, множество тактовых ключей $K = \{0, 1\}^{128}$.

В AES октеты отождествляются с векторами \mathbb{F}_2^8 , элементами \mathbb{F}_{2^8} и числами \mathbb{Z}_{2^8} так, как это описано в примере 12.6. Для обработки блоков данных используется матрица размером 4×4 . В ячейки матрицы записываются октеты x_{ij} , $0 \leq i, j \leq 3$. К матрице применяются преобразования *AddRoundKey*, *SubBytes*, *ShiftRows*, *MixColumns*.

Действие преобразования **AddRoundKey** определяется тактовым ключом и состоит в сложении октетов матрицы с соответствующими октетами тактового ключа. Сложение выполняется по правилу \oplus .

Преобразование **SubBytes** состоит в одновременной замене всех октетов x_{ij} на $s(x_{ij})$. Здесь $s \in S(\{0, 1\}^8)$ – модифицированный инверсный S -блок (см. подпункт 12.5.9).

Под действием **ShiftRows** строка матрицы с номером $i = 0, 1, 2, 3$ сдвигается циклически влево на i позиций.

Самым сложным алгоритмически является преобразование **MixColumns**. Каждый из столбцов $(b_0, b_1, b_2, b_3)^T$ матрицы связывается с многочленом

$$b(\lambda) = b_3\lambda^3 + b_2\lambda^2 + b_1\lambda + b_0 \in \mathbb{F}_{2^8}[\lambda],$$

который умножается на фиксированный многочлен

$$a(\lambda) = 03_{16}\lambda^3 + 01_{16}\lambda^2 + 01_{16}\lambda + 02_{16} \in \mathbb{F}_{2^8}[\lambda].$$

Затем полученное произведение делится на многочлен $\lambda^4 + 1$:

$$a(\lambda)b(\lambda) = g(\lambda)(\lambda^4 + 1) + b'(\lambda), \quad g, b' \in \mathbb{F}_{2^8}[\lambda], \quad \deg b' < 4.$$

Коэффициенты остатка $b'(\lambda)$ определяют новый столбец матрицы – результат преобразования **MixColumns**.

Важно, что $a(\lambda)$ взаимно прост с $\lambda^4 + 1$: имеется многочлен $a^{-1}(\lambda) \in \mathbb{F}_{2^8}[\lambda]$ такой, что $a(\lambda)a^{-1}(\lambda) \equiv 1 \pmod{\lambda^4 + 1}$. Поэтому

$$b'(\lambda)a^{-1}(\lambda) = b(\lambda)a(\lambda)a^{-1}(\lambda) \equiv 1 \pmod{\lambda^4 + 1},$$

т. е. обратное преобразование определяется умножением на $a^{-1}(\lambda)$.

При зашифровании выполняются следующие преобразования:

Σ_{1, κ_1}	AddRoundKey SubBytes ShiftRows MixColumns
.....	
$\Sigma_{d-1, \kappa_{d-1}}$	AddRoundKey SubBytes ShiftRows MixColumns
Σ_{d, κ_d}	AddRoundKey SubBytes ShiftRows
Отбеливание	AddRoundKey

При расшифровании преобразования меняются на композиционно обратные и применяются в обратном порядке.

12.8. τ -ИНВОЛЮТИВНЫЕ ПОДСТАНОВКИ

Как мы только что видели, при расшифровании с помощью SP- или SA-криптосистем итерационные преобразования должны применяться в обратном порядке, прямые преобразования S_i , P или A должны заменяться на композиционно обратные. Это оказывается не всегда удобным. Например, при аппаратной реализации криптосистемы приходится хранить в устройстве как код программы зашифрования, так и код расшифрования; как таблицы S_i , так и таблицы S_i^{-1} .

Покажем, как организовать переключение между зашифрованием и расшифрованием, изменения только порядок следования тактовых ключей.

Определение 12.2. Пусть τ и σ – подстановки, которые действуют на одном и том же множестве. Подстановка σ называется τ -инволютивной, если $\sigma\tau\sigma = \tau$.

Вместо «*id*-инволютивная» будем говорить просто «инволютивная», как это принято в алгебре.

Теорема 12.9. Пусть преобразования зашифрования блочно-итерационной криптосистемы F имеют вид

$$F_\theta = \tau \Sigma_{\kappa_d} \dots \Sigma_{\kappa_2} \Sigma_{\kappa_1},$$

где $\tau \in S(\{0, 1\}^{n_b})$ – инволютивна, а тактовые подстановки $\Sigma_\kappa \in S(\{0, 1\}^{n_b})$ – τ -инволютивны при любом $\kappa \in K$. Тогда

$$F_\theta^{-1} = \tau \Sigma_{\kappa_1} \Sigma_{\kappa_2} \dots \Sigma_{\kappa_d}.$$

Доказательство.

$$\begin{aligned} F_\theta \tau \Sigma_{\kappa_1} \Sigma_{\kappa_2} \dots \Sigma_{\kappa_d} &= \\ &= \tau \Sigma_{\kappa_d} \dots \Sigma_{\kappa_2} (\Sigma_{\kappa_1} \tau \Sigma_{\kappa_1}) \Sigma_{\kappa_2} \dots \Sigma_{\kappa_d} = \\ &= \tau \Sigma_{\kappa_d} \dots (\Sigma_{\kappa_2} \tau \Sigma_{\kappa_2}) \dots \Sigma_{\kappa_d} = \dots = \tau \tau = id, \end{aligned}$$

что и требовалось доказать. \square

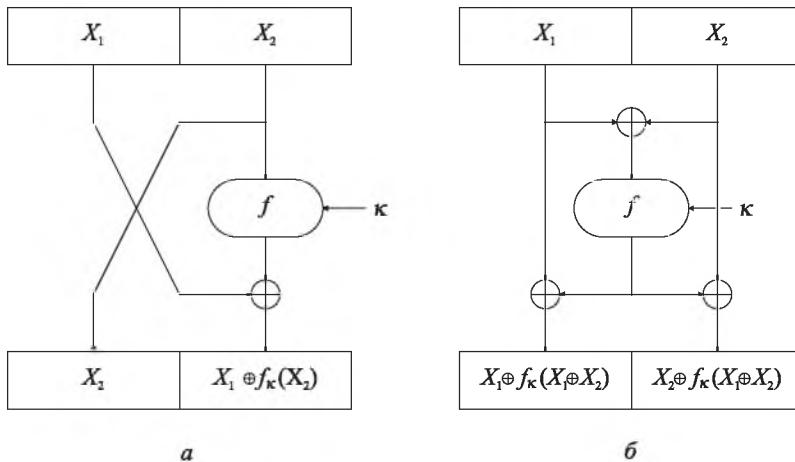
Для реализации подстановок F_θ и F_θ^{-1} , описанных в теореме, требуется использовать семейство τ -инволютивных тактовых подстановок. Рассмотрим две распространенные конструкции таких семейств. Будем считать, что $n_b = 2m$ – четное число, и использовать преобразования

$$f_\kappa: \{0, 1\}^m \rightarrow \{0, 1\}^m, \quad \kappa \in K,$$

которые назовем *тактовыми функциями*. Тактовые функции не обязательно биективны.

Подстановки Фейстеля. При разработке криптосистемы DES было решено использовать следующие подстановки, предложенные ранее X. Фейстелем (рис. 12.3, a):

$$\Sigma_\kappa(X_1 \parallel X_2) = X_2 \parallel (X_1 \oplus f_\kappa(X_2)), \quad X_i \in \{0, 1\}^m.$$

Рис. 12.3. τ -инволютивные подстановки

Пусть преобразование τ состоит в перестановке половинок слов из $\{0, 1\}^{2m}$: $\tau(X_1 \parallel X_2) = X_2 \parallel X_1$. Тогда

$$\begin{aligned}\Sigma_\kappa \tau \Sigma_\kappa (X_1 \parallel X_2) &= \\ &= \Sigma_\kappa \tau (X_2, X_1 \oplus f_\kappa(X_2)) = \\ &= \Sigma_\kappa (X_1 \oplus f_\kappa(X_2) \parallel X_2) = \\ &= X_2 \parallel X_1 = \tau(X_1 \parallel X_2).\end{aligned}$$

Следовательно, Σ_κ является τ -инволютивной подстановкой.

Подстановки Лай – Мэсси. При разработке криптосистемы IDEA С. Лай и Дж. Мэсси использовали следующие подстановки (рис. 12.3, б):

$$\Sigma_\kappa(X_1 \parallel X_2) = (X_1 \oplus f_\kappa(X_1 \oplus X_2) \parallel X_2 \oplus f_\kappa(X_1 \oplus X_2)), \quad X_i \in \{0, 1\}^m.$$

Подстановка Σ_κ является *id*-инволютивной (т. е. просто инволютивной) при любом выборе тактовой функции f_κ .

Вместо \oplus можно использовать другие операции, например, перейти к подстановке

$$\Sigma'_\kappa(X_1 \parallel X_2) = (X_1 \boxplus f_\kappa(X_1 \boxplus X_2) \parallel X_2 \boxminus f_\kappa(X_1 \boxplus X_2)),$$

которая также является инволютивной.

Подстановка Σ_κ сохраняет сумму половинок: если $\Sigma(X_1 \parallel X_2) = Y_1 \parallel Y_2$, $Y_i \in \{0, 1\}^m$, то $Y_1 \oplus Y_2 = X_1 \oplus X_2$. Поэтому Σ_κ целесообразно использовать в совокупности с другими преобразованиями, как это сделано в криптосистемах IDEA и Belt.

12.9. КРИПТОСИСТЕМЫ ФЕЙСТЕЛЯ

В криптосистемах, основанных на подстановках Фейстеля, шифрование выполняется с помощью следующего алгоритма.

АЛГОРИТМ ШИФРОВАНИЕ В КРИПТОСИСТЕМАХ ФЕЙСТЕЛЯ

Вход: $X \in \{0, 1\}^{2m}$, $\kappa_1, \dots, \kappa_d \in K$.

Выход: $Y \in \{0, 1\}^{2m}$.

Шаги:

1. $(Y_1 \parallel Y_2) \leftarrow X$.
2. Для $i = 1, \dots, d$: $(Y_1 \parallel Y_2) \leftarrow (Y_2 \parallel (Y_1 \oplus f_{\kappa_i}(Y_2)))$.
3. Возвратить $Y_2 \parallel Y_1$.

При зашифровании на вход алгоритма подаются тактовые ключи $(\kappa_1, \dots, \kappa_d) = \text{KS}(\theta)$, при расшифровании – ключи $(\kappa_d, \dots, \kappa_1)$. Для окончательного определения криптосистемы Фейстеля требуется уточнить расписание ключей KS и действие тактовых функций f_{κ} .

В табл. 12.2 приводятся характеристики некоторых известных криптосистем Фейстеля. У всех криптосистем длина блока $n_b = 64$.

Таблица 12.2
Криптосистемы Фейстеля

Криптосистема	n_b	d	Θ	K	S -блоки
DES (стандарт США в 1979 – 2005 гг.)	64	16	$\{0, 1\}^{56}$	$\{0, 1\}^{48}$	$S_i: \{0, 1\}^6 \rightarrow \{0, 1\}^4$ (постоянные)
GOST (стандарт СССР ГОСТ 28147, введен в 1989 г.)	64	32	$\{0, 1\}^{256}$	$\{0, 1\}^{32}$	$S_i: \{0, 1\}^4 \rightarrow \{0, 1\}^4$ (долговр. ключ)
Blowfish (США)	64	16	$\{0, 1\}^{40 \div 448}$	$\{0, 1\}^{32}$	$S_i: \{0, 1\}^8 \rightarrow \{0, 1\}^{32}$ (определяются по θ)

Тактовые функции $f_{\kappa}: \{0, 1\}^{32} \rightarrow \{0, 1\}^{32}$ криптосистем схематически представлены на рис. 12.4. На схеме, соответствующей DES, отображение $E: \{0, 1\}^{32} \rightarrow \{0, 1\}^{48}$ состоит в повторе и перестановке координат слова-пробраза, P – преобразование перестановки.

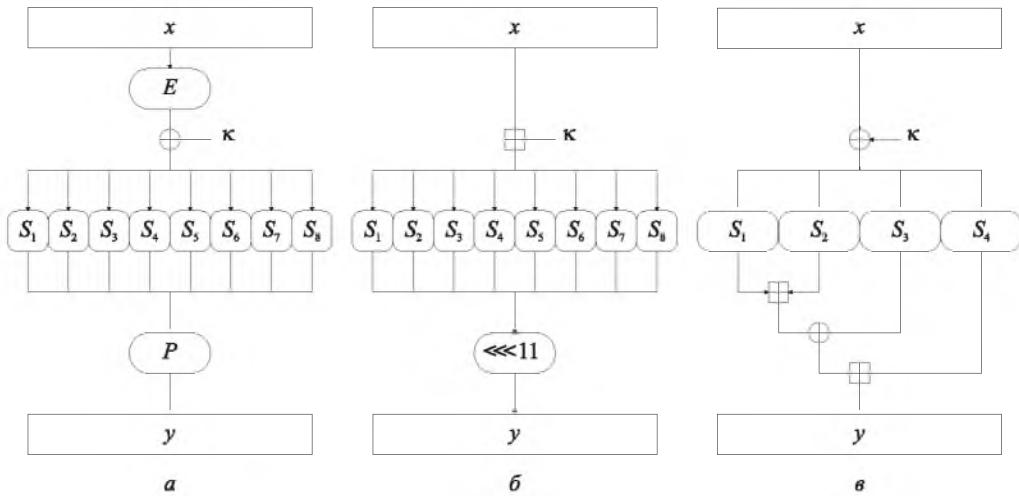


Рис. 12.4. Тактовые функции криптосистем Фейстеля: DES (а), GOST (б), Blowfish (в)

Пример 12.10 (модельная криптосистема G). Методы криptoанализа будем иллюстрировать примерами с модельной криптосистемой Фейстеля, названной нами G (рис. 12.5).

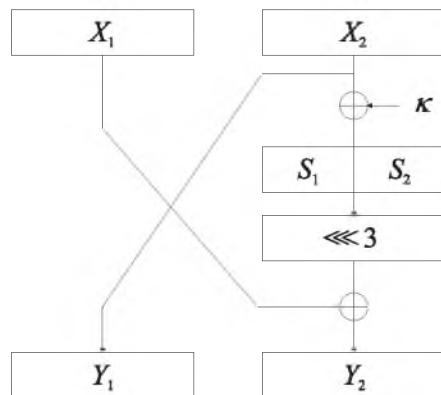


Рис. 12.5. Тактовая подстановка криптосистемы G

В этой криптосистеме $n_b = 16$, $d = 8$, $\Theta = \{0, 1\}^{32}$, $K = \{0, 1\}^8$. Алгоритм KS ставит в соответствие ключу

$$\theta = \theta_1 \parallel \theta_2 \parallel \theta_3 \parallel \theta_4, \quad \theta_i \in \{0, 1\}^8,$$

последовательность тактовых ключей

$$\kappa_1 = \theta_1, \quad \kappa_2 = \theta_2, \quad \kappa_3 = \theta_3, \quad \kappa_4 = \theta_4,$$

$$\kappa_5 = \theta_1, \quad \kappa_6 = \theta_2, \quad \kappa_7 = \theta_3, \quad \kappa_8 = \theta_4.$$

В тактовой функции $f_\kappa: \{0, 1\}^8 \rightarrow \{0, 1\}^8$ используются модельные S -блоки S_1 и S_2 , описанные в примере 12.6. Тактовые функции действуют следующим образом:

$$f_\kappa(x) = (S_1(z_1) \parallel S_2(z_2)) \lll 3,$$

где $z_i \in \{0, 1\}^4$, $z_1 \parallel z_2 = x \oplus \kappa$.

12.10. КРИПТОСИСТЕМА Belt

Криптосистема **Belt**, разработанная в 2001 г., определена в государственном стандарте Республики Беларусь СТБ 34.101.31-2011 «Информационные технологии и безопасность. Защита информации. Криптографические алгоритмы шифрования и контроля целостности». Этот стандарт был введен как предварительный в 2007 г. и получил статус окончательного 4 годами позже.

Belt не является ни SP-криптосистемой, ни криптосистемой Фейстеля, хотя и содержит конструктивные элементы этих систем. Длина блока $n_b = 128$, число тактов $d = 8$, множество ключей $\Theta = \{0, 1\}^{256}$.

В **Belt** используются операции \oplus , \boxplus , \boxminus , циклические сдвиги, подстановки на S -блоках. При представлении чисел двоичными словами применяются соглашения «от младших к старшим» (little endian), действующие для большинства современных микропроцессоров. Согласно этим соглашениям первый октет слова представляет младший байт числа, последний октет – старший. Например, слову

10110001 10010100 10111010 11001000

соответствуют байты $B1_{16}$, 94_{16} , BA_{16} , $C8_{16}$ и число $C8BA94B1_{16}$ (в шестнадцатеричной записи). Удобно считать, что при переходе от слов к числам и назад октеты слов неявно разворачиваются. Неявный разворот октетов выполняется до и после операций \boxplus , \boxminus . Операция циклического сдвига слова на r позиций влево, которая обозначается через RotHi^r , также выполняется с разворотом октетов. Например,

$$\text{RotHi}^5(C8BA94B1_{16}) = 17529639_{16} \sim$$

$\sim 00111001 10010110 01010010 00010111$.

Расписание ключей Belt очень простое. Тактовые ключи $\kappa_1, \kappa_2, \dots, \kappa_{56} \in \{0, 1\}^{32}$ строятся по ключу θ следующим образом:

$$\kappa_1 \parallel \kappa_2 \parallel \dots \parallel \kappa_{56} \leftarrow \underbrace{\theta \parallel \theta \parallel \dots \parallel \theta}_{7 \text{ раз}}.$$

Используется модифицированный экспоненциальный S -блок $H \in S(\{0, 1\}^8)$. Модификация состоит в предварительном сдвиге прообраза по правилу \boxplus . По S -блоку строятся подстановки $G_r \in S(\{0, 1\}^{32})$, $r = 5, 13, 21$:

$$G_r(x_1 \parallel x_2 \parallel x_3 \parallel x_4) = \text{RotHi}^r(H(x_1) \parallel H(x_2) \parallel H(x_3) \parallel H(x_4)).$$

В алгоритмах зашифрования и расшифрования открытый текст X и шифртекст Y разбиваются на слова $a, b, c, d \in \{0, 1\}^{32}$. Алгоритмы имеют следующий вид.

АЛГОРИТМ ЗАШИФР. BELT

Вход: $X \in \{0, 1\}^{128}$, $\theta \in \{0, 1\}^{256}$.

Выход: $Y \in \{0, 1\}^{128}$.

Шаги:

1. $\kappa_1 \parallel \dots \parallel \kappa_{56} \leftarrow \theta \parallel \dots \parallel \theta$.
 2. $a \parallel b \parallel c \parallel d \leftarrow X$.
 3. Для $t=1, 2, \dots, 8$ выполнить:
 - 3.1. $b \leftarrow b \oplus G_5(a \boxplus \kappa_{7t-6})$;
 - 3.2. $c \leftarrow c \oplus G_{21}(d \boxplus \kappa_{7t-5})$;
 - 3.3. $a \leftarrow a \boxminus G_{13}(b \boxplus \kappa_{7t-4})$;
 - 3.4. $e \leftarrow G_{21}(b \boxplus c \boxplus \kappa_{7t-3}) \oplus \langle t \rangle_{32}$;
 - 3.5. $b \leftarrow b \boxplus e$;
 - 3.6. $c \leftarrow c \boxminus e$;
 - 3.7. $d \leftarrow d \boxplus G_{13}(c \boxplus \kappa_{7t-2})$;
 - 3.8. $b \leftarrow b \oplus G_{21}(a \boxplus \kappa_{7t-1})$;
 - 3.9. $c \leftarrow c \oplus G_5(d \boxplus \kappa_{7t})$;
 - 3.10. $a \leftrightarrow b$;
 - 3.11. $c \leftrightarrow d$;
 - 3.12. $b \leftrightarrow c$.
 4. $Y \leftarrow b \parallel d \parallel a \parallel c$.
 5. Возвратить Y .
-

АЛГОРИТМ РАСШИФР. BELT

Вход: $Y \in \{0, 1\}^{128}$, $\theta \in \{0, 1\}^{256}$.

Выход: $X \in \{0, 1\}^{128}$.

Шаги:

1. $\kappa_1 \parallel \dots \parallel \kappa_{56} \leftarrow \theta \parallel \dots \parallel \theta$.
 2. $a \parallel b \parallel c \parallel d \leftarrow Y$.
 3. Для $t=8, 7, \dots, 1$ выполнить:
 - 3.1. $b \leftarrow b \oplus G_5(a \boxplus \kappa_{7t})$;
 - 3.2. $c \leftarrow c \oplus G_{21}(d \boxplus \kappa_{7t-1})$;
 - 3.3. $a \leftarrow a \boxminus G_{13}(b \boxplus \kappa_{7t-2})$;
 - 3.4. $e \leftarrow G_{21}(b \boxplus c \boxplus \kappa_{7t-3}) \oplus \langle t \rangle_{32}$;
 - 3.5. $b \leftarrow b \boxplus e$;
 - 3.6. $c \leftarrow c \boxminus e$;
 - 3.7. $d \leftarrow d \boxplus G_{13}(c \boxplus \kappa_{7t-4})$;
 - 3.8. $b \leftarrow b \oplus G_{21}(a \boxplus \kappa_{7t-5})$;
 - 3.9. $c \leftarrow c \oplus G_5(d \boxplus \kappa_{7t-6})$;
 - 3.10. $a \leftrightarrow b$;
 - 3.11. $c \leftrightarrow d$;
 - 3.12. $a \leftrightarrow d$.
 4. $X \leftarrow c \parallel a \parallel d \parallel b$.
 5. Возвратить X .
-

Обратим внимание, что такты зашифрования и расшифрования структурно близки. Переключение между Belt_θ и Belt_θ^{-1} реализуется изменением порядка следования тактовых ключей, а также перестановками переменных a, b, c, d на шагах 3.12 и 4.

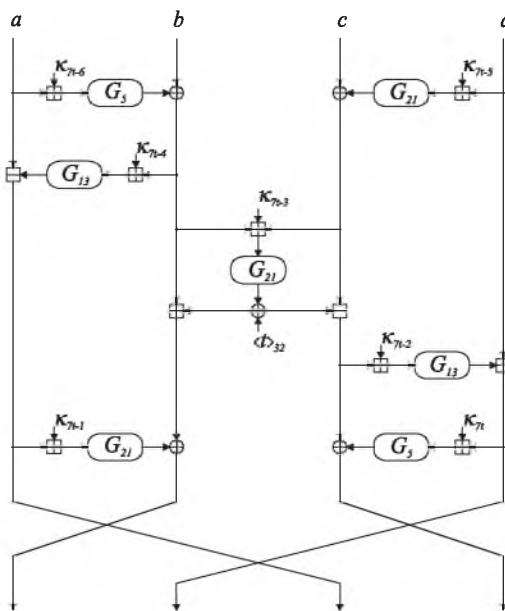


Рис. 12.6. Вычисления на t -м такте зашифрования Belt

Обработка слов a , b , c и d на t -м такте зашифрования изображена на рис. 12.6.

12.11. АТАКА «ГРУБОЙ СИЛОЙ»

Мы достаточно подробно обсудили принципы построения блочных криптосистем, перейдем теперь к атакам на них. Начнем с атаки «грубой силой» – самой простой и в некоторых случаях единственной подходящей. Термин «грубой силой» – это калька с английского brute force. (Некоторые начинающие последователи Виктора так и говорят: *брутфорс*.)

Атака «грубой силой» проводится при известном открытом тексте и направлена на решение задачи C1. Атака является *универсальной* в том смысле, что может быть применена к любой криптосистеме.

Пусть Виктор располагает T блоками открытого текста $X_t \in \{0, 1\}^{n_b}$ и соответствующими блоками шифртекста $Y_t \in \{0, 1\}^{n_b}$. По этому шифрматериалиу Виктор строит систему уравнений

$$Y_t = F_\theta(X_t), \quad t = 1, \dots, T,$$

относительно θ . Как правило, это система имеет единственное решение уже при небольшом T .

Виктор проводит атаку «грубой силой», выбирая последовательно кандидатов $\hat{\theta} \in \Theta$ и проверяя равенства $Y_t \stackrel{?}{=} F_{\hat{\theta}}(X_t)$, $t = 1, \dots, T$. При выполнении всех равенств принимается решение о том, что $\theta = \hat{\theta}$ и поиск прекращается.

Сложность атаки «грубой силой» характеризуется величиной τ – числом использованных кандидатов $\hat{\Theta}$. Пусть искомый ключ θ был выбран Трентом из Θ случайно равновероятно и пусть $\theta_1, \dots, \theta_{|\Theta|}$ – последовательные ключи-кандидаты, проверяемые Виктором. Тогда среднее значение

$$\mathbf{E}\tau = \sum_{t=1}^{|\Theta|-1} t \mathbf{P}\{\theta = \theta_t\} = \frac{(|\Theta|-1)|\Theta|}{2|\Theta|} = \frac{1}{2}(|\Theta|-1).$$

При оценке мы опустили проверку последнего кандидата $\theta_{|\Theta|}$ – его проверять не надо, он заведомо будет правильным.

Реализация атаки. Для проведения атаки «грубой силой» используются специализированные микропроцессорные устройства или распределенные сетевые вычисления. Например, в 1999 г. с использованием специализированного компьютера Deep Crack стоимостью 250 тыс. долл. была проведена атака «грубой силой» на DES ($|\Theta| = 2^{56}$). Deep Crack проверял около 80 млрд ключей-кандидатов в секунду. Поиск ключа занял 22 ч и 15 мин. Для сравнения, в 2010 г. устройство RIVYERA, разработанное компанией SciEngines, проверяло в секунду уже 292 млрд ключей и его стоимость составляла всего 15 тыс. долл. В сентябре 2002 г. методом «грубой силы» был найден ключ крипtosистемы RC5 ($|\Theta| = 2^{64}$). Поиск ключа занял около 4 лет и проводился на 331 252 компьютерах сети Интернет.

В рамках проекта ECRYPT (<http://www.ecrypt.eu.org>), выполняемого под эгидой Евросоюза, разработаны и периодически обновляются оценки стойкости к атаке «грубой силой» в зависимости от длины ключа. В табл. 12.3 представлены оценки 2012 г.

Таблица 12.3

Стойкость к атаке «грубой силой» (оценки ECRYPT)

Длина ключа (в битах)	Уровень защиты
32	Зашита от атак «реального времени» отдельных лиц
64	Краткосрочная защита от атак малой организации (бюджет – 10 тыс. долл.)
72	Краткосрочная защита от атак средней организации (бюджет – 300 тыс. долл.)
80	Краткосрочная защита от атак государственного агентства (бюджет – 300 млн долл.)
112	Среднесрочная защита (на 20 лет) от атак государственного агентства
128	Долгосрочная защита (на 30 лет) от атак государственного агентства
256	Зашита на все обозримое будущее

Простые соотношения. Простым соотношением для F называются равенства

$$g_2 F_{h(\theta)} g_1(X) = F_\theta(X), \quad g_1, g_2 \in S(\{0, 1\}^{n_b}), \quad h \in S(\Theta), \quad h \neq id,$$

которые выполняются для всех $X \in \{0, 1\}^{n_b}$ и $\theta \in \Theta$.

Простое соотношение можно использовать для снижения сложности атаки «грубой силой». Покажем, как это сделать. Пусть известны пары (X, Y_1) , $(g_1(X), Y_2)$ – «открытый текст – шифртекст» преобразования F_θ . Виктор выбирает ключ-кандидат $\hat{\theta}$ и выполняет зашифрование $\hat{Y} = F_{\hat{\theta}}(X)$. Если $\hat{\theta} = \theta$, то $\hat{Y} = Y_1$, а если $h(\hat{\theta}) = \theta$, то

$$Y_2 = F_\theta g_1(X) = F_{h(\hat{\theta})} g_1(X) = g_2^{-1} g_2 F_{h(\hat{\theta})} g_1(X) = g_2^{-1} F_{\hat{\theta}}(X) = g_2^{-1}(\hat{Y})$$

и $\hat{Y} = g_2(Y_2)$. Таким образом, выполнив одно зашифрование, Виктор может проверить сразу два ключа: $\hat{\theta}$ и $h(\hat{\theta})$.

Пример 12.11 (простые соотношения для G). Имеется 255 простых соотношений для криптосистемы G , описанной в примере 12.10. Действительно, для любого $a \in \{0, 1\}^8$ выполняется

$$\begin{aligned} f_{\kappa \oplus a}(x \oplus a) &= f_\kappa(x) \Rightarrow \Sigma_{\kappa \oplus a}(X \oplus (a \parallel a)) \oplus (a \parallel a) = \Sigma_\kappa(X) \Rightarrow \\ &\Rightarrow F_{\theta \oplus (a \parallel a \parallel a)}(X \oplus (a \parallel a)) \oplus (a \parallel a) = F_\theta(X). \end{aligned}$$

Последнее тождество является простым соотношением при $a \neq 0$. Используя все эти соотношения, Виктор за одно зашифрование может проверять не один, а сразу 256 ключей криптосистемы G .

Баланс «время – память». Пусть известна пара «открытый текст X , шифртекст $Y = F_\theta(X)$ », и по ней требуется найти ключ θ . Будем считать, что $N = |\Theta| = 2^{n_b}$ и уравнение $Y = F_\theta(X)$ имеет малое число решений относительно θ (одно из решений совпадает с θ).

Описанная выше атака «грубой силой» проводится за время $O(N)$ на памяти $O(1)$. Возможна модификация атаки. На подготовительном этапе Виктор для всевозможных кандидатов $\hat{\theta}$ вычисляет $\hat{Y} = F_{\hat{\theta}}(X)$ и помещает в ячейку памяти по адресу \hat{Y} значение $\hat{\theta}$ (вообще говоря, ячейки могут содержать несколько значений). На оперативном этапе Виктор получает Y , обращается к ячейке по адресу Y и определяет все $\hat{\theta}$, которые переводят X в Y . Виктор делает это за время $O(1)$ на памяти $O(N)$. Сложность подготовительного этапа при этом не учитывается. Виктору важно провести с минимальными издержками именно оперативный этап атаки, как можно быстрее обработать полученную пару (X, Y) . К сожалению Виктора, память объема $O(N)$ может быть ему недоступна.

В 1980 г. М. Хеллман предложил метод организации атаки «грубой силой», промежуточный между двумя описанными выше. Этот метод, названный *балансом «время – память»*, позволяет найти θ за время $O(N^{2/3})$ на памяти $O(N^{2/3})$.

Суть метода Хеллмана состоит в следующем. Пусть r – некоторая просто вычисляемая и близкая к биективной функция $\{0, 1\}^{n_b} \rightarrow \Theta$ (например, перестановка битов) и пусть

$$h_r: \Theta \rightarrow \Theta, \quad h_r(\theta) = r(F_\theta(X)).$$

Тогда задача определения ключа θ по заданному Y сводится к решению уравнения $h_r(\theta) = r(Y)$, т. е. к обращению функции h_r .

На подготовительном этапе Виктор выбирает ключи-кандидаты $\theta_0 \in \Theta$ и для каждого из них строит траекторию

$$\theta_0, \quad \theta_1 = h_r(\theta_0), \quad \theta_2 = h_r(\theta_1), \dots, \quad \theta_T = h_r(\theta_{T-1}).$$

Виктор сохраняет начало θ_0 и конец θ_T траектории, а остальные элементы отбрасывает. При этом вместо $O(T)$ ячеек памяти требуется использовать только $O(1)$. Сжатие данных является обратимым – отброшенные элементы траектории могут быть восстановлены по θ_0 за время $O(T)$.

На оперативном этапе Виктор получает Y , вычисляет траекторию

$$Y_0 = r(Y), \quad Y_t = h_r(Y_{t-1}), \quad t = 1, \dots, T,$$

и на каждом шаге вычислений проверяет совпадение $Y_t \stackrel{?}{=} \theta_T$. Если $\theta = \theta_{T-\tau-1}$ (траектория ключей-кандидатов покрыла θ), то

$$\begin{aligned} Y_0 &= r(Y) = r(F_{\theta_{T-\tau-1}}(X)) = h_r(\theta_{T-\tau-1}) = \theta_{T-\tau}; \\ Y_1 &= h_r(Y_0) = h_r(\theta_{T-\tau}) = \theta_{T-\tau+1}; \\ &\dots \\ Y_\tau &= \theta_T, \end{aligned}$$

т. е. искомое совпадение будет найдено. Располагая номером итерации τ , на которой совпадение произошло, и началом траектории θ_0 , Виктор определяет искомый ключ $\theta_{T-\tau-1}$.

Хеллман предложил использовать R таблиц, в каждой из которых выбирать S различных значений θ_0 и применять разные функции r при построении h_r . Тогда время подготовительного этапа (игнорируется) – $O(RST)$, память – $O(RS)$, время оперативного этапа – $O(RT)$. Оказывается, что вероятность успеха атаки (т. е. покрытия θ хотя бы одной траекторией ключей-кандидатов) максимальна, если $R = S = T = O(N^{1/3})$.

12.12. РАЗНОСТНАЯ АТАКА

Пусть F – d -тактовая блочно-итерационная криптосистема. *Разностная атака* на F проводится при выбираемом открытом тексте. Целью атаки является определение последнего тактового ключа κ_d . Определив κ_d , можно перейти к $(d - 1)$ -тактовой криптосистеме, определить ее последний тактовый ключ κ_{d-1} и так далее, вплоть до определения всех тактовых ключей и решения таким образом задачи **C2**. Сразу оговоримся, что существует множество модификаций разностной атаки (о некоторых мы поговорим в конце пункта). В частности, κ_d может определяться не полностью, а только частично.

Разностная атака была предложена в 1991 г. Э. Бихамом и А. Шамиром применительно к криптосистеме DES. Как выяснилось впоследствии (эти сведения были первоначально секретными), разработчики DES уже знали о технике разностного криptoанализа и предусмотрели в DES соответствующие защитные механизмы.

Разностные атаки сегодня являются одними из наиболее эффективных. Так, сразу после появления работы Бихама и Шамира с помощью предложенной ими техники были успешно атакованы криптосистемы Lucifer, FEAL, REDOC-II, Khafre и др. В настоящее время при разработке блочных криптосистем стойкость к разностным атакам проверяется, как правило, в первую очередь.

Характеристики и дифференциалы. Для описания разностной атаки нам потребуется несколько построений. Рассмотрим зашифрование пары открытых текстов $X, \tilde{X} \in \{0, 1\}^{nb}$. Пусть $\Delta X = X \oplus \tilde{X}$ – разность между ними. Эта разность порождает последовательность разностей

$$\Delta Y(1) = Y(1) \oplus \tilde{Y}(1), \dots, \Delta Y(d) = Y(d) \oplus \tilde{Y}(d).$$

Здесь $Y(i), \tilde{Y}(i)$ – результаты выполнения i тактов преобразования $F_\theta = \Sigma_{\kappa_d} \dots \Sigma_{\kappa_1}$ над X и \tilde{X} соответственно. В частности, $Y(d) = F_\theta(X)$ и $\tilde{Y}(d) = F_\theta(\tilde{X})$.

Последовательность $(\alpha, \beta(1), \dots, \beta(r))$ возможных значений разностей $(\Delta X, \Delta Y(1), \dots, \Delta Y(r))$ называется *r-тактовой характеристикой*, а пара $(\alpha, \beta(r))$ значений $(\Delta X, \Delta Y(r))$ – *r-тактовым дифференциалом*, $1 \leq r \leq d$. Характеристика $(0, 0, \dots, 0)$ и дифференциал $(0, 0)$ являются тривиальными и далее не рассматриваются.

В предположении, что X, \tilde{X} – случайные независимые слова с равномерным на $\{0, 1\}^{nb}$ распределением, а ключ θ выбран из Θ случайно равновероятно, введем *вероятность характеристики*

$$\mathbf{P}\{\Delta Y(1) = \beta(1), \dots, \Delta Y(r) = \beta(r) \mid \Delta X = \alpha\}$$

и *вероятность дифференциала* $\mathbf{P}\{\Delta Y(r) = \beta(r) \mid \Delta X = \alpha\}$.

Подготовительный этап. На подготовительном этапе Виктор находит $(d - 1)$ -тактовый дифференциал $(\alpha, \beta(d - 1))$ с максимальной вероятностью p . Сразу скажем, что для успеха атаки требуется выполнение условия $p \gg 2^{-n_b}$.

Поиск высоковероятного дифференциала — непростая задача, успешное решение которой зависит от квалификации криптоаналитика. Виктор должен тщательно изучить криптосистему, проанализировать таблицы разностей S -блоков, исследовать характеристики перемешивания преобразований перестановки и т. д.

Обычно вместо высоковероятного дифференциала $(\alpha, \beta(d - 1))$ Виктор ищет высоковероятную характеристику $(\alpha, \beta(1), \dots, \beta(d - 1))$ и использует оценки:

$$\begin{aligned} p &\geq \mathbf{P}\{\Delta Y(1) = \beta(1), \dots, \Delta Y(r) = \beta(d - 1) \mid \Delta X = \alpha\} \approx \\ &\approx \mathbf{P}\{\Delta Y(1) = \beta(1) \mid \Delta X = \alpha\} \times \\ &\times \prod_{i=2}^{d-1} \mathbf{P}\{\Delta Y(i) = \beta(i) \mid \Delta Y(i-1) = \beta(i-1)\}. \end{aligned}$$

При этом Виктор идеализирует расписание ключей F , считая, что при случайному равновероятном выборе θ последовательность $(\kappa_1, \dots, \kappa_{d-1})$ равномерно распределена на K^{d-1} . Как правило, такая идеализация не сказывается на адекватности окончательных выводов.

Оперативный этап. Выбирая ключ-кандидат $\hat{\kappa}_d \in K$, Виктор может построить прогноз

$$Z(Y, \tilde{Y}; \hat{\kappa}_d) = \Sigma_{\hat{\kappa}_d}^{-1}(Y) \oplus \Sigma_{\hat{\kappa}_d}^{-1}(\tilde{Y})$$

предпоследней разности $\Delta Y(d - 1)$.

На оперативном этапе разностной атаки Виктор случайным образом выбирает открытые тексты $X_t \in \{0, 1\}^{n_b}$ и определяет шифртексты $Y_t = F_\theta(X_t)$, $\tilde{Y}_t = F_\theta(X_t \oplus \alpha)$, $t = 1, \dots, T$. В качестве оценки искомого ключа κ_d Виктор выбирает значение $\hat{\kappa}_d$, доставляющее максимум сумме

$$W(\hat{\kappa}_d) = \sum_{t=1}^T \mathbf{I}\left\{Z(Y_t, \tilde{Y}_t; \hat{\kappa}_d) = \beta(d - 1)\right\}.$$

Проведем грубый анализ сложности атаки (более тонкий анализ предполагает учет свойств конкретной криптосистемы). При $\hat{\kappa}_d = \kappa_d$ среднее значение

$$\mathbf{E}W(\hat{\kappa}_d) = \sum_{t=1}^T \mathbf{P}\left\{Z(Y_t, \tilde{Y}_t; \hat{\kappa}_d) = \beta(d - 1)\right\} = Tp.$$

Если же $\hat{\kappa}_d \neq \kappa_d$, то можно предположить (это еще одно допущение, которое в определенных случаях оказывается адекватным), что слова $Z(Y_t, \tilde{Y}_t; \hat{\kappa}_d)$ равномерно распределены на $\{0, 1\}^{n_b} \setminus \{0\}$. При этом

$$\mathbf{E}W(\hat{\kappa}_d) = T/(2^{n_b} - 1).$$

Потребуем, чтобы среднее значение (целочисленной) целевой функции W на верном ключе было больше, чем на остальных:

$$\mathbf{E}W(\kappa_d) \geq \mathbf{E}W(\hat{\kappa}_d) + 1, \quad \hat{\kappa}_d \neq \kappa_d.$$

Отсюда

$$T \geq \frac{1}{p - 1/(2^{n_b} - 1)}.$$

Модификации. Перечислим основные модификации разностной атаки.

Обобщенные разности. Разность определяется не с помощью операции \oplus , а с помощью другой алгебраической операции. Например, $\Delta X = X \boxminus \tilde{X}$.

Усеченные разности. Контролируется не все, а только выборочные символы разностей $\Delta Y(i)$.

Запрещенные дифференциалы. Вместо высоковероятных дифференциалов $(\alpha, \beta(r))$ используются запрещенные дифференциалы, т. е. дифференциалы с нулевой вероятностью. Если $(\alpha, \beta(d-1))$ – запрещенный дифференциал, то для определения κ_d нужно не максимизировать $\sum Z(Y_t, \tilde{Y}_t; \hat{\kappa}_d)$, а браковать те $\hat{\kappa}_d$, для которых $Z(Y_t, \tilde{Y}_t; \hat{\kappa}_d) = \beta(d-1)$.

Атака на несколько тактовых ключей. Определяется не один, а несколько последних тактовых ключей. Для определения $(\kappa_{r+1}, \dots, \kappa_d)$ Виктор находит высоковероятный r -тактовый дифференциал $(\alpha, \beta(r))$, просматривает ключи-кандидаты $(\hat{\kappa}_{r+1}, \dots, \hat{\kappa}_d)$ и составляет по ним прогнозы

$$Z(Y, \tilde{Y}; \hat{\kappa}_{r+1}, \dots, \hat{\kappa}_d) = \Sigma_{\hat{\kappa}_{r+1}}^{-1} \dots \Sigma_{\hat{\kappa}_d}^{-1}(Y) \oplus \Sigma_{\hat{\kappa}_{r+1}}^{-1} \dots \Sigma_{\hat{\kappa}_d}^{-1}(\tilde{Y})$$

разности $\Delta Y(r)$. Как и ранее, в качестве искомой оценки последних тактовых ключей выбирается тот набор ключей-кандидатов, на котором прогноз разности чаще всего совпадает с $\beta(r)$.

Частичная информация о ключах. В некоторых случаях различные наборы ключей-кандидатов могут давать одинаковые прогнозы $Z(Y, \tilde{Y}; \hat{\kappa}_{r+1}, \dots, \hat{\kappa}_d)$. В таких случаях Виктор может определить только частичную информацию $\mathcal{I} = \mathcal{I}(\kappa_{r+1}, \dots, \kappa_d)$ о последних тактовых ключах. Эту информацию часто называют *эффективными битами ключа*. Эффективные биты выбираются так, чтобы при любой их оценке $\hat{\mathcal{I}} = \mathcal{I}(\hat{\kappa}_{r+1}, \dots, \hat{\kappa}_d)$ для пары шифртекстов Y, \tilde{Y} можно было построить прогноз разности $\Delta Y(r)$. Частичная информация о ключах определяется по той же схеме, что и полная.

После определения \mathcal{I} недостающие данные о последних тактовых ключах Виктор может получить с помощью атаки «грубой силой» или с помощью той же разностной атаки, но уже с другим дифференциалом.

Пример 12.12 (разностная атака на DES). В атаке Бихама и Шамира на DES используются два 13-тактовых дифференциала, каждый с вероятностью $\approx 2^{-47.2}$. С помощью этих дифференциалов можно в совокупности определить 52 бита двух последних тактовых ключей. При этом требуется использовать $T = 2^{47}$ пар «открытый текст – шифртекст». Вероятность успеха атаки $\approx 0,58$.

Пример 12.13 (разностная атака на G). Проведем разностную атаку на модельную крипtosистему G , описанную в примере 12.10. Для простоты будем опускать при зашифровании заключительную перестановку половинок, которая выполняется в крипtosистемах Фейстеля.

Рассмотрим тактовую функцию f_k . Пусть на вход f_k подана пара случайных октетов x, \tilde{x} с разностью $\gamma = 10000000$. После сложения с k разность не изменится, и на вход S_1 попадут тетрады с разностью 1000, а на вход S_2 – одинаковые тетрады. В соответствии с расчетами, проведенными в примере 12.9, на выходе S_1 с вероятностью $1/4$ будет получена разность 0001. Выходы S_2 будут одинаковыми. Вычисление значений f_k заканчивается объединением выходов S -блоков в октеты и циклическим сдвигом этих октетов на 3 позиции влево. Окончательная разность будет равняться γ . Таким образом, $P\{f_k(x) \oplus f_k(\tilde{x}) = \gamma \mid x \oplus \tilde{x} = \gamma\} = 1/4$ или, другими словами, f_k сохраняет разность γ с вероятностью $1/4$.

Рассуждения можно продолжить и построить следующую 7-тактовую разностную характеристику для G :

$$\begin{aligned}\alpha &= \gamma \parallel 0 \\ \beta(1) &= 0 \parallel \gamma \quad (\text{с вероятностью } 1) \\ \beta(2) &= \gamma \parallel \gamma \quad (\text{с вероятностью } 1/4) \\ \beta(3) &= \gamma \parallel 0 \quad (\text{с вероятностью } 1/4) \\ \beta(4) &= 0 \parallel \gamma \quad (\text{с вероятностью } 1) \\ \beta(5) &= \gamma \parallel \gamma \quad (\text{с вероятностью } 1/4) \\ \beta(6) &= \gamma \parallel 0 \quad (\text{с вероятностью } 1/4) \\ \beta(7) &= 0 \parallel \gamma \quad (\text{с вероятностью } 1)\end{aligned}$$

Вероятность характеристики и соответствующего дифференциала ($\alpha, \beta(7)$) оценим как произведение однотактовых характеристик: $p \geq 2^{-8}$. Поскольку $n_b = 16$ и $p \gg 2^{-16}$, дифференциал можно использовать для определения информации $\mathcal{I} = \mathcal{I}(k_8)$ о последнем тактовом ключе. Выясним, какую информацию \mathcal{I} мы можем восстановить, и обсудим детали восстановления.

Разобъем слова $Y(i)$ на тетрады:

$$Y(i) = Y_1(i) \parallel Y_2(i) \parallel Y_3(i) \parallel Y_4(i), \quad Y_k(i) \in \{0, 1\}^4.$$

Аналогичные разбиения введем для $\tilde{Y}(i)$. Пусть $\kappa_8 = a \parallel b$, $a, b \in \{0, 1\}^4$. Пару (X, \tilde{X}) открытых текстов назовем верной, если она удовлетворяет дифференциальному, т. е. $\Delta X = \alpha$ и $\Delta Y(7) = \beta(7)$. Для верной пары

$$\tilde{Y}_1(7) = Y_1(7), \quad \tilde{Y}_2(7) = Y_2(7), \quad \tilde{Y}_3(7) = Y_3(7) \oplus 1000, \quad \tilde{Y}_4(7) = Y_4(7),$$

и поэтому

$$\Delta Y(8) = \gamma \parallel *0000***. \quad (12.3)$$

Здесь знак * соответствует символам, которые могут принимать любые значения, только не все нулевые одновременно.

Пару (X, \tilde{X}) , для которой выполняется соотношение (12.3), назовем подходящей. Существуют подходящие пары, которые не являются верными. Впрочем вероятность их появления $\approx 2^{-12} \ll p$. Поэтому будем упрощать рассуждения и предполагать далее, что всякая пара, удовлетворяющая (12.3), является верной.

По верной паре построим слово $Z \in \{0, 1\}^4$. Это слово составим из символов $\Delta Y(8)$ с номерами 14, 15, 16 и 9, т. е. из символов, помеченных знаком *. Тетрады a и b последнего тактового ключа участвуют в следующих разностных соотношениях:

$$\begin{aligned} S_1(Y_3(7) \oplus a) \oplus S_1(Y_3(7) \oplus 1000 \oplus a) &= Z, \\ S_2(Y_4(7) \oplus b) \oplus S_2(Y_4(7) \oplus b) &= 0. \end{aligned}$$

Второе соотношение является тождеством и не позволяет определить символы b , а вот с помощью первого можно получить информацию об a .

Перепишем первое соотношение:

$$S_1(Y_1(8) \oplus a) \oplus S_1(Y_1(8) \oplus 1000 \oplus a) = Z. \quad (12.4)$$

Нам известны все используемые здесь слова, за исключением a . S -блок S_1 устроен так, что при изменении 1-го символа a равенство не нарушится. Поэтому информация J о последнем тактовом ключе, которую можно получить, касается 2-го, 3-го и 4-го символов a . Для определения J следует ожидать появления верных пар, проверять для них выполнение (12.4) при всевозможных значениях J и принимать решение в пользу того значения, на котором (12.4) выполнилось наибольшее количество раз.

Пусть y – случайное слово с равномерным распределением на $\{0, 1\}^4$. Прямые расчеты показывают, что слово $z = S_1(y) \oplus S_1(y \oplus 1000)$ имеет следующее распределение вероятностей:

$$\mathbf{P}\{z = 0001\} = 1/4, \quad \mathbf{P}\{z = 1001\} = 1/4, \quad \mathbf{P}\{z = 0101\} = 1/2$$

(по первому равенству и рассчитывалась вероятность разностной характеристики). Поэтому случайное значение J приведет к выполнению (12.4) с вероятностью

$$(1/4)^2 + (1/4)^2 + (1/2)^2 = 3/8.$$

С другой стороны, при $\hat{J} = J$ равенство будет выполняться с вероятностью 1. Различие вероятностей и является основанием результивности разностной атаки.

В ходе вычислительных экспериментов 3 бита последнего тактового ключа гарантированно восстанавливались при анализе ≈ 8 верных пар. При этом требовалось генерировать $T \approx 2000$ пар открытого текста.

12.13. ЛИНЕЙНАЯ АТАКА

Линейная атака проводится при известном открытом тексте. В самом простом случае, который мы и рассмотрим, объектом атаки является d -тактовая блочно-итерационная криптосистема F с множеством тактовых ключей $K = \{0, 1\}^{nb}$ и тактовыми подстановками следующего вида:

$$\Sigma_{\kappa}: X \mapsto s(X \oplus \kappa), \quad s \in S(\mathbb{F}_2^n).$$

Целью атаки является определение одного бита информации о $(\kappa_1, \dots, \kappa_d)$. Другие биты могут быть определены с помощью той же линейной атаки, но с другими настройками.

Линейная атака была предложена в 1993 г. М. Мацуи применительно к DES. Мацуи довел атаку до практической стадии – в 1994 г. в ходе трудоемкого вычислительного эксперимента была впервые решена задача С1 для DES.

Линейные аппроксимации. Прежде чем описывать линейную атаку, проведем подготовительные построения. Будем предполагать, что ключ θ выбран из Θ случайно равновероятно. Как и в разностной атаке, будем идеализировать расписание ключей F , считая, что последовательность $(\kappa_1, \dots, \kappa_d)$ равномерно распределена на K^d .

Зашифрование $Y = F_\theta(X)$ состоит в выполнении тактовых преобразований

$$Y(1) = s(X \oplus \kappa_1), \quad Y(2) = s(Y(1) \oplus \kappa_2), \quad \dots, \quad Y = s(Y(d-1) \oplus \kappa_d).$$

Для подстановки s , которая применяется на i -м такте, используем линейную аппроксимацию $(\beta(i-1), \beta(i))$, пусть q_i – ее вероятность. Обозначим

$$b(\kappa_1, \dots, \kappa_d) = \beta(0) \cdot \kappa_1 \oplus \dots \oplus \beta(d-1) \cdot \kappa_d.$$

Объединение аппроксимаций $(\beta(i-1), \beta(i))$, $i = 1, \dots, d$, называется d -тактовой аппроксимацией и обозначается через $(\beta(0), \beta(1), \dots, \beta(d))$. Вероятность d -тактовой аппроксимации определяется как

$$p = \mathbf{P} \{ \beta(0) \cdot X \oplus \beta(d) \cdot F_\theta(X) = b(\kappa_1, \dots, \kappa_d) \}.$$

Тривиальные аппроксимации $(0, 0, \dots, 0)$ с вероятностью 1 договоримся далее не рассматривать.

Введем случайные величины

$$\begin{aligned}\xi_1 &= \beta(0) \cdot (X \oplus \kappa_1) \oplus \beta(1) \cdot Y(1); \\ \xi_2 &= \beta(1) \cdot (Y(1) \oplus \kappa_2) \oplus \beta(2) \cdot Y(2); \\ &\dots \\ \xi_d &= \beta(d-1) \cdot (Y(d-1) \oplus \kappa_d) \oplus \beta(d) \cdot Y.\end{aligned}$$

Эти величины независимы, для них $\mathbf{P} \{\xi_i = 0\} = q_i$ и $\mathbf{P} \{\xi_1 \oplus \dots \oplus \xi_d = 0\} = p$. Определить p по q_i можно с помощью следующей леммы.

Лемма 12.2 (лемма о набегании знаков). Пусть ξ_1, \dots, ξ_d – независимые бернульиевские случайные величины и $\mathbf{P} \{\xi_i = 0\} = q_i$. Тогда

$$\mathbf{P} \{\xi_1 \oplus \xi_2 \dots \oplus \xi_d = 0\} = \frac{1}{2} + 2^{d-1} \prod_{i=1}^d \left(q_i - \frac{1}{2} \right).$$

Доказательство. Введем преобладания

$$\varepsilon_i = \mathbf{P} \{\xi_1 \oplus \dots \oplus \xi_i = 0\} - \frac{1}{2}, \quad i = 1, \dots, d.$$

Формула леммы справедлива при $d = 1$. Для $d \geq 2$ имеем

$$\begin{aligned}\varepsilon_d &= \sum_{c \in \{0, 1\}} \mathbf{P} \{\xi_1 \oplus \dots \oplus \xi_{d-1} = c, \xi_d = c\} - \frac{1}{2} = \\ &= \left(\varepsilon_{d-1} + \frac{1}{2} \right) q_d + \left(\frac{1}{2} - \varepsilon_{d-1} \right) (1 - q_d) - \frac{1}{2} == 2\varepsilon_{d-1} \left(q_d - \frac{1}{2} \right),\end{aligned}$$

откуда и следует требуемый результат. \square

Подготовительный этап. На подготовительном этапе линейной атаки Виктор находит d -тактовую линейную аппроксимацию $(\beta(0), \dots, \beta(d))$ с максимально отличной от $1/2$ вероятностью p . Если $(p - 1/2)^2 \gg 2^{-n_b}$, то с помощью этой аппроксимации Виктор сможет определить бит $b(\kappa_1, \dots, \kappa_d)$.

Как и в разностной атаке, подготовительный этап является творческим. Виктор должен изучить структуру криптосистемы, проанализировать нелинейность S -блоков, исследовать характеристики перемешивания и т. д.

Оперативный этап. На оперативном этапе Виктор получает открытые тексты X_t и перехватывает соответствующие шифртексты $Y_t = F_\theta(X_t)$, $t = 1, 2, \dots, T$. Сделаем еще одно допущение: пусть открытые тексты X_t являются реализациями независимых случайных величин с равномерным распределением на $\{0, 1\}^{n_b}$. Это допущение упрощает рассуждения, но не скрывается на адекватности окончательных выводов в практических важных случаях.

По полученному шифрматериалу Виктор определяет сумму

$$W = \sum_{t=1}^T \mathbf{1}\{\beta(0) \cdot X_t = \beta(d) \cdot Y_t\}.$$

Если $p > 1/2$, то оценка бита $\hat{b} = b(\kappa_1, \dots, \kappa_d)$ строится следующим образом:

$$\hat{b} = \begin{cases} 0, & W \geq \frac{T}{2}, \\ 1 & \text{в противном случае.} \end{cases}$$

Если $p < 1/2$, то эта оценка инвертируется.

Проведем анализ требуемого для атаки числа пар T . Пусть, не нарушая общности, $p > 1/2$ и $b = 0$. Величина W представляет собой сумму независимых бернуlliевских случайных величин $\delta_t = \mathbf{1}\{\beta(0) \cdot X_t = \beta(d) \cdot Y_t\}$ со средним $E\delta_t = p$. Поэтому вероятность ошибки при оценивании b можно оценить по теореме Муавра – Лапласа:

$$\begin{aligned} \mathbf{P}\left\{\hat{b} \neq b\right\} &= \mathbf{P}\left\{W < \frac{T}{2}\right\} = \\ &= \mathbf{P}\left\{\frac{W - pT}{\sqrt{Tp(1-p)}} < \frac{(1/2 - p)T}{\sqrt{Tp(1-p)}}\right\} \approx \\ &\approx \Phi\left(-2\sqrt{T}(p - 1/2)\right). \end{aligned}$$

Здесь $\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-x^2/2} dx$ – функция распределения стандартного нормального закона.

Для того чтобы вероятность ошибки была мала, должно использоваться порядка $|p - 1/2|^{-2}$ пар «открытый текст – шифртекст».

Модификации. Методы линейного криптоанализа можно применить не только к криптосистемам с тактовыми подстановками $X \mapsto s(X \oplus \kappa)$, но и к другим криптосистемам, например Фейстеля. При этом используются различные модификации базовой линейной атаки, описанной выше. Перечислим основные из них.

Несколько аппроксимаций. Используется не одна, а несколько линейных аппроксимаций. Разные аппроксимации позволяют определить разные биты ключа, или разные аппроксимации повышают качество определения одного и того же бита.

Атака на последние тактовые ключи. Определяется информация $J = J(\kappa_{r+1}, \dots, \kappa_d)$ о последних тактовых ключах. Для этого используется

r -тактовая линейная аппроксимация $(\beta(0), \dots, \beta(r))$. Предполагается, что по всякой оценке \hat{J} можно построить прогноз бита $\beta(r) \cdot Y(r)$. В ходе атаки выбирается та оценка, для которой эти прогнозы максимально часто совпадают с битами $\beta(0) \cdot X$ (случай А) или еще чаще отличаются от них (случай Б). Одновременно строится оценка для бита $b(\kappa_1, \dots, \kappa_r)$. Это либо 0 в случае А, либо 1 в случае Б.

Аппроксимации с нулевым преобладанием. Вместо аппроксимаций, вероятности которых существенно отличаются от $1/2$, используются аппроксимации с вероятностью $1/2$. Среди оценок \hat{J} выбираются те, для которых корреляции между $\beta(0) \cdot X$ и прогнозом $\beta(r) \cdot Y(r)$ не проявляются.

Пример 12.14 (линейная атака на DES). В своей второй линейной атаке на DES, проведенной в 1994 г., М. Мацуи использовал две 14-тактовые линейные аппроксимации с вероятностью $\frac{1}{2} + 1,19 \cdot 2^{-22}$ каждая. На оперативном этапе было обработано $T = 2^{43}$ пар «открытый текст – шифртекст». По полученному шифрматериалу были определены 26 битов ключа θ . Оставшиеся 30 битов ключа были найдены «грубой силой».

Пример 12.15 (линейная атака на G). Проведем линейную атаку на криптосистему G. Как и в предыдущем примере для G, будем игнорировать заключительную перестановку половинок при зашифровании.

Тактовую функцию f_κ можно представить в виде $f_\kappa(x) = s(x \oplus \kappa)$, где

$$s(x) = (S_1(x_1) \parallel S_2(x_2)) \lll 3, \quad x = x_1 \parallel x_2, \quad x_i \in \{0, 1\}^4.$$

Пусть $\gamma = 11100001$. С учетом вычислений, проведенных в примере 12.7, вероятность $P\{\gamma \cdot x = \gamma \cdot s(x)\}$ есть

$$P\{1110 \cdot x_1 \oplus 0001 \cdot x_2 = 0011 \cdot S_1(x_1) \oplus 1100 \cdot S_2(x_2)\} = \frac{11}{16}.$$

Перейдем к тактовым подстановкам. Рассмотрим следующие соотношения, связывающие прообраз $X = X_1 \parallel X_2$, образ $Y_1 \parallel Y_2$ и тактовый ключ κ :

$$\begin{aligned} \gamma \cdot (X_2 \oplus Y_1) &= 0; \\ \gamma \cdot (X_1 \oplus Y_1 \oplus Y_2) &= \gamma \cdot \kappa; \\ \gamma \cdot (X_1 \oplus X_2 \oplus Y_2) &= \gamma \cdot \kappa. \end{aligned}$$

Поскольку $Y_1 = X_2$ и $Y_2 = X_1 \oplus s(X_2 \oplus \kappa)$, для случайного X с равномерным распределением на $\{0, 1\}^{16}$ первое равенство выполняется с вероятностью 1, а второе и третье – с вероятностью $\frac{11}{16}$.

Используя данные факты, можно построить следующую 7-тактовую линейную аппроксимацию для G :

$$\begin{aligned}\beta(0) &= \mathbf{0} \parallel \gamma \\ \beta(1) &= \gamma \parallel \mathbf{0} \quad (\text{с вероятностью 1}) \\ \beta(2) &= \gamma \parallel \gamma \quad (\text{с вероятностью } 11/16) \\ \beta(3) &= \mathbf{0} \parallel \gamma \quad (\text{с вероятностью } 11/16) \\ \beta(4) &= \gamma \parallel \mathbf{0} \quad (\text{с вероятностью 1}) \\ \beta(5) &= \gamma \parallel \gamma \quad (\text{с вероятностью } 11/16) \\ \beta(6) &= \mathbf{0} \parallel \gamma \quad (\text{с вероятностью } 11/16) \\ \beta(7) &= \gamma \parallel \mathbf{0} \quad (\text{с вероятностью 1})\end{aligned}$$

Вероятность аппроксимации:

$$q = \frac{1}{2} + 2^3 \left(\frac{3}{16}\right)^4 \approx \frac{1}{2} + 2^{-6.6}.$$

Аппроксимацию можно использовать для определения последнего тактового ключа κ_8 (частично) и бита $\gamma \cdot (\kappa_2 \oplus \kappa_3 \oplus \kappa_5 \oplus \kappa_6)$. В вычислительных экспериментах удавалось определять 5 битов информации о κ_8 .

12.14. РЕЖИМЫ ШИФРОВАНИЯ

В протоколе, описанном в начале главы, Алиса и Боб выполняют зашифрование и расшифрование следующим образом:

$$Y_t = F_\theta(X_t), \quad X_t = F_\theta^{-1}(Y_t), \quad t = 1, 2, \dots, T.$$

При этом говорят о шифровании в режиме *простой замены*. Этот режим принято обозначать аббревиатурой ECB (Electronic CodeBook).

В режиме ECB каждый блок открытого текста обрабатывается отдельно от остальных. Поэтому возникают следующие угрозы:

Перестановка блоков. Виктор располагает форматами банковских документов и переставляет местами блоки с младшими и старшими цифрами суммы платежа.

Анализ повторов блоков. Виктор располагает форматами банковских документов и располагает информацией о повторе их блоков. Анализ повторов блоков перехваченного шифртекста позволяет Виктору установить тип документа.

Для защиты от этих угроз используются другие режимы шифрования. В режимах *цеплениия блоков* (CBC, Cipher Block Chaining) и *гаммирования с обратной связью* (CFB, Cipher FeedBack) результат зашифрования блока X_t зависит от всех предыдущих блоков открытого текста X_1, \dots, X_{t-1} .

Правила шифрования в этих режимах соответственно имеют вид:

$$\begin{aligned} \text{CBC: } Y_t &= F_\theta(X_t \oplus Y_{t-1}), & X_t &= Y_{t-1} \oplus F_\theta^{-1}(Y_t); \\ \text{CFB: } Y_t &= X_t \oplus F_\theta(Y_{t-1}), & X_t &= Y_t \oplus F_\theta(Y_{t-1}). \end{aligned}$$

Здесь Y_0 – некоторое наперед заданное слово, называемое *синхропосылкой*.

Синхропосылка обеспечивает уникальность результатов криптографического преобразования на одном и том же ключе. Синхропосылка является несекретным объектом и может передаваться вместе с зашифрованными данными.

В режимах *обратной связи по выходу* (OFB, Output FeedBack) и счетчика (CTR, Counter) зависимость от блоков X_1, \dots, X_{t-1} отсутствует. В данных режимах вырабатывается гамма – последовательность векторов $\Gamma_1, \dots, \Gamma_T \in \{0, 1\}^{n_b}$, которая используется как для зашифрования, так и для расшифрования:

$$Y_t = X_t \oplus \Gamma_t, \quad X_t = Y_t \oplus \Gamma_t, \quad t = 1, \dots, T.$$

Гамма вырабатывается по правилам:

$$\begin{aligned} \text{OFB: } \Gamma_t &= F_\theta(\Gamma_{t-1}); \\ \text{CTR: } \Gamma_t &= F_\theta(S_t), \quad S_t = \varphi(S_{t-1}). \end{aligned}$$

Здесь Γ_0, S_0 – синхропосылки, $\varphi: \{0, 1\}^{n_b} \rightarrow \{0, 1\}^{n_b}$ – *функция инкремента*. Функция инкремента выбирается так, чтобы обеспечить большой период последовательности S_t . Часто инкремент состоит в добавлении числа 1 (представленного двоичным словом) по правилу \boxplus .

Сравнительные характеристики режимов приведены в табл. 12.4.

Таблица 12.4

Режимы шифрования

Свойства	Режимы				
	ECB	CBC	CFB	OFB	CTR
Зависимость от X_1, \dots, X_{t-1}	–	+	+	–	–
Использование F_θ^{-1}	+	+	–	–	–
Восстановление после ошибки в шифртексте ¹	+	+	+	+	+
Восстановление после ошибки в синхропосылке ²	не исп.	+	+	–	–
Распараллеливание ³	+	–	–	–	+

¹ Даже если один из блоков Y_t изменен при передаче, при расшифровании, начиная с некоторого $\tau > t$ будут получены корректные блоки X_τ .

² Даже если синхропосылка изменена при передаче, при расшифровании начиная с некоторого τ будут получены корректные блоки X_τ .

³ Шифрование различных блоков может выполняться одновременно на нескольких процессорах.

12.15. ИМИТОЗАЩИТА

Блочные криптосистемы могут использоваться не только для обеспечения конфиденциальности, но и для контроля целостности сообщений. Для этого по блочной криптосистеме строится система имитозащиты.

Определение 12.3. Системой имитозащиты называется семейство

$$I = \{I_\theta : \theta \in \Theta\}$$

ключезависимых функций $I_\theta : \{0, 1\}^* \rightarrow \{0, 1\}^{n_m}$. Значение $I_\theta(X)$ называется имитовставкой X (на ключе θ).

В англоязычной литературе системы имитозащиты называются также *MAC-системами* (от Message Authentication Codes). Отсюда индекс m в размерности имитовставки.

Для контроля целостности сообщения X Алиса вместе с шифртекстом Y отправляет Бобу имитовставку $Z = I_\theta(X)$. Боб получает шифртекст Y' (который может отличаться от Y), находит открытый текст X' , вычисляет имитовставку $Z' = I_\theta(X')$ и сравнивает ее с Z . Если имитовставки различаются, то Боб принимает решение о том, что $Y' \neq Y$, т. е. шифртекст был изменен в канале связи. Если имитовставки совпадают, то Боб принимает X' .

Детали могут отличаться. Например, имитовставка может вычисляться не от открытого текста X , а от шифртекста Y . При этом Боб сначала проверяет имитовставку, а только затем, при успешной проверке, выполняет расшифрование Y .

При атаках на систему имитозащиты I Виктор получает полную или частичную информацию о сообщениях X (или даже выбирает эти сообщения) и перехватывает соответствующие имитовставки $Z = I_\theta(X)$. Виктору требуется решить одну из следующих задач:

M1: определить ключ θ ;

M2: найти имитовставку заданного сообщения, отличного от предыдущих;

M3: найти имитовставку произвольного сообщения, отличного от предыдущих.

Ясно, что решение первой задачи приводит к решению второй, а решение второй – к решению третьей. Таким образом, третья задача является самой простой и на ней концентрируется основное внимание при оценке стойкости систем имитозащиты.

Если ключ имитозащиты θ используется также для шифрования $X \mapsto Y$, то у Виктора появляется дополнительный шифрматериал Y и, как следствие, дополнительный потенциал при решении **M1**, **M2**, **M3**. Известный криптографический принцип – *ключ должен использоваться по одному назначению* – запрещает совмещение ключей шифрования и имитозащиты.

Тем не менее такое совмещение может быть разрешено, если оно тщательно проанализировано и слабости не обнаружены. Например, в СТБ 34.101.31 определены алгоритмы, которые на одном и том же ключе θ выполняют одновременно и шифрование, и имитозащиту данных.

Рассмотрим два распространенных подхода к построению систем имитозащиты.

Схема CBC-MAC. Для имитозащиты сообщения $X = X_1 \parallel X_2 \parallel \dots \parallel X_T$, $X_t \in \{0, 1\}^{n_b}$, выполняются вычисления, аналогичные шифрованию в режиме CBC с нулевой синхропосылкой: $Y_t = F_\theta(X_t \oplus Y_{t-1})$, $t = 1, 2, \dots, T$, $Y_0 = 0^{n_b}$. Имитовставкой объявляется слово Y_T , если $n_m = n_b$, или выборочные символы этого слова, если $n_m < n_b$. Построенные таким образом системы имитозащиты принято обозначать аббревиатурой CBC-MAC.

Алгоритм выработки имитовставки, определенный в ГОСТ 28147, соответствует схеме CBC-MAC. В этом алгоритме $T \geq 2$, $n_b = 64$, $n_m \leq 32$, имитовставкой объявляются последние символы Y_n .

Существует несколько модификаций схемы CBC-MAC. Одна из них, известная как OMAC, была предложена Т. Иватой и К. Курасовой в 2003 г. Эта схема использована в следующем алгоритме имитозащиты, определенном в СТБ 34.101.31.

АЛГОРИТМ ВЫРАБОТКА ИМИТОВСТАВКИ (СТБ 34.101.31)

Вход: $X \in \{0, 1\}^*$ – сообщение, $\theta \in \{0, 1\}^{256}$ – ключ.

Выход: $Z \in \{0, 1\}^{64}$ – имитовставка.

Шаги:

1. Непустое сообщение X представить в виде $X_1 \parallel \dots \parallel X_{T-1} \parallel X_T$, где $|X_1| = \dots = |X_{T-1}| = 128$, $0 < |X_T| \leq 128$. Для пустого X считать, что $T = 1$ и $|X_1| = 0$.
 2. Установить $s \leftarrow 0^{128}$, $r \leftarrow \text{Belt}_\theta(s)$.
 3. Для $t = 1, 2, \dots, T - 1$ выполнить: $s \leftarrow \text{Belt}_\theta(s \oplus X_i)$.
 4. Если $|X_T| = 128$, то $s \leftarrow s \oplus X_T \oplus \varphi_1(r)$, иначе $s \leftarrow s \oplus \text{Pad}_{128}(X_T) \oplus \varphi_2(r)$.
 5. Записать в Z первые 64 символа слова $\text{Belt}_\theta(s)$.
 6. Возвратить Z .
-

В этом алгоритме Pad_{128} – описанное в начале главы расширение неполного блока до полного (с выбором $\alpha = 1$ и $\beta = 0$):

$$\varphi_1(x_1 \parallel x_2 \parallel x_3 \parallel x_4) = x_2 \parallel x_3 \parallel x_4 \parallel (x_1 \oplus x_2);$$

$$\varphi_2(x_1 \parallel x_2 \parallel x_3 \parallel x_4) = (x_1 \oplus x_4) \parallel x_1 \parallel x_2 \parallel x_3, \quad x_i \in \{0, 1\}^{32}.$$

Отметим, что φ_i выбраны так, что каждое из преобразований

$$\begin{aligned} x &\mapsto \varphi_1(x) \oplus x; \quad x \mapsto \varphi_2(x) \oplus x; \\ x &\mapsto \varphi_1(x) \oplus \varphi_2(x); \quad x \mapsto \varphi_1(x) \oplus \varphi_2(x) \oplus x \end{aligned}$$

является биекцией.

Схема Вигмана – Картера. Пусть определено инъективное отображение, которое ставит в соответствие сообщениям X многочлены f_X над некоторым конечным полем k . Пусть H – случайный равновероятный элемент k . Если $\deg(f_X) \leq D \ll |k|$, то значения $f_X(H)$ могут совпасть только с контролируемо малой вероятностью:

$$\begin{aligned} \mathbf{P}\{f_X(H) = f_{X'}(H)\} &= \mathbf{P}\{H \text{ – корень } f_X - f_{X'}\} \leqslant \\ &\leqslant \frac{\deg(f_X - f_{X'})}{|k|} \leqslant \frac{D}{|k|} \ll 1. \end{aligned}$$

Это наблюдение было использовано в 1981 г. М. Вигманом и Дж. Картером для организации имитозащиты. Существует несколько вариантов реализации базовой схемы Вигмана – Картера. Мы рассмотрим один из них, использованный в системе GHASH.

Пусть $n_m = n_b$ и $k = \mathbb{F}_{2^{n_b}}$. Сообщение X разобьем на блоки $X_1, \dots, X_n \in \{0, 1\}^{n_b}$, которые будем интерпретировать как элементы k . Если длина X не кратна n_b , то дополним X до границы блока нулевыми символами. Сформируем дополнительный блок X_{T+1} , который представляет первоначальную (до дополнения) длину X . По X строится многочлен

$$f_X(\lambda) = X_1\lambda^{T+1} + X_2\lambda^T + \dots + X_T\lambda^2 + X_{T+1}\lambda.$$

Различным X соответствуют различные последовательности $(X_1, \dots, X_T, X_{T+1})$, и отображение $X \mapsto f_X$ действительно является инъективным.

Для $H \in k$ значение $Y = f_X(H)$ можно найти по схеме Горнера. Для этого следует установить $Y \leftarrow 0$ и выполнить следующие итерации:

$$Y \leftarrow (Y \oplus X_t) * H, \quad t = 1, 2, \dots, T + 1.$$

Эти итерации похожи на вычисления в режиме CBC, только вместо шифрования выполняется умножение на H .

Имитовставка сообщения X определяется следующим образом:

$$Z = f_X(H) \oplus F_\theta(S).$$

Здесь $H = F_\theta(0^{n_b})$ (интерпретируется как случайный секретный элемент k), $S \in \{0, 1\}^{n_b}$.

В выражении для имитовставки зависимое от сообщения значение $f_X(H)$ «зашумляется» независимым от сообщения значением $F_\theta(S)$. Доказано, что если F – надежная криптосистема, а синхропосылки S не повторяются, то система имитозащиты также криптографически надежна. Однако как только происходит повтор синхропосылок и Виктор получает две имитовставки

$$Z = f_X(H) \oplus F_\theta(S), \quad Z' = f_{X'}(H) \oplus F_\theta(S), \quad X \neq X',$$

он может решить полиномиальное уравнение $f_X(H) \oplus f_{X'}(H) = Z \oplus Z'$ относительно секретного значения H . Поэтому требование неповторяемости синхропосылок критически важно в системах типа GHASH.

12.16. ЗАДАНИЯ

1. Разработать способ представления числа $n \in \{1, 2, \dots, n_b\}$ словом из A^{n_b} (A – произвольный алфавит).
2. Пусть a и b – случайные независимые слова с равномерным на $\{0, 1\}^n$ распределением, $c = a \boxplus b$. Доказать, что

$$\mathbf{P}\{c_i = a_i \oplus b_i\} = \frac{1}{2} + \frac{1}{2^{n-i+1}}, \quad i = 1, \dots, n.$$

3. Доказать, что всякое простое Ферма имеет вид

$$p = 2^{2^k} + 1,$$

где k – неотрицательное целое.

4. Доказать, что для любых $f, g, h \in \mathcal{F}_n$ выполняется неравенство треугольника: $\rho(f, g) \leq \rho(f, h) + \rho(g, h)$.

5. Доказать, что всякая отличная от константы аффинная функция уравновешенна.

6. Доказать, что функция $f(x) = g(x_1, \dots, x_m) \oplus h(x_{m+1}, \dots, x_n)$ является уравновешенной, если таковой является g или h .

7. Найти число уравновешенных функций от n переменных.

8. Определить числа $C_d = |\{f \in \mathcal{F}_n : \deg(f) = d\}|$, $d = 1, 2, \dots, n$.

9. Переменная x_i является *существенной* для $f \in \mathcal{F}_n$, если найдется аргумент x , для которого

$$f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \neq f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n).$$

Доказать, что число функций f , существенных от всех своих переменных, равняется

$$\sum_{i=0}^n (-1)^i \binom{n}{i} 2^{2^{n-i}}.$$

10. Пусть $f(x) = g(x_1) + h(x_2)$, $x = (x_1, x_2)$, $x_i \in \mathbb{F}_2^{n_i}$. Выразить коэффициент Уолша – Адамара $\hat{f}(u)$ через коэффициенты $\hat{g}(u_1)$ и $\hat{h}(u_2)$, $u = (u_1, u_2)$, $u_i \in \mathbb{F}_2^{n_i}$.

11. Пусть g – функция с аргументом $x \in \mathbb{F}_2^n$. Если перестановка координат x не изменяет значения g , то g – *симметрическая* функция. Доказать, что если $f \in \mathcal{F}_n$ – симметрическая функция, то и \hat{f} – симметрическая функция.

12. Найти число симметрических функций $f \in \mathcal{F}_n$.

13. Доказать, что для $\sigma \in \mathcal{F}_{n,m}$ характеристики $\text{nl}(\sigma)$, $R_{\oplus\oplus}(\sigma)$ не изменятся, если к прообразам или к образам σ применить обратимое аффинное преобразование.

14. Пусть s , $s^{-1} \in S(\mathbb{F}_2^n)$ – взаимно обратные подстановки. Доказать, что

$$R_{\oplus\oplus}(s) = R_{\oplus\oplus}(s^{-1}), \quad \text{nl}(s) = \text{nl}(s^{-1}).$$

15. Пусть $\mu_{ab}(\sigma)$ – элементы таблицы разностей для отображения $\sigma \in \mathcal{F}_{n,m}$ при выборе операций $+ = \oplus$ и $+' = \oplus$. Доказать:

1) $\mu_{ab}(\sigma)$ – четные числа;

2) $\sum_b \mu_{ab}(\sigma) = 2^n$;

3) если $n = m$ и σ – подстановка, то $\sum_a \mu_{ab} = 2^n$.

(Последние два свойства означают, что матрица $(2^{-n}\mu_{ab}(\sigma))$ – дважды стохастическая.)

16. Пусть $s \in S(\mathbb{F}_2^n)$ и $\varepsilon_{ab}(s)$ – преобладание линейной аппроксимации (a, b) для s . Доказать, что матрица $(4\varepsilon_{ab}(s)^2)$ – дважды стохастическая.

17. Пусть $s(x) = x^d$, $d \in \mathbb{N}$, $x \in \mathbb{F}_{2^n}$. Доказать, что s – биекция тогда и только тогда, когда $(d, 2^n - 1) = 1$. Доказать, что

$$R_{\oplus\oplus}(s) \leq d - 1.$$

18. Пусть $s \in S(\mathbb{F}_2^2)$. Доказать, что $\text{nl}(s) = 0$.

19. Пусть $F = \{F_\theta: \theta \in \Theta\} \subseteq S(\mathbb{F}_2^n)$ – блочная криптосистема. Пусть $F_\theta(X) = XA_\theta + b_\theta$, где A_θ – обратимая $n \times n$ матрица над полем \mathbb{F}_2 , $b_\theta \in \mathbb{F}_2^n$. Провести на F атаку по решению задачи **C2** при известном открытом тексте.

20. Пусть в SP-криптосистеме используются S -блоки $S_1, \dots, S_r \in S(\mathbb{F}_2^m)$, которые являются аффинными преобразованиями, т. е.

$$S_i(x) = xA_i + b_i,$$

где A_i – обратимая матрица порядка m , $b_i \in \mathbb{F}_2^m$. Провести на C атаку по решению задачи **C2**. Разрешается выбирать открытые тексты.

21. Пусть тактовые подстановки имеют вид

$$\begin{aligned}\Sigma_{\kappa}(X_1 \parallel X_2 \parallel X_3) &= \\ &= f_{\kappa}(X_2) \parallel (X_3 \boxplus f_{\kappa}(X_1 \boxminus X_3)) \parallel (X_1 \boxplus X_2 \boxplus f_{\kappa}(X_1 \boxminus X_3)),\end{aligned}$$

где $X_i \in \{0, 1\}^n$, f_{κ} – некоторое преобразование $\{0, 1\}^n$. Какие условия следует наложить на f_{κ} , чтобы Σ_{κ} действительно являлась биекцией? Как действует обратная подстановка Σ_{κ}^{-1} ?

22. В блочной криптосистеме Skipjack по тактовой функции $G_{\kappa} \in S(\{0, 1\}^{16})$ строится тактовая подстановка

$$\begin{aligned}A_{\kappa}(X_1 \parallel X_2 \parallel X_3 \parallel X_4) &= \\ &= (X_4 \oplus G_{\kappa}(X_1)) \parallel G_{\kappa}(X_1) \parallel X_2 \parallel X_3,\end{aligned}$$

где $X_i \in \{0, 1\}^{16}$. Описать обратную подстановку A_{κ}^{-1} . Найти неподвижные точки A_{κ} , т. е. такие прообразы $X \in \{0, 1\}^{64}$, что $A_{\kappa}(X) = X$.

23. В блочной криптосистеме Skipjack по тактовой функции $G_{\kappa} \in S(\{0, 1\}^{16})$ строится тактовая подстановка

$$\begin{aligned}B_{\kappa}(X_1 \parallel X_2 \parallel X_3 \parallel X_4) &= \\ &= (X_4 \parallel G_{\kappa}(X_1) \parallel (G_{\kappa}(X_1) \oplus X_2) \parallel X_3),\end{aligned}$$

где $X_i \in \{0, 1\}^{16}$. Описать обратную подстановку B_{κ}^{-1} . Найти неподвижные точки B_{κ} .

24. В блочной криптосистеме SMS4 (Китай) по тактовой функции $G_{\kappa} \in S(\{0, 1\}^{32})$ строится тактовая подстановка

$$\begin{aligned}\Sigma_{\kappa}(X_1 \parallel X_2 \parallel X_3 \parallel X_4) &= \\ &= X_2 \parallel X_3 \parallel X_4 \parallel X_1 \oplus (G_{\kappa}(X_2 \oplus X_3 \oplus X_4)),\end{aligned}$$

где $X_i \in \{0, 1\}^{32}$. Описать обратную подстановку Σ_{κ}^{-1} . Найти τ , при котором Σ_{κ} является τ -инволютивной подстановкой. Найти неподвижные точки Σ_{κ} .

25. Пусть тактовая подстановка блочно-итерационной криптосистемы имеет вид

$$\begin{aligned}\Sigma_{\kappa}(X_1 \parallel X_2 \parallel X_3) &= \\ &= (X_3 \oplus f_{\kappa}(X_1 \oplus X_2)) \parallel X_1 \parallel X_2,\end{aligned}$$

где $X_i \in \mathbb{F}_2^n$, $f_{\kappa} \in \mathcal{F}_{n,n}$. Найти подстановку τ , при которой Σ_{κ} является τ -инволютивной.

26. Пусть $f_{\kappa} \in S(\{0, 1\}^m)$ – тактовая функция и Σ_{κ} – соответствующая подстановка Фейстеля. Описать все неподвижные точки Σ_{κ} .

27. В SA-криптосистеме **Serpent** $n_b = 128$, $m = 4$. Действие линейное преобразования A задается следующими вычислениями над вектором $X_0 \parallel X_1 \parallel X_2 \parallel X_3$, $X_i \in \{0, 1\}^{32}$:

- | | |
|---|---|
| 1) $X_0 \leftarrow X_0 \lll 13$; | 6) $X_3 \leftarrow X_3 \lll 7$; |
| 2) $X_2 \leftarrow X_2 \lll 3$; | 7) $X_0 \leftarrow X_0 \oplus X_1 \oplus X_3$; |
| 3) $X_1 \leftarrow X_1 \oplus X_0 \oplus X_2$; | 8) $X_2 \leftarrow X_2 \oplus X_3 \oplus (X_1 \ll 7)$; |
| 4) $X_3 \leftarrow X_3 \oplus X_2 \oplus (X_0 \ll 3)$; | 9) $X_0 \leftarrow X_0 \lll 5$; |
| 5) $X_1 \leftarrow X_1 \lll 1$; | 10) $X_2 \leftarrow X_2 \lll 22$. |

Здесь $a \ll d$ – обычный (не циклический) сдвиг слова a с записью 0 в освобождающиеся разряды. Построить матрицу преобразования A и определить среднее число нулевых элементов в ее столбцах, т. е. среднее число несущественных переменных координатных функций A . Построить алгоритм вычисления образов обратного преобразования L^{-1} .

28. Временем размножения ошибки блочно-итерационной криптосистемы F называется минимальное r такое, что при некотором выборе тактовых ключей $\kappa_1, \dots, \kappa_r$ ни одна из координатных булевых функций подстановки $s = \Sigma_{\kappa_r} \dots \Sigma_{\kappa_1} \in \mathcal{F}_{n,n}$ не имеет несущественных переменных. Другими словами, при использовании r -тактового криптоизменения изменение (ошибки) в любом символе открытого текста может привести к изменению любого символа шифртекста. Оценить время размножения ошибки криптосистемы G при различных величинах циклического сдвига в тактовой функции. Является ли сдвиг на 3 оптимальным?

29. Ключ θ криптосистемы F называется слабым, если $F_\theta = F_\theta^{-1}$ или, другими словами, F_θ – инволютивная подстановка. Пусть F – d -тактовая криптосистема Фейстеля и пусть ключу θ соответствуют тактовые ключи κ_i такие, что $\kappa_i = \kappa_{d+1-i}$, $i = 1, \dots, d$. Доказать, что θ – слабый ключ.

30. Пара ключей θ, θ' криптосистемы F называется полуслабой, если $F_\theta = F_{\theta'}^{-1}$. Пусть F – d -тактовая криптосистема Фейстеля и пусть ключам θ и θ' соответствуют тактовые ключи κ_i и κ'_i такие, что $\kappa_i = \kappa'_{d+1-i}$, $i = 1, \dots, d$. Доказать, что θ, θ' – пара полуслабых ключей.

31. В криптосистеме Фейстеля CAST используется ключ $\theta = \theta_1 \parallel \dots \parallel \theta_8$, $\theta_i \in \{0, 1\}^8$. Расписание ключей:

$$\text{KS}(\theta) = (\theta_1 \parallel \theta_2, \theta_3 \parallel \theta_4, \theta_5 \parallel \theta_6, \theta_7 \parallel \theta_8, \theta_4 \parallel \theta_3, \theta_2 \parallel \theta_1, \theta_8 \parallel \theta_7, \theta_6 \parallel \theta_5).$$

Найти слабые и пары полуслабых ключей CAST.

32. В криптосистеме GOST используется ключ $\theta = K_1 \parallel \dots \parallel K_8$, $K_i \in \{0, 1\}^{32}$. Расписание ключей:

$$\text{KS}(\theta) = (K_1, K_2, \dots, K_8, K_1, K_2, \dots, K_8, K_1, K_2, \dots, K_8, K_8, \dots, K_2, K_1).$$

Найти слабые и пары полуслабых ключей GOST.

33. В криптосистеме TripleDES используется ключ $\theta_1 \parallel \theta_2 \parallel \theta_3$, $\theta_i \in \{0, 1\}^{56}$, преобразование зашифрования имеет вид $\text{DES}_{\theta_3}\text{DES}_{\theta_2}^{-1}\text{DES}_{\theta_1}$. Оценить среднее число лет, которое понадобится для атаки «грубой силой» на TripleDES, предполагая, что:

1) используется специализированное устройство RIVYERA, которое проверяет 292 млрд ключей DES в секунду;

2) согласно эвристическому закону Мура, описывающему темпы роста производительности вычислительной техники, мощность RIVYERA будет удваиваться каждые 18 месяцев.

34. Расписание ключей DES обладает следующим свойством: если

$$\text{KS}(\theta) = (\kappa_1, \dots, \kappa_{16}),$$

то $\text{KS}(\bar{\theta}) = (\bar{\kappa}_1, \dots, \bar{\kappa}_{16})$, где черта обозначает инверсию символов двоичных слов (замена 0 на 1 и 1 на 0). Доказать простое соотношение для DES:

$$F_\theta(X) = \bar{F}_{\bar{\theta}}(\bar{X}).$$

35. Найти простое соотношение для GOST (расписание ключей описано выше).

36. Пусть в криптосистеме Фейстеля, действующей на $\{0, 1\}^{2m}$, используются биективные тактовые функции. Доказать, что для любого ненулевого $\gamma \in \{0, 1\}^m$ вероятность 5-тактового дифференциала $(\gamma \parallel 0, 0 \parallel \gamma)$ равняется 0.

37. Пусть в криптосистеме Фейстеля, действующей на $\{0, 1\}^{2m}$, используются биективные тактовые функции. Доказать, что для любого ненулевого $\gamma \in \{0, 1\}^m$ преобладание 5-тактовой линейной аппроксимации $(0 \parallel \gamma, \gamma \parallel 0)$ равняется 0.

38. В каких из режимов шифрования (ECB, CBC, CFB, OFB, режим счетчика) биективность преобразования F_θ не обязательна для однозначного расшифрования?

39. В режиме OFB используемая подстановка $F_\theta \in S(\{0, 1\}^{nb})$ должна обеспечивать большой период последовательности $\Gamma_t = F_\theta(\Gamma_{t-1})$, $t = 1, 2, \dots$. Максимально большой период обеспечивает полноцикловая подстановка F_θ , для которой все элементы $\Gamma_0, \dots, \Gamma_{2^{nb}-1}$ различаются. Доказать, что среди элементов $S(\{0, 1\}^{nb})$ имеется $(2^{nb} - 1)!$ полноцикловых подстановок.

40. В ГОСТ 28147 для шифрования в режиме счетчика определена следующая функция инкремента:

$$\varphi(S_{t,1} \parallel S_{t,2}) = (S_{t,1} \boxplus' C_1) \parallel (S_{t,2} \boxplus C_2), \quad S_{t,i}, C_i \in \{0, 1\}^{32},$$

где \boxplus' – операция сложения слов-как-чисел по модулю $2^{32} - 1$ (вычет 0 по этому модулю представляется не числом 0, как обычно, а числом $2^{32} - 1$). Найти период последовательности (S_t) в зависимости от выбора констант C_1, C_2 .

41. В режиме CBC синхропосылка должна быть не только уникальной, но и непредсказуемой. Обосновать данное требование. Предположить, что Виктор может выбирать открытый текст и до своего выбора знает, какая синхропосылка будет использоваться. Виктору требуется проверить, что блок открытого текста X_t , соответствующий перехваченному блоку шифртекста Y_t , совпадает с определенным значением a .

42. Проанализировать характеристики следующего режима шифрования: $Y_t = F_\theta(X_i) \oplus Y_{t-1}$, $t = 1, 2, \dots$. Найти недостатки.

43. Имеется смарт-карта, которая реализует зашифрование блоков открытого текста X_1, X_2 на ключе θ и синхропосылке S следующим образом:

$$Y_1 = X_1 \oplus F_\theta(S);$$

$$Y_2 = X_2 \oplus X_1 \oplus F_\theta(X_1) \oplus F_\theta(S).$$

Виктор получает тройку (S, Y_1, Y_2) и смарт-карту, с помощью которой он может зашифровывать любые данные. Требуется определить X_1, X_2 (ключ θ Виктору неизвестен).

44. Пусть блоки открытого текста X_1, X_2, \dots, X_T зашифровываются по правилам:

$$Y_t = F_\theta(X_1 \boxplus X_2 \boxplus \dots \boxplus X_t) \oplus F_\theta(Y_0 \boxminus Y_1 \boxminus \dots \boxminus Y_{t-1}), \quad t = 1, 2, \dots, T$$

(Y_0 – синхропосылка). Как выполнить расшифрование?

45. Для проверки подлинности друг друга Алиса и Боб используют общий секретный ключ θ блочной криптосистемы F , действующей на $\{0, 1\}^{n_b}$. Стороны выполняют следующий протокол:

$$\begin{aligned} \text{Алиса: } R_A &\xleftarrow{R} \{0, 1\}^{n_b}; \\ \text{Алиса} \rightarrow \text{Боб: } R_A; \\ \text{Боб: } R_B &\xleftarrow{R} \{0, 1\}^{n_b}; \\ \text{Алиса} \leftarrow \text{Боб: } E_\theta(R_B \parallel R_A); \\ \text{Алиса} \rightarrow \text{Боб: } E_\theta(R_A \parallel R_B). \end{aligned}$$

Здесь E_θ – зашифрование в режиме CBC на основе криптосистемы F . При зашифровании используется нулевая синхропосылка. После получения зашифрованных сообщений каждая из сторон расшифровывает их и проверяет, что полученное слово R_A (для Алисы) или R_B (для Боба) совпадает с словом, первоначально генерированным стороной. Если это так, то стороны признают подлинность друг друга. Провести атаку на протокол.

46. В криптосистеме IDEA при зашифровании блока данных 34 раза выполняется операция \odot (умножение в \mathbb{F}_{65537}^* $\sim \{0, 1\}^{16}$). Пусть для вычисления $a \odot b$ используется следующая программа на языке Си:

```

uint32 Mul(uint32 a, uint32 b)
{
    int32 p;
    uint32 q;
    if (a == 0)                                // (*)
        p = 0x10001 - b;
    else if (b == 0)                            // (*)
        p = 0x10001 - a;
    else {                                       // (**)
        q = a * b;
        p = (q & 0xFFFF) - (q >> 16);
        if (p <= 0)
            p += 0x10001;
    }
    return (uint32)(p & 0xFFFF);
}

```

В этой программе компоненты операндов хранятся в младших разрядах 4-байтовых целых переменных беззнаковых / знаковых типа `uint32/int32`. Для вычисления результата в условиях (*) и (**) программы требуется выполнить различное число инструкций. Если Виктор имеет возможность измерять время зашифрования, то он может оценить число встретившихся нулевых операндов и упростить тем самым решение задачи криптоанализа. Предложить модификацию программы, которая позволяет запутиться от описанной атаки.

КОММЕНТАРИИ

Блочные криптосистемы описываются в книгах [2, 133, 153, 158]. Книга [41] рассчитана не только на студентов и преподавателей, но и на школьников старших классов, в ней имеется много интересных исторических подробностей и фотографий.

Огюст Керкгоффс (Kerkhoffs), голландский криптограф, написал в 1883 г. книгу «Военная криптография». В этой книге введено правило: *компрометация (криптографической) системы не должна причинять неудобства корреспондентам*, которое впоследствии трансформировалось в принцип Керкгоффса.

Русскоязычная терминология в области блочных криптосистем не до конца устоялась. Вместо «такты» и «тактовый» говорят «раунды» и «раундовый», что кажется автору не очень удачным. Вместо «S-блоки» говорят «S-боксы», что кажется совсем неприемлемым.

Булевые функции и отображения – это отдельная область исследования, подробно рассмотренная в [30]. Теория конечных полей изложена в классической книге [29]. Инверсные S -блоки введены в работах [73, 137]. Теорема 12.5 доказана в [120]. Пункт 12.5.10 основан на материалах статьи [1].

Некоторые исследователи считают оценки табл. 12.3 весьма пессимистичными (для Алисы и Боба). Имеются расчеты, согласно которым для перебора уже 2^{128} вариантов ключа требуется практически недостижимое количество энергии.

Баланс «время – память» введен в работе [108]. Разностная атака предложена в работе [74], линейная – в [129]. Подробные примеры 12.13, 12.15 принципиальны. Автор считает, что методы криptoанализа должны обязательно иллюстрироваться исчерпывающими «историями успеха».

В США при введении DES был выпущен стандарт FIPS PUB 81, в котором определялись режимы CBC, CFB и OFB (Output Feedback), лишенные недостатков ECB. Со временем наибольшее распространение из этих режимов получил CBC. В частности, шифрование CBC по умолчанию используется в протоколах SSL/TLS, которые широко применяются для защиты каналов Интернет (см. п. 16.9). Популярность CBC до конца непонятна. Возможно, она объясняется тем, что в FIPS PUB 81 режимы CFB и OFB были представлены как поточные методы шифрования, а CBC оставлен блочным. При разработке криптонаборов на основе блочной криптосистемы F отдавать предпочтение поточным методам кажется нелогичным. Кроме этого, в FIPS PUB 81 режимы CFB и OFB перегружены и представляют собой целые параметрические семейства. Необходимость учитывать в криптонаборах дополнительные параметры этих семейств могла являться дополнительным аргументом в пользу CBC.

В СССР при введении ГОСТ 28147 были определены три режима шифрования: простой замены, гаммирования с обратной связью и гаммирования. Наибольшее распространение получил режим гаммирования с обратной связью – аналог CFB. В ГОСТ 28147, в отличие от FIPS PUB 81, параметризация CFB отсутствует, режим всегда является полноблоковым.

Отличия между режимами выглядят несущественными. Но это только на первый взгляд. Изменение правил зашифрования приводит к изменению двух важных показателей (сравните с табл. 12.4):

Показатель	CBC	CFB
Можно обрабатывать открытые тексты с последним неполным блоком?	нет	да
Синхропосылка должна быть непредсказуемой?	да	нет

По обоим показателям CFB предпочтительнее CBC. Во-первых, в режиме CFB можно обрабатывать открытые тексты любой длины. Напротив, в CBC длина открытого текста должна быть кратна длине блока. Казалось бы, этот недостаток легко преодолевается выравниванием данных на границу блока перед зашифрованием и снятием выравнивания после расшифрования (см. п. 12.1). При этом если полученный после расшифрования текст не удовлетворяет формату выравнивания, то логично его отбросить, а отправителю выслать сообщение об ошибке. Такая схема обработки открытых текстов произвольной длины была применена в старых версиях SSL/TLS и оказалась уязвимой. Выяснилось, что противник может расшифровать любой шифртекст, используя другие специально подобранные шифртексты и анализируя сообщения о нарушениях формата их выравнивания после расшифрования. В последних версиях TLS выравнивание сохранено, но сообщение о нарушении формата выравнивания не отсылается. Во-вторых, в режиме CFB достаточно обеспечить уникальность синхропосылки, обеспечивать ее непредсказуемость не требуется. В CBC ситуация другая. В старых версиях SSL/TLS синхропосылка при CBC-зашифровании очередного пакета данных определялась как последний блок шифртекста из предыдущего пакета и, таким образом, была известна противнику. Оказалось, что противник может использовать знание синхропосылки для проверки того, что перехваченному блоку шифртекста соответствует открытый текст с определенным значением (см. задание 41). Правда для этого противник должен иметь возможность навязывать открытый текст, который будет передан в очередном пакете. Тем не менее в последних версиях TLS синхропосылка выбирается как случайное и, следовательно, непредсказуемое слово.

Описанные в п. 12.15 схемы построения систем имитозащиты предложены в работах [109, 160].

Г л а в а 13

ФУНКЦИИ ХЭШИРОВАНИЯ

13.1. ОПРЕДЕЛЕНИЕ И ИСПОЛЬЗОВАНИЕ

Пусть Бобу требуется подписать сообщение $X \in \{0, 1\}^*$. Боб сталкивается со следующей проблемой: алгоритм выработки ЭЦП принимает на вход слова фиксированной длины n , хотя длина X может быть произвольной. Выходом в данной ситуации является использование функции h , которая ставит в соответствие сообщению X слово $Y \in \{0, 1\}^n$. Боб использует h и подписывает не X , а Y . Важно при этом, чтобы алгоритм вычисления значений $h(X)$ имел высокое быстродействие, по крайней мере был полиномиальным (от $|X|$). Важно также, чтобы этот алгоритм был общедоступным: и Алиса, и другие абоненты информационной системы будут проверять подпись Боба и при этом также вычислять $h(X)$.

Первоначально выходное слово $h(X)$ называли *отпечатком* (Digest) входного сообщения (Message), а h , соответственно, – MD-функцией. Затем терминология изменилась.

Определение 13.1. *Функция хэширования (хэш-функция) – это отображение $h: \{0, 1\}^* \rightarrow \{0, 1\}^n$, действие которого задается общедоступным полиномиальным алгоритмом.*

Со временем функции хэширования стали использоваться не только для построения систем ЭЦП, они превратились в один из основных криптографических примитивов. Укажем наиболее важные примеры использования хэш-функций.

Контрольные суммы. Алиса вычисляет и сохраняет (с мерами защиты от модификаций) хэш-значение файла $X \in \{0, 1\}^*$. Последующее совпадение сохраненного хэш-значения с $h(X)$ служит подтверждением того, что файл X не был изменен.

Например, в дистрибутивы многих операционных систем входит программа `md5sum`. Программа реализует весьма распространенный в недавнем времени алгоритм хэширования MD5. Программа принимает на вход файл и возвращает его хэш-значение (16 октетов).

Построение ключей. По паролю $X \in \{0, 1\}^*$ Алиса строит секретный ключ $\theta = h(X)$. Этот ключ Алиса использует для шифрования или имитозащиты своих критических данных (например, контрольных сумм).

Число возможных паролей может быть сравнительно небольшим. Поэтому в процедуру построения ключей вводятся дополнительные ме-

низмы, направленные на защиту от атаки «грубой силой», направленной на определение X . Во-первых, Алиса применяет h не один, а несколько раз:

$$\theta = h^c(X) = \underbrace{h(h(\dots h(X)\dots))}_{c \text{ раз}}.$$

Регулируя число итераций c , можно сделать неприемлемо большим время, которое требуется Виктору для перебора паролей, оставляя допустимым время, затрачиваемое Алисой на генерацию ключа. Во-вторых, при построении θ Алиса кроме пароля использует синхропосылку S (ее еще называют «соль»). Ключ θ зависит от выбранной синхропосылки, и Виктор лишается возможности предварительно рассчитывать ключи для определенных классов паролей, т. е. проводить так называемые *словарные атаки*. Число итераций c и синхропосылка S являются несекретными элементами и могут сохраняться вместе с защищенными на θ данными.

Аутентификация. Алиса регистрируется на сервере Боба и отсылает ему свой пароль X по секретному каналу связи. Перед тем как предоставить Алисе доступ к ресурсам сервера, Боб проводит ее аутентификацию. Аутентификация основана на проверке знания X . Боб пересыпает Алисе случайное слово R , Алиса возвращает $Y = h(X \parallel R)$. Аутентификация завершена успешно, если полученное Бобом слово действительно совпадает с $h(X \parallel R)$.

Похожим образом выполняется, например, протокол NTLM. Протокол имеет недостаток: Виктор может перехватить (R, Y) и определить X «грубой силой», проверяя совпадение $Y \stackrel{?}{=} h(\hat{X} \parallel R)$ для паролей-кандидатов \hat{X} . Известны другие протоколы аутентификации, в которых Виктор по данным перехвата не в состоянии определить пароль X , даже если он короткий или низкоэнтропийный.

Имитозащита. Алиса преобразует функцию h в систему имитозащиты $H = \{h_\theta : \theta \in \Theta\}$, где h_θ действует из $\{0, 1\}^*$ в $\{0, 1\}^n$ (подробнее см. п. 12.15). Алиса строит H так, что значение $h_\theta(X)$ является результатом применения h (возможно многократного) к X и θ .

Например, в известной системе имитозащиты HMAC при использовании определенных h и для $\theta \in \{0, 1\}^n$ выполняется 2-кратное хэширование:

$$h_\theta(X) = h((\theta \oplus \alpha) \parallel h((\theta \oplus \beta) \parallel X)).$$

Здесь $\alpha, \beta \in \{0, 1\}^n$ – различные фиксированные слова.

Генерация псевдослучайных чисел. Алиса комбинирует свой секретный ключ θ с неповторяющейся синхропосылкой S и вычисляет хэш-значение $Y = h(\theta \parallel S)$, которое интерпретируется как псевдослучайное число. Это число Алиса использует для построения других секретных или личных ключей.

Детали генерации могут отличаться. Например, в СТБ 34.101.47-2012 «Информационные технологии и безопасность. Криптографические алгоритмы генерации псевдослучайных чисел» определен следующий алгоритм.

Алгоритм ГЕНЕРАЦИЯ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ (СТБ 34.101.47)

Вход: $T \in \mathbb{N}$, $\theta \in \{0, 1\}^n$ – ключ, $S \in \{0, 1\}^n$ – синхропосылка, $X_1, \dots, X_T \in \{0, 1\}^n$ – произвольные входные данные, которые могут повысить неопределенность выходных (текущее время, сетевая активность, данные от физических источников случайности и др.), или нулевые блоки.

Выход: $Y_1, \dots, Y_T \in \{0, 1\}^n$ – псевдослучайные числа.

Шаги:

1. $s \leftarrow S$.
 2. $r \leftarrow S \oplus 1^n$.
 3. Для $t = 1, 2, \dots, T$:
 - 3.1. $Y_t \leftarrow h(\theta \parallel s \parallel X_t \parallel r)$;
 - 3.2. $s \leftarrow s \boxplus \langle 1 \rangle_n$;
 - 3.3. $r \leftarrow r \oplus Y_t$.
 4. $Y \leftarrow Y_1 \parallel Y_2 \parallel \dots \parallel Y_T$.
 5. Возвратить Y .
-

Здесь и далее используются обозначения, введенные в гл. 12. Как и в крипtosистеме **Belt**, числа представляются двоичными словами по правилам «от младших к старшим».

13.2. ЗАДАЧИ КРИПТОАНАЛИЗА

Виктору требуется решить одну из следующих задач:

H1: по заданному $Y = h(X)$ определить X ;

H2: для заданного X найти $X' \neq X$ такое, что $h(X) = h(X')$;

H3: найти различные X и X' такие, что $h(X) = h(X')$.

В табл. 13.1 приводятся общепринятые названия этих задач, объясняется мотивация атак по решению задач, характеризуются функции хэширования, атаки на которые провести вычислительно трудно.

Обратим внимание на фразу «вычислительно трудно». Дело в том, что фиксированная длина хэш-значений не позволяет дать такие же строгие определения криптографической стойкости, как, к примеру, определение односторонности в гл. 8. Мы говорим «вычислительно трудно», имея в виду «невозможно решить за приемлемое время при современном уровне развития вычислительной техники».

Таблица 13.1

Задачи криптоанализа функций хэширования

Задача	Название	Мотивация атак	Стойкая хэш-функция
H1	Обращение	Виктор может определить пароль X по аутентификационным данным $h(X \parallel R)$	Односторонняя
H2	Определение 2-го прообраза	Виктор может осуществить подмену файла X' на файл X с такой же контрольной суммой	Свободная от коллизий
H3	Построение коллизии	Виктор может подобрать два различных документа – подлинный X и поддельный X' – с одинаковыми хэш-значениями. Виктор передает Алисе подлинный документ для ознакомления и выработки ЭЦП, а затем прилагает полученную подпись Алисы к поддельному документу	Строго свободная от коллизий

Всякий алгоритм, который находит 2-й прообраз, является также алгоритмом построения коллизии. Поэтому если h строго свободна от коллизий, то h свободна от коллизий. Следующая теорема показывает, что если h строго свободна от коллизий, то h является односторонней.

Теорема 13.1. Пусть A – алгоритм, который решает задачу **H1** для h . Тогда существует вероятностный алгоритм, который с вероятностью не менее $1/2$ решает задачу **H3**, используя одно обращение к A и еще – фиксированное количество операций.

Доказательство. Заявленный алгоритм решения **H1** выглядит следующим образом.

1. $X \xleftarrow{R} \{0, 1\}^{n+1}$.
2. $Y \leftarrow h(X)$.
3. $X' \leftarrow A(Y)$.
4. Если $X = X'$, то возвратить \perp .
5. Возвратить (X, X') .

На шаге 3 алгоритм A возвращает слово X' такое, что $h(X') = Y = h(X)$. Поэтому пара, возвращаемая на шаге 5, действительно является решением **H3**.

Введем на $\{0, 1\}^{n+1}$ отношение эквивалентности, считая эквивалентными слова с одинаковыми хэш-значениями. Пусть C – множество всех классов эквивалентности. Для $c \in C$ вероятность

$$\begin{aligned}\mathbf{P}\{X \neq A(h(X)) \mid X \in c\} &\geq [A(h(X)) \notin \{0, 1\}^{n+1} \text{ или } A(h(X)) \in c] \geq \\ &\geq \frac{|c| - 1}{|c|}.\end{aligned}$$

Пусть p – вероятность успеха алгоритма (вероятность того, что будет возвращена коллизионная пара, а не символ \perp). Имеем

$$\begin{aligned}p = \mathbf{P}\{X \neq A(h(X))\} &= \sum_{c \in C} \mathbf{P}\{X \neq M(h(X)) \mid X \in c\} \mathbf{P}\{X \in c\} \geq \\ &\geq \sum_{c \in C} \frac{|c|}{2^{n+1}} \cdot \frac{|c| - 1}{|c|} = 1 - \frac{|C|}{2^{n+1}} \geq \frac{1}{2}.\end{aligned} \quad \square$$

В обозначениях гл. 8 сказанное выше означает, что $\mathbf{H3} \leq_P \mathbf{H2}$ и $\mathbf{H3} \leq_R \mathbf{H1}$. Задача **H3** является самой простой из трех рассмотренных.

13.3. БЛОЧНО-ИТЕРАЦИОННЫЕ ФУНКЦИИ ХЭШИРОВАНИЯ

Блочно-итерационные функции хэширования, как и блочные крипто-системы, обрабатывают данные блоками некоторой фиксированной длины n_b . Обработка выполняется с помощью *шаговой функции хэширования* σ : $\{0, 1\}^{n_b+n} \rightarrow \{0, 1\}^n$. Типовой алгоритм блочно-итерационного хэширования имеет следующий вид.

АЛГОРИТМ БЛОЧНО-ИТЕРАЦИОННОЕ ХЭШИРОВАНИЕ

Вход: $X \in \{0, 1\}^*$ – сообщение.

Выход: $Y \in \{0, 1\}^n$ – хэш-значение.

Шаги:

1. Определить m – минимальное неотрицательное целое такое, что n_b делит $|X| + m$.
 2. Разбить $X \parallel 0^m$ на блоки: $X_1 \parallel \dots \parallel X_T \leftarrow X \parallel 0^m$, $X_t \in \{0, 1\}^{n_b}$.
 3. Сформировать дополнительный блок $X_{T+1} \in \{0, 1\}^{n_b}$, представляющий число $|X|$.
 4. $Y \leftarrow Y_0$, где $Y_0 \in \{0, 1\}^n$ – фиксированное *начальное хэш-значение*.
 5. Для $t = 1, 2, \dots, T+1$: $Y \leftarrow \sigma(X_t \parallel Y)$.
 6. Возвратить Y .
-

Обратим внимание на дополнительный блок X_{T+1} , который формируется на шаге 3. Обработку этого блока принято называть *усилением Меркля – Дамгарда*. Без усиления Виктор легко решает задачу **H3**, выбирая сообщения X и X' , после дописывания нулей к которым на шаге 2 получаются одинаковые слова: $X \parallel 0^m = X' \parallel 0^{m'}$.

Детали алгоритма могут отличаться. Например, в схеме хэширования, предложенной И. Дамгардом, отличия следующие:

- 1) последний блок X_{T+1} представляет не число $|X|$, а число m дописанных к X нулевых символов;
- 2) шаговая функция хэширования σ действует на $\{0, 1\}^{n_b+n+1}$, а не на $\{0, 1\}^{n_b+n}$;
- 3) итерации хэширования на шаге 5 немного меняются: сначала $Y \leftarrow \leftarrow (X_1 \parallel 0 \parallel Y)$, а затем $Y \leftarrow (X_t \parallel 1 \parallel Y)$, $t = 2, \dots, T + 1$.

Введенные для хэш-функции h задачи **H1**, **H2**, **H3** можно поставить также для шаговой функции хэширования σ . При обосновании стойкости h стараются показать, что решение некоторой задачи для h приводит к решению некоторой задачи для σ . Другими словами, пытаются свести одну задачу к другой. Если задача для σ трудна, то сведение означает, что соответствующая задача для h также трудна. Примером подобного обоснования является следующая теорема.

Теорема 13.2. *Пусть в схеме Дамгарда шаговая функция хэширования σ строго свободна от коллизий. Тогда построенная на ее основе функция h также строго свободна от коллизий.*

Доказательство. Предположим от противного, что σ строго свободна от коллизий, а h – нет, и найдены различные $X, X' \in \{0, 1\}^*$ такие, что $h(X) = h(X')$. Далее мы покажем, что по X и X' можно построить различные слова $x, x' \in \{0, 1\}^{n_b+n+1}$, которые дают коллизию для шаговой функции хэширования: $\sigma(x) = \sigma(x')$. Тем самым мы получим противоречие и докажем нужный результат.

Через Y_t обозначим значение переменной Y после завершения t -й итерации хэширования X . Все выражения, касающиеся обработки X' , снабдим штрихами. Пусть $\alpha_t = \mathbf{I}\{t > 1\}$. Рассмотрим три случая.

1. $|X| \not\equiv |X'| \pmod{n_b}$. Тогда $X_{T+1} \neq X'_{T'+1}$ и найдена коллизия:

$$\sigma(X_{T+1} \parallel \alpha_{T+1} \parallel Y_T) = h(X) = h(X') = \sigma(X'_{T'+1} \parallel \alpha_{T'} \parallel Y'_{T'}) .$$

2. $|X| = |X'|$. Имеем

$$\sigma(X_{T+1} \parallel \alpha_{T+1} \parallel Y_T) = h(X) = h(X') = \sigma(X_{T+1} \parallel \alpha_{T+1} \parallel Y'_T) .$$

Если $Y_T \neq Y'_T$, то мы получаем коллизию для σ . Если же $Y_T = Y'_T$, то перейдем к соотношению

$$\sigma(X_T \parallel \alpha_T \parallel Y_{T-1}) = Y_T = Y'_T = \sigma(X'_T \parallel \alpha_T \parallel Y'_{T-1}).$$

Если $X_T \neq X'_T$ или $Y_{T-1} \neq Y'_{T-1}$, то мы получаем коллизию. В случае двух равенств рассматриваем новое соотношение и т. д. Ясно, что в некоторый момент мы либо найдем коллизию для σ , либо приедем к равенству $X = X'$, которое противоречит первоначальному предположению.

3. $|X| \equiv |X'| \pmod{n_b}$, $|X| < |X'|$. Будем рассуждать как в предыдущем случае. Рассматривая обработку блоков X_{T+1} и $X'_{T'+1}$, X_T и $X'_{T'}$ и далее, мы либо найдем коллизию, либо достигнем пары блоков X_1 и $X'_{T'-T+1}$ и получим соотношение вида

$$\sigma(X_1 \parallel 0 \parallel Y_0) = \sigma(X'_{T'-T+1} \parallel 1 \parallel Y'_{T'-T}).$$

Данное соотношение дает коллизию, поскольку (n_b+1) -е символы прообразов обязательно отличаются. \square

13.4. ШАГОВЫЕ ФУНКЦИИ ХЭШИРОВАНИЯ

Шаговые функции хэширования строят по тем же принципам, что и подстановки шифрования блочных криптосистем: многократно комбинируя преобразования усложнения и перемешивания. При построении шаговой функции $\sigma: \{0, 1\}^{n_b+n} \rightarrow \{0, 1\}^n$ можно, в отличие от подстановок шифрования, не заботиться о биективности и, наоборот, следует продумать организацию сжатия данных. Конечно, следует также обеспечить криптографическую стойкость и учесть требования односторонности и (строгой) свободы от коллизий.

Пусть $F = \{F_\theta: \theta \in \Theta\}$ – блочная криптосистема, которая действует на $\{0, 1\}^m$ и имеет множество ключей $\Theta = \{0, 1\}^l$. Эту криптосистему можно интерпретировать как сжимающее отображение $X \parallel \theta \mapsto F_\theta(X)$ и использовать для построения шаговой функции хэширования.

Пусть длина блока хэшируемых данных, длина хэш-значения, длина блока и длина ключа криптосистемы F совпадают: $n_b = n = m = l$. В этом случае σ можно строить по F по следующей схеме:

$$\sigma(X \parallel Y) = F_{\alpha_1 X \oplus \alpha_2 Y}(\beta_1 X \oplus \beta_2 Y) \oplus \gamma_1 X \oplus \gamma_2 Y, \quad (13.1)$$

где $X, Y \in \{0, 1\}^n \sim \mathbb{F}_2^n$, $\alpha_i, \beta_i, \gamma_i \in \mathbb{F}_2$ – фиксированные константы.

Не всякий выбор констант дает криптографически стойкую функцию σ . Например, функция $\sigma(X \parallel Y) = F_Y(X \oplus Y) \oplus Y$ не является односторонней. Действительно, для заданного $Z \in \{0, 1\}^n$ можно выбрать произвольное Y и определить $X = F_Y^{-1}(Z \oplus Y) \oplus Y$. При этом $\sigma(X \parallel Y) = Z$ и задача обращения решена.

Криптографы проанализировали всевозможные варианты выбора α_i , β_i , γ и определили 12 надежных конструкций шаговых функций хэширования вида (13.1). Эти конструкции приведены в табл. 13.2.

Таблица 13.2

**Шаговые функции хэширования на основе
блочной криптосистемы F**

№	$\sigma(X \parallel Y)$	Название
1	$F_Y(X) \oplus X$	Матиаса – Мейера – Озеаса
2	$F_Y(X \oplus Y) \oplus X \oplus Y$	
3	$F_Y(X) \oplus X \oplus Y$	Миягучи – Приниля
4	$F_Y(X \oplus Y) \oplus X$	
5	$F_X(Y) \oplus Y$	Дэвиса – Мейера
6	$F_X(X \oplus Y) \oplus X \oplus Y$	
7	$F_X(Y) \oplus X \oplus Y$	
8	$F_X(X \oplus Y) \oplus Y$	
9	$F_{X \oplus Y}(X) \oplus X$	
10	$F_{X \oplus Y}(Y) \oplus Y$	LOKI
11	$F_{X \oplus Y}(X) \oplus Y$	
12	$F_{X \oplus Y}(Y) \oplus X$	

Длина блока m криптосистемы F может быть меньше n , и тогда описанные конструкции применить нельзя. Но все равно функцию σ можно строить на основе F , выполняя не одно, а с зашифрований и комбинируя результаты зашифрований с помощью перестановок и сложений.

При построении σ кроме обеспечения криптографической стойкости стараются достичь максимальной скорости хэширования. Ее часто характеризуют величиной $n_b/(cm)$, которая примерно показывает, во сколько раз хэширование медленнее шифрования. Скорость хэширования всех конструкций, рассмотренных в табл. 13.2, равняется 1. Но для $n_b > m$ такой скорости добиться не удается без ущерба для криптографической стойкости.

В СТБ 34.101.31-2011 «Информационные технологии и безопасность. Защита информации. Криптографические алгоритмы шифрования и контроля целостности» определен алгоритм хэширования на основе блочной криптосистемы *Belt* ($m = 128$, $l = 256$). Длина блока хэшируемых данных $n_b = 256$, длина хэш-значения $n = 256$. Используется шаговая функция хэширования $\sigma: \{0, 1\}^{512} \rightarrow \{0, 1\}^{256}$, которая дает скорость хэширования $2/3$. Функция σ определяется следующим алгоритмом (см. также рис. 13.1).

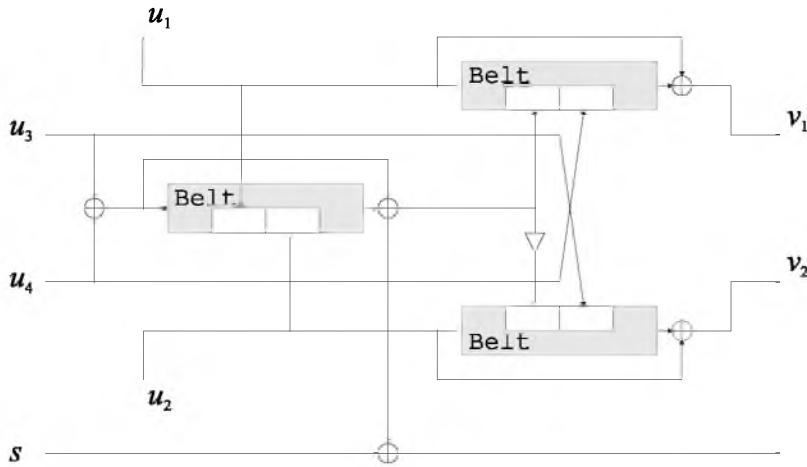


Рис. 13.1. Шаговая функция хэширования СТБ 34.101.31

АЛГОРИТМ ШАГОВАЯ ФУНКЦИЯ ХЭШИРОВАНИЯ (СТБ 34.101.31)

Вход: $u_1 \parallel u_2 \parallel u_3 \parallel u_4$, $u_i \in \{0, 1\}^{128}$.

Выход: $v_1 \parallel v_2$, $v_i \in \{0, 1\}^{128}$.

Шаги:

1. $\kappa \leftarrow \text{Belt}_{u_1 \parallel u_2}(u_3 \oplus u_4) \oplus u_3 \oplus u_4$.
2. $\theta_1 \leftarrow \kappa \parallel u_4$.
3. $\theta_2 \leftarrow (\kappa \oplus 1^{128}) \parallel u_3$.
4. $v_1 \leftarrow \text{Belt}_{\theta_1}(u_1) \oplus u_1$.
5. $v_2 \leftarrow \text{Belt}_{\theta_2}(u_2) \oplus u_2$.
6. Возвратить $v_1 \parallel v_2$.

Слово κ , которое вычисляется на шаге 1 алгоритма, используется не только в σ , но еще для обновления служебного слова s , которое участвует в формировании дополнительного блока X_{T+1} при хэшировании X .

13.5. АТАКА «ДНЕЙ РОЖДЕНИЯ»

Атака «дней рождения» направлена на решение задачи НЗ, т. е. на нахождение коллизии. Атака является универсальной и может быть применена к произвольной функции хэширования h . Пусть, как обычно, хэш-значения – это слова длины n и пусть $N = 2^n$.

АЛГОРИТМ АТАКА «ДНЕЙ РОЖДЕНИЯ»

Вход: h (задается описанием алгоритма хэширования).

Выход: $X, X' \in \{0, 1\}^*$ – коллизионная пара ($X \neq X'$, $h(X) = h(X')$).

Шаги:

1. Выбрать конечное множество $\mathcal{X} \subset \{0, 1\}^*$ мощностью $|\mathcal{X}| \gg N$.
 2. Зарезервировать массив H из N ячеек памяти. В ячейках размещаются элементы \mathcal{X} , ячейки индексируются словами из $\{0, 1\}^n$: $H[Y]$ – ячейка по индексу Y . Первоначально все ячейки заполняются символом \perp (пусто).
 3. $X \xleftarrow{R} \mathcal{X}$.
 4. $Y \leftarrow h(X)$.
 5. Если $H[Y] \neq \perp$ и $X \neq H[Y]$, то перейти к шагу 7.
 6. $H[Y] \leftarrow X$ и перейти к шагу 3.
 7. Возвратить $(X, H[Y])$.
-

Проанализируем среднее время атаки. Будем идеализировать функцию хэширования и считать, что вычисляемые в ходе атаки хэш-значения Y являются реализациями независимых случайных величин с равномерным распределением на $\{0, 1\}^n$.

Хэш-значения Y можно интерпретировать как частицы, которые случайно независимо друг от друга размещаются в N ячеек. Пусть v – номер первой частицы, которая попадает в уже занятую ячейку. Попадание означает, что $h(X) = h(X')$ для некоторых случайных $X, X' \in \mathcal{X}$. Поскольку $|\mathcal{X}| \gg N$, вероятность совпадения $X = X'$ пренебрежимо мала и попадание дает коллизионную пару (X, X') . При этом v – время ожидания коллизии, или время атаки «дней рождения», выраженное в количестве обращений к алгоритму h .

В следующей теореме мы получим точные и асимптотические выражения для Ev . Предварительно напомним, что через $N^{[t]} = N(N-1)\dots(N-t+1)$ обозначается t -я факториальная степень N (считается, что $N^{[0]} = 1$) и приведем без доказательства следующий результат.

Лемма 13.1 (метод Лапласа). *Пусть выполнены следующие условия:*

- 1) $[a, b]$ – конечный отрезок;
- 2) функция $S(x)$ бесконечное число раз дифференцируема на $[a, b]$;
- 3) $\max_{x \in [a, b]} S(x)$ достигается при $x = a$;
- 4) $S''(a) \neq 0$.

Тогда при $N \rightarrow \infty$

$$\int_a^b \exp(NS(x))dx = \sqrt{-\frac{\pi}{2NS''(a)}} e^{NS(a)}(1 + o(1)).$$

Теорема 13.3. Среднее время ожидания коллизии

$$\mathbf{E}\nu = \sum_{t=0}^N \frac{N^{[t]}}{N^t} = N \int_0^\infty e^{-Nx}(1+x)^N dx = \sqrt{\frac{\pi N}{2}}(1 + o(1))$$

(последнее равенство в асимптотике $N \rightarrow \infty$).

Доказательство. Наша цель – последовательно доказать три равенства из формулировки теоремы.

1. Имеется N^t способов размещения t частиц по N ячейкам. При этом $N^{[t]}$ способов размещения не приведут к появлению коллизии. Поэтому $\mathbf{P}\{\nu > t\} = N^{[t]}/N^t$ и

$$\mathbf{E}\nu = \sum_{t=1}^\infty t \cdot \mathbf{P}\{\nu = t\} = \sum_{t=0}^\infty \mathbf{P}\{\nu > t\} = \sum_{t=0}^N \frac{N^{[t]}}{N^t}.$$

2. Преобразуем интеграл (Γ – гамма-функция Эйлера):

$$\begin{aligned} N \int_0^\infty e^{-Nx}(1+x)^N dx &= \int_0^\infty e^{-x} \left(1 + \frac{x}{N}\right)^N dx = \\ &= \int_0^\infty e^{-x} \left(\sum_{t=0}^N \binom{N}{t} \left(\frac{x}{N}\right)^t\right) dx = \sum_{t=0}^N \binom{N}{t} \frac{1}{N^t} \int_0^\infty e^{-x} x^t dx = \\ &= \sum_{t=0}^N \binom{N}{t} \frac{1}{N^t} \Gamma(t+1) = \sum_{t=0}^N \binom{N}{t} \frac{t!}{N^t} = \mathbf{E}\nu. \end{aligned}$$

3. Разобьем искомый интеграл на два: $I_1 = N \int_0^1$ и $I_2 = N \int_1^\infty$. Для второго интеграла справедлива оценка (с учетом неравенства $\ln(1+y/2) < y/2$)

$$\begin{aligned} I_2 &= N \int_1^\infty e^{-Nx}(1+x)^N dx = \\ &= N(2/e)^N \int_0^\infty e^{-Ny}(1+y/2)^N dy < \\ &< N(2/e)^N \int_0^\infty e^{-Ny} e^{Ny/2} dy = N(2/e)^N \cdot 2/N = 2(2/e)^N = o(1). \end{aligned}$$

Для оценки первого интеграла применим метод Лапласа с функцией

$$S(x) = -x + \ln(1 + x).$$

Условия леммы выполнены, и мы получаем нужный результат. \square

Название атаки объясняется известным парадоксом дней рождения: при случайному равновероятном «размещении» дней рождения 23 учеников класса по 365 дням года у некоторых двух учеников дни рождения совпадут с вероятностью более 1/2. Парадоксом является то, что совпадения начинают происходить при достаточно небольшом числе учеников. Теорема говорит о классах из $O(\sqrt{N})$ учеников. Отметим, что оценки типа «корень квадратный из N » часто имеют место в криптографии.

13.6. МОДЕРНИЗИРОВАННАЯ АТАКА «ДНЕЙ РОЖДЕНИЯ»

Атака «дней рождения» обладает двумя серьезными недостатками. Во-первых, она имеет низкую криптографическую мотивацию. Действительно, Виктора интересуют коллизионные пары (X, X') , в которых слово X семантически близко к истинному сообщению, а слово X' – к поддельному (см. табл. 13.1), в то время как атака позволяет найти среди элементов \mathcal{X} случайную коллизионную пару. Во-вторых, для атаки требуются большие ресурсы памяти (N ячеек).

Известны модернизации атаки «дней рождения», лишенные указанных недостатков. Приведем описание одной из них.

Подготовительный этап. При подготовке к атаке выполняются следующие построения:

1. Строится разбиение $\mathcal{B}_1 \cup \mathcal{B}_2$ множества $\{0, 1\}^n$. Части разбиения равновелики: $|\mathcal{B}_1| = |\mathcal{B}_2|$.

2. Строится функция модификаций $\delta: \{0, 1\}^* \times \{0, 1\}^n \rightarrow \{0, 1\}^*$. Значение $\delta(Z, Y)$ есть результат модификации слова Z , определяемой словом Y , причем $\delta(Z, Y) \neq \delta(Z, Y')$, если $Y \neq Y'$. Модификации не изменяют сути Z и могут, например, состоять в добавлении или удалении незначащих символов.

3. Строится функция $\varphi: \{0, 1\}^n \rightarrow \{0, 1\}^n$,

$$\varphi(Y) = \begin{cases} h(\delta(X, Y)), & \text{если } Y \in \mathcal{B}_1, \\ h(\delta(X', Y)), & \text{если } Y \in \mathcal{B}_2. \end{cases}$$

Выполнив построения, можно выбрать $Y_0 \in \{0, 1\}^n$ и определить последовательность $Y_t = \varphi(Y_{t-1}) = \varphi^t(Y_0)$, $t = 1, 2, \dots$. Данная последовательность

(траектория) является периодической, пусть r – ее минимальный период и t_0 – предпериод (рис. 13.2).

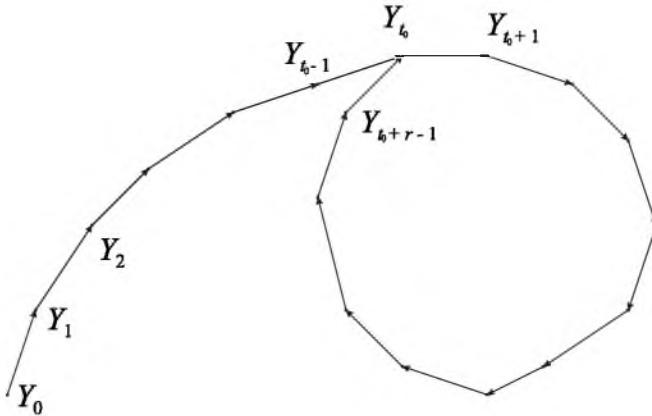


Рис. 13.2. Периодическая траектория

Если $t_0 \neq 0$, то (Y_{t_0-1}, Y_{t_0+r-1}) – коллизионная пара для φ :

$$\varphi(Y_{t_0-1}) = \varphi(Y_{t_0+r-1}).$$

В частности, если Y_{t_0-1} и Y_{t_0+r-1} лежат в разных множествах \mathcal{B}_1 и \mathcal{B}_2 , например, $Y_{t_0-1} \in \mathcal{B}_1$ и $Y_{t_0+r-1} \in \mathcal{B}_2$, то модификациям $\delta(X, Y_{t_0-1})$ и $\delta(X', Y_{t_0+r-1})$ истинного и поддельного сообщений соответствует одинаковое хэш-значение Y_{t_0} .

Известно, что если φ выбрано случайно равновероятно из множества всех преобразований на $\{0, 1\}^n$, то

$$\mathbf{E} t_0 = \frac{1}{2} \sqrt{\frac{\pi N}{2}} + O(1), \quad \mathbf{E} r = \frac{1}{2} \sqrt{\frac{\pi N}{2}} + O(1),$$

т. е. для обнаружения коллизии снова потребуется выполнить порядка \sqrt{N} хэширований в среднем.

Оперативный этап. Оперативный этап проводится следующим образом (реализацию шага 2 рассмотрим немного позже):

$$1. Y_0 \xleftarrow{R} \{0, 1\}^n.$$

2. Определить минимальный период r последовательности

$$Y_t = \varphi^t(Y_0), \quad t = 1, 2, \dots.$$

$$3. A \leftarrow Y_0, B \leftarrow \varphi^r(Y_0).$$

4. Если $A = B$, то возвратить \perp ($t_0 = 0$).
5. Пока $\varphi(A) \neq \varphi(B)$: $A \leftarrow \varphi(A)$, $B \leftarrow \varphi(B)$.
6. Если $A, B \in \mathcal{B}_1$ или $A, B \in \mathcal{B}_2$, то возвратить \perp (коллизия для φ найдена, но оказалась бесполезной).
7. Если $A \in \mathcal{B}_1$ и $B \in \mathcal{B}_2$, то возвратить $(\delta(X, A), \delta(X', B))$.
8. Если $A \in \mathcal{B}_2$ и $B \in \mathcal{B}_1$, то возвратить $(\delta(X, B), \delta(X', A))$.

При вероятностной идеализации φ (φ выбирается наудачу из множества всех преобразований $\{0, 1\}^n$) и достаточно большом n вероятность события $\{t_0 = 0\}$ незначительна, а переменные A и B попадут в различные множества \mathcal{B}_1 и \mathcal{B}_2 с вероятностью, близкой к $1/2$. Таким образом, атака завершится успехом примерно в половине случаев. При необходимости атаку можно повторить, повышая вероятность успеха.

Остается определиться с реализацией шага 2. Для определения минимального периода последовательности (Y_t) можно использовать следующий алгоритм, который работает на совсем небольшой памяти.

АЛГОРИТМ БРЕНТА

Вход: $Y_0 \in \{0, 1\}^n$, φ – преобразование $\{0, 1\}^n$ (задается алгоритмически).

Выход: r – минимальный период последовательности Y_0 , $Y_1 = \varphi(Y_0)$, $Y_2 = \varphi(Y_1)$, \dots .

Шаги:

1. $A \leftarrow Y_0$.
 2. Для $i = 0, 1, \dots$ выполнить:
 - 2.1. $B \leftarrow \varphi(A)$;
 - 2.2. Для $r = 1, 2, \dots, 2^i$:
 - если $A = B$, то возвратить r , иначе $B \leftarrow \varphi(B)$;
 - 2.3. $A \leftarrow B$.
-

На i -й итерации алгоритма проверяется совпадение сохраненного значения Y_t , $t = 2^i - 1$, с вычисляемыми значениями Y_τ , $\tau = 2^i, 2^i + 1, \dots, 2^{i+1} - 1$:

Y_0	Y_1	Y_3	\dots	Y_{2^i-1}	\dots
Y_1	Y_2, Y_3	Y_4, Y_5, Y_6, Y_7	\dots	$Y_{2^i}, \dots, Y_{2^{i+1}-1}$	\dots

Первое найденное совпадение $Y_t = Y_\tau$ означает, что минимальный период $r = t - \tau$. Действительно, если минимальный период $r' < r$, то $Y_t = Y_{t+r'}$ и совпадение $Y_t = Y_\tau$ не является первым.

13.7. ЗАДАНИЯ

1. Пусть p – простое, g – примитивный элемент \mathbb{F}_p^* , $h: \{0, 1\}^* \rightarrow \mathbb{F}_p^*$, $x \mapsto g^x \bmod p$ (слово x отождествляется с числом). Сформулировать для h криптоаналитические задачи. Какие задачи допускают простое решение? (Считать, что задача дискретного логарифмирования в \mathbb{F}_p^* является труднорешаемой.)

2. Пусть $n = pq$ – модуль RSA, $h: \{0, 1\}^* \rightarrow \mathbb{Z}_n$, $x \mapsto x^2 \bmod n$ – функция хэширования (слово x отождествляется с числом). Сформулировать для h криптоаналитические задачи. Какие задачи допускают простое решение? (Считать, что задача факторизации n является труднорешаемой.)

3. Пусть функция $h_1: \{0, 1\}^{2^n} \rightarrow \{0, 1\}^n$ является строго свободной от коллизий. Доказать, что строго свободной от коллизий будет также всякая функция

$$h_i: \{0, 1\}^{2^{i-1}n} \rightarrow \{0, 1\}^n;$$

$$X_1 \parallel X_2 \mapsto h_i(h_{i-1}(X_1) \parallel h_{i-1}(X_2)),$$

где $X_1, X_2 \in \{0, 1\}^{2^{i-1}n}$, $i = 2, 3, \dots$.

4. Количество образов блочно-итерационной хэш-функции h не больше количества образов ее шаговой функции хэширования σ . Доказать, что имеется

$$\sum_{i=0}^N (-1)^i \binom{N}{i} (N-i)^R$$

различных сюръективных отображений $\{0, 1\}^{n_b+n} \rightarrow \{0, 1\}^n$. Здесь $N = 2^n$, $R = 2^{n_b+n}$.

5. Выбрать произвольную шаговую функцию хэширования из табл. 13.2 и показать, что атаки на нее сводятся к атакам на криптосистему F .

6. Предложить способ обращения и построения коллизии для шаговой функции хэширования

$$\sigma(X \parallel Y) = F_Y(X \oplus Y) \oplus Y.$$

7. Предложить способ обращения и построения коллизии для шаговой функции хэширования

$$\sigma(X \parallel Y) = F_{X \oplus Y}(X) \oplus X \oplus Y.$$

8. Предложить способ обращения и построения коллизии для шаговой функции хэширования

$$\sigma(X \parallel Y) = F_{X \oplus Y}(Y) \oplus X \oplus Y.$$

9. Алгоритм Флойда поиска совпадения элементов периодической (с минимальным периодом r и предпериодом t_0) последовательности Y_0, Y_1, \dots состоит в проверке равенств $Y_t = Y_{2t}$ для $t = 1, 2, \dots$. Доказать, что номер первого совпадения

$$t = r(1 + \lfloor t_0/r \rfloor).$$

10. Пусть $h: \{0, 1\}^* \rightarrow \{0, 1\}^n$ – блочно-итерационная функция и ее шаговая функция хэширования σ такова, что по заданным X_t и $\sigma(X_t \parallel Y)$ можно легко найти Y . Разработать атаку на h , направленную на решение задачи **H2**. Время атаки должно иметь порядок $2^{n/2}$. Считать, что слово X , которое входит в постановку задачи, состоит из не менее чем двух блоков.

11. Пусть в системе имитозащиты I (см. п. 12.15) используется блочно-итерационная хэш-функция $h: \{0, 1\}^* \rightarrow \{0, 1\}^n$, $\Theta = \{0, 1\}^n$, ключом является начальное хэш-значение Y_0 . Разработать атаку на I , направленную на решение задачи **M3**. В ходе атаки разрешается выбирать сообщения, подлежащие имитозащите.

КОММЕНТАРИИ

Функции хэширования описываются в книгах [2, 133, 153, 158]. Самые современные методы построения и оценки стойкости функций хэширования представлены в материалах конкурса SHA-3, проведенного в США в 2007 – 2012 гг. (см. <http://csrc.nist.gov/groups/ST/hash/sha-3/index.html>).

Среди алгоритмов построения ключа по паролю наибольшее распространение получил алгоритм PBKDF2, определенный в PKCS#5. Этот алгоритм включен в СТБ 34.101.45-2013 «Информационные технологии и безопасность. Алгоритмы электронной цифровой подписи и транспорта ключа на основе эллиптических кривых».

В некоторых алгоритмах построения ключа по паролю используются большие массивы вспомогательных данных. Делается это для того, чтобы алгоритмы трудно было перенести на спецустройства с высоким быстродействием, но малой оперативной памятью.

Кроме задач, перечисленных в табл. 13.1, существуют и другие. Например, задача построения r -коллизии

H4: найти r различных сообщений X_1, \dots, X_r таких, что

$$h(X_1) = \dots = h(X_r).$$

Если h – идеальная функция хэширования, то для построения r -коллизии требуется порядка $N^{(r-1)/r}$ операций ($N = 2^n$). А. Жу в [113] показал, что если h является блочно-итерационной, то атаку по решению **H4** можно провести за существенно меньшее время: порядка $\log r \cdot \sqrt{N}$. Задача **H4** является достаточно искусственной. Тем не менее атака Жу показывает, что итераци-

онная структура функций хэширования является потенциальным источником слабостей и должна каким-то образом корректироваться.

В 2005 г. группа китайских исследователей под руководством С. Ванг разработала метод построения коллизий для функции хэширования MD5. Метод является чрезвычайно эффективным и позволяет строить коллизии почти в реальном времени. Коллизии MD5 были использованы в компьютерном вирусе Flame (2012 г.) для обхода механизмов защиты операционной системы Windows. Предложенный группой Ванг метод, как оказалось, можно применить к схожим хеш-функциям, например, к распространенной функции SHA-1. На сегодняшний день самая эффективная атака по построению коллизии для SHA-1 требует около 2^{60} хэширований. По некоторым оценкам, такой объем вычислений станет подъемным для обычного университетского вычислительного кластера в 2021 г.

Сумма для $\mathbf{E}\nu$, которая фигурирует в формулировке теоремы 13.3, без первого члена $N^{[0]}/N^0$ введена Д. Кнутом в [22] и названа им Q -функцией Раманужана. Более точная асимптотика для Q -функции:

$$Q(N) \sim \sqrt{\frac{\pi N}{2}} - \frac{1}{3} + \frac{1}{12} \sqrt{\frac{\pi}{2N}} - \frac{4}{135N} + \dots$$

Пусть поиск коллизии ведется для блочно-итерационной хеш-функции с $n_b = n$. В работе [62] показано, что если искать коллизию на сообщениях одинаковой длины, то при идеализации шаговой функции хэширования время ожидания коллизии уменьшается до

$$\mathbf{E}\nu = \frac{\sqrt{\pi N}}{2} + \frac{5}{6} + o(1),$$

т. е. примерно в $\sqrt{2}$ раза. В рассуждениях используется двойная Q -функция:

$$Q(M, N) = \sum_{k=0}^{\min(M, N)} \frac{M^{[k]} N^{[k]}}{M^k N^k}.$$

Г л а в а 14

КРИПТОСИСТЕМЫ С ОТКРЫТЫМ КЛЮЧОМ

14.1. RSA-КРИПТОСИСТЕМА

Идею, лежащую в основе криптосистем с открытым ключом, высказали в 1975 г. У. Диффи и М. Хеллман. Они ввели понятие односторонней функции с секретом. Более подробно об этом речь пойдет в следующей главе. Это дало принципиальную возможность разрабатывать криптосистемы с открытым ключом, в которых алгоритм шифрования является общедоступным, и поэтому нет необходимости в секретных каналах связи для предварительного обмена ключами.

Таким образом, эти криптосистемы принципиально отличаются от криптосистем, описанных в предыдущей главе. Их называют *асимметричными*, или *двухключевыми*, поскольку они имеют два ключа: несекретный – для зашифрования и секретный – для расшифрования.

Метод шифрования RSA предложен в 1977 г. Р. Ривестом, А. Шамиром и Л. Адлеманом как реализация идеи У. Диффи и М. Хеллмана.

Опишем процесс шифрования сообщений. Исходный текст должен быть переведен в числовую форму. Метод преобразования текста в числовую форму считается известным и необязательно держится в секрете. В результате текст представляется в виде одного большого числа. Затем полученное число разбивается на части так, чтобы каждая из них была числом в промежутке от 0 до N , где N будет выбрано ниже. Процесс зашифрования одинаков для каждой части. Поэтому можно считать, что исходный текст представлен числом x таким, что $0 < x < N$.

Предположим, что некоторый пользователь (назовем его B) желает, чтобы ему передали секретное сообщение. Для этого он делает общедоступными два числа: N и e (открытый ключ), которые подчинены двум условиям:

1) $N = pq$, где p и q – большие простые числа, которые B держит в секрете. Числа p и q обычно выбираются порядком не ниже, чем 2^{256} ;

2) число e берется взаимно простым с $\varphi(N) = (p - 1)(q - 1)$.

Пользователь A , отправляющий сообщение x , шифрует его следующим образом: $E(x) \equiv x^e \pmod{N}$. Это и есть зашифрованный текст, который получает B .

Чтобы восстановить исходный текст, B поступает так:

1) находит число d такое, что $1 \leq d \leq N - 1$ и $ed \equiv 1 \pmod{\varphi(N)}$. Это сравнение разрешимо единственным образом, поскольку $(e, \varphi(N)) = 1$.

Здесь как раз и проявляется особенность RSA. Для решения сравнения $ed \equiv 1 \pmod{\varphi(N)}$ пользователь B должен вычислить $\varphi(N)$, что для него не составит труда, так как $\varphi(N) = \varphi(pq) = \varphi(p)\varphi(q) = (p-1)(q-1)$. Любой другой пользователь, который знает только N , вынужден находить p и q , т. е. разлагать число N на простые множители, а эта задача при больших p и q имеет большую вычислительную сложность;

2) далее, имея в распоряжении число y , пользователь B вычисляет величину $D(y) \equiv y^d \pmod{N}$, которая и есть представление x исходного текста. Действительно, применяя теорему Эйлера, получаем

$$y^d \equiv x^{ed} \equiv x^{\varphi(N)k+1} \equiv (x^{\varphi(N)})^k x \equiv x \pmod{N}.$$

Замечание 14.1. Из малой теоремы Ферма вытекает (см. главу 2), что $x^{p-1} \equiv 1 \pmod{p}$ и $x^{q-1} \equiv 1 \pmod{q}$. Тогда если положить $l = \varphi(N) = (p-1)(q-1)$, то выполнено сравнение $x^l \equiv 1$ как по модулю p , так и по модулю q и, следовательно, по модулю $\varphi(N)$. Из этого вытекает, что любые числа e' и d' с условиями $e' \equiv e \pmod{l}$ и $d' \equiv d \pmod{l}$ также будут соответственно открытым и секретным ключами.

Корректность работы системы RSA

Пусть параметры $(N = pq; e; d)$ задают RSA-криптосистему. Обозначим через $G = \mathbb{Z}_N^*$ мультиплексивную группу кольца \mathbb{Z}_N . Она имеет порядок $\varphi(N) = (p-1)(q-1)$. Вероятность того, что случайно взятое число $0 < x < N$ не принадлежит G , равна $\frac{1}{p} + \frac{1}{q}$, что при больших p и q практически невозможно. С этой точки зрения можно считать, что все наши сообщения выбираются из G . Очевидно, что отображения E и D переводят элементы из G в элементы из G . Более того, они являются автоморфизмами G , т. е. взаимно однозначными отображениями с условием $E(x_1x_2) = E(x_1)E(x_2)$ и $D(x_1x_2) = D(x_1)D(x_2)$. Кроме этого E и D являются взаимно обратными, т. е. $E(D(x)) = D(E(x)) = x$, что обеспечивается условием $ed \equiv 1 \pmod{\varphi(N)}$. Таким образом, секретный и несекретный ключи можно менять местами.

Обычно для обеспечения стойкости RSA предъявляют следующие требования к выбору параметров системы:

- 1) простые числа p и q должны быть большими;
- 2) разность $|p - q|$ должна быть большой;
- 3) числа $p \pm 1, q \pm 1, r - 1, s - 1$ должны содержать большой простой множитель;
- 4) число $(p - 1, q - 1)$ должно быть небольшим.

Далее будет показано, что нельзя выбирать ключи e и d с короткой секретной экспонентой: необходимо придерживаться «нормального» случая, когда d – однопорядковая величина с $L = [p - 1, q - 1]$.

Если какое-либо из требований 1–4 не выполнено, то существует способ довольно быстро взломать эту криптосистему, что мы обсудим в следующих пунктах (см. также задания к данной главе).

14.2. ВОЗМОЖНЫЕ АТАКИ НА КРИПТОСИСТЕМУ RSA

Пусть параметры $(N = p q; e; d)$ задают RSA-криптосистему. Для ее взлома, т. е. для прочтения секретного сообщения или подделки подписи, необходимо по известным входным N , e и зашифрованному или подписенному сообщению y найти такое $x \in \mathbb{Z}_N^*$, что

$$y \equiv x^e \pmod{N}. \quad (14.1)$$

Например, можно пытаться решить сравнение (14.1) при конкретном y или y из некоторого класса $M \subset G$, чтобы затем, используя гомоморфность отображения $D(x)$, решить его на более широком множестве. Например, если $M = \{p_1, \dots, p_k\}$ – множество малых простых и каким-либо образом удалось решить сравнение (14.1), то это сравнение становится легко разрешимым для всех y , которые являются произведениями чисел из M . Более детально это означает следующее. Пусть имеем набор пар $XY = \{(x_1, y_1), \dots, (x_k, y_k)\}$ с условием $x_i^e \equiv y_i \pmod{N}$. Пусть $1 < y < N$ и $(y, N) = 1$. Если каким-либо образом удастся представить y в виде

$$y \equiv y_1^{s_1} \cdots y_k^{s_k} \pmod{N}$$

с целыми s_j , то решением (14.1) будет

$$x \equiv x_1^{s_1} \cdots x_k^{s_k} \pmod{N}.$$

Однако данный подход не менее трудный, чем поиск алгоритма решения сравнения (14.1) при любом y .

Еще один вид атаки (он сводится к поиску ключа d' для дешифрования) – метод повторного шифрования. Он состоит в следующем. Пусть e – открытая экспонента, y – зашифрованное сообщение. Это значит, что для некоторого x верно (14.1). Строим последовательность y_i :

$$y_1 = y;$$

$$y_i \equiv y_{i-1}^e \pmod{N}, \quad i > 1.$$

Это значит, что $y_m \equiv y^{e^m} \pmod{N}$. Поскольку $(e, \varphi(N)) = 1$, то существует натуральное m такое, что $e^m \equiv 1 \pmod{\varphi(N)}$. Но тогда

$$y^{e^m-1} \equiv 1 \pmod{N},$$

откуда вытекает

$$y^{e^m} \equiv y \pmod{N}.$$

Тогда y_{m-1} будет решением (14.1).

14.3. СТОЙКОСТЬ RSA ПРОТИВ АТАКИ ПОВТОРНОГО ШИФРОВАНИЯ

Проанализируем более детально *метод повторного шифрования* и покажем необходимость соблюдения требований на выбор p и q для обеспечения стойкости.

Пусть $N = pq$, e , d – параметры системы RSA, y – зашифрованное сообщение, $y \in \mathbb{Z}_N^*$.

Для достижения успеха необходимо путем m -кратного шифрования необходимо и достаточно, чтобы порядок $\text{ord } y$ делил $e^m - 1$.

Теорема 14.1. *Пусть $p - 1 = rk$ и $q - 1 = sl$, где r и s – различные простые, $(r, k) = 1$, $(s, l) = 1$. Тогда вероятность того, что случайно взятый элемент $y \in G$ имеет порядок t , делящийся на rs , равна $\left(1 - \frac{1}{r}\right)\left(1 - \frac{1}{s}\right)$.*

Доказательство. Группа G есть прямое произведение двух своих циклических подгрупп G_1 и G_2 , которые соответственно имеют порядки $p - 1$ и $q - 1$. Всякий $y \in G$ представляется однозначно в виде произведения $y = ab$, где $a \in G_1$, $b \in G_2$. Более того, его порядок равен $\text{ord } y = [\text{ord } a, \text{ord } b]$. Тогда каждой паре (t_1, t_2) , где $t_1|k$, $t_2|l$, соответствует $\varphi(rt_1)\varphi(st_2)$ элементов группы G порядка $rs(t_1, t_2)$. Различным парам соответствуют различные наборы элементов, и объединение таких наборов по всем парам (t_1, t_2) исчерпывает множество тех y , у которых порядок делится на rs . Ввиду того что количество элементов в циклической группе, имеющих порядок w , равно $\varphi(w)$, получаем, что количество тех y , для которых $rs|\text{ord } y$, равно

$$\begin{aligned} \sum_{t_1|k} \sum_{t_2|l} \varphi(rt_1)\varphi(st_2) &= \sum_{t_1|k} \sum_{t_2|l} \varphi(r)\varphi(t_1)\varphi(s)\varphi(t_2) = \\ &= (r-1)(s-1) \sum_{t_1|k} \varphi(t_1) \sum_{t_2|l} \varphi(t_2) = (r-1)(s-1)kl. \end{aligned}$$

Всего в группе G имеется $rskl$ элементов. Тогда искомая вероятность равна

$$\frac{(r-1)(s-1)kl}{rskl} = \left(1 - \frac{1}{r}\right)\left(1 - \frac{1}{s}\right). \quad \square$$

Из теоремы 14.1 вытекает, что требуется взять примерно $\min(r, s)$ элементов, чтобы порядок одного из них не делился хотя бы на одно из чисел r, s . Поэтому без ограничения общности можно считать, что $sr \mid \text{ord } y$.

Аналогично доказывается, что вероятность того, что ни s , ни r не делят $\text{ord } y$, равна $1 - 1/rs$.

Если предположить, что $sr \mid \text{ord } y$, то успех в атаке методом повторного шифрования будет достигнут, только если $rs \mid (e^m - 1)$. Число $e < \varphi(N)$ выбирают взаимно простым с $\varphi(N)$. Иначе говоря, e – элемент группы $H = \mathbb{Z}_{\varphi(N)}^*$. Если r_1 – большой простой делитель числа r , s_1 – большой простой делитель числа s , то аналогично показывается, что с вероятностью $\left(1 - \frac{1}{r_1}\right) \left(1 - \frac{1}{s_1}\right)$ число e имеет порядок в группе H , делящийся на $r_1 s_1$. В этом случае атака методом повторного шифрования будет успешной только при условии $r_1 s_1 \mid m$, что требует огромных вычислительных затрат.

Если два последних условия в требовании 3 не выполнены, то вероятность того, что случайный ключ e будет иметь небольшой порядок, возрастает. Это приводит к тому, что если в результате ряда атак с различными e при некотором $e = e'$ мы достигаем успеха, то с большой вероятностью можно ожидать, что $e_1^{m_1} \equiv 1 \pmod{\varphi(N)}$ с найденным относительно небольшим m_1 . Тогда число $e_1^{m_1}$ можно использовать для факторизации N методом, описанным в следующем пункте.

14.4. ПОИСК СЕКРЕТНОГО КЛЮЧА d И ФАКТОРИЗАЦИЯ МОДУЛЯ N

Если известно разложение на простые множители числа N , то можно определить секретный ключ d с помощью полиномиального от длины N и e алгоритма. Оказывается, верно и обратное: существует эффективный вероятностный полиномиальный от длины d, e, N алгоритм разложения на множители числа N , если известно натуральное d с условием $ed \equiv 1 \pmod{\varphi(N)}$. Только в этом смысле можно говорить об эквивалентности задач взлома RSA путем определения секретного ключа d (или вообще любого числа d' такого, что $D(x) \equiv x^{d'} \pmod{N}$), и путем факторизации. Это значит, что нет алгоритма, который бы определял секретный ключ d со сложностью, превосходящей сложность «самого лучшего» алгоритма факторизации в полиномиальное количество раз. Данный результат опирается на следующую теорему [118].

Теорема 14.2. *Если тройка (N, e, d) образует RSA-криптосистему и известно натуральное d такое, что $ed \equiv 1 \pmod{\varphi(N)}$, то существует эффективный вероятностный алгоритм полиномиальной сложности для факторизации N .*

Доказательство. Пусть известны параметры e и d , удовлетворяющие условию теоремы. Тогда $s = ed - 1$ делится на $\varphi(N)$. Следовательно, для любого $x \in \mathbb{Z}_N^* = G$ верно

$$x^s \equiv 1 \pmod{N}.$$

Запишем $s = 2^t u$, где u – нечетное, и рассмотрим множество $A = G \setminus B$, где B состоит из тех $x \in G$, для которых либо при некотором целом $j \in \{1, \dots, t-1\}$ верно $x^{2^j u} \equiv -1 \pmod{N}$, либо $x^u \equiv 1 \pmod{N}$.

Для любого элемента $a \in A$ выберем число k наименьшим с условием $a^{2^k u} \equiv 1 \pmod{N}$. Поскольку $a \notin B$, то $k \geq 1$. Тогда положим $b = a^{2^{k-1} u} \pmod{N}$. Следовательно,

$$b^2 \equiv 1 \pmod{N}, \quad b \not\equiv \pm 1 \pmod{N}.$$

Поэтому $(b-1, N)$ – собственный делитель N . Тем самым достигается факторизация.

Далее запишем $p-1 = 2^{\nu_1} u_1$, $q-1 = 2^{\nu_2} u_2$, где u_1, u_2 – нечетные числа. Положим $\nu = \min(\nu_1, \nu_2)$ и $K = (u, u_1)(u, u_2)$. Используя теорию сравнений, можно получить оценки для количества решений сравнений $x^u \equiv 1 \pmod{N}$ и $x^{2^j u} \equiv -1 \pmod{N}$, $j \leq t-1$.

Сравнение $x^u \equiv 1 \pmod{N}$ равносильно системе

$$x^u \equiv 1 \pmod{p}, \quad x^u \equiv 1 \pmod{q}.$$

Проиндексируем первое сравнение. Получим $u \text{ ind } x \equiv 0 \pmod{2^{\nu_1} u_1}$. Следовательно, оно имеет (u, u_1) решений. Аналогично, второе сравнение имеет (u, u_2) решений. С учетом $(p, q) = 1$ и китайской теоремы об остатках, сравнение $x^u \equiv 1 \pmod{N}$ имеет $K = (u, u_1)(u, u_2)$ решений. Этим же приемом можно найти и количество решений сравнения $x^{2^j u} \equiv -1 \pmod{N}$. Оно оказывается равным $4^j K$, $j < \nu$. Поэтому

$$|B| = \left(1 + 1 + 4 + 4^2 + \dots + 4^{\nu-1}\right) K = \left(1 + \frac{4^\nu - 1}{3}\right) K.$$

Путем элементарных вычислений находим, что

$$|B| = \left(1 + \frac{4^\nu - 1}{3}\right) K \leq \frac{\varphi(N)}{2} = \frac{1}{2}|G|.$$

Из этого вытекает, что вероятность того, что случайно взятый элемент $x \in G$ будет лежать в A , не менее $1/2$. Тогда за m попыток мы с вероятностью $\geq 1 - 1/2^m$ встретим элемент из A и найдем факторизацию N по алгоритму, вытекающему из доказательства. \square

Замечание 14.2. Эквивалентность задач факторизации и поиска ключа d означает, что нельзя строить многопользовательскую RSA-криптосистему, чтобы разные пользователи имели свои ключи с одним и тем же модулем N . Потеря стойкости RSA-криптосистемы с параметрами N, e, d вследствие потери секретности ключа d влечет необходимость замены не только ключа d , но и модуля N .

14.5. БИТЫ КЛЮЧЕЙ В RSA-КРИПТОСИСТЕМЕ

Изучим, как секретность отдельных битов сообщения x влияет на секретность сообщения в целом.

Пусть $y = E(x)$, где E – преобразование криптосистемы с параметрами $N = pq, e, d$. Введем функции:

$$p(y) = \begin{cases} 0, & x - \text{четное}, \\ 1, & x - \text{нечетное}; \end{cases}$$

$$h(y) = \begin{cases} 0, & 0 < x < N/2, \\ 1, & N/2 < x < N. \end{cases}$$

Теорема 14.3. Следующие утверждения эквивалентны:

- 1) существует эффективный алгоритм вычисления функции $p(y)$;
- 2) существует эффективный алгоритм вычисления функции $h(y)$;
- 3) существует эффективный алгоритм вычисления функции $D(y)$.

Доказательство. Пусть $0 < x < N$, тогда легко проверить утверждение

$$x < \frac{1}{2}N \Leftrightarrow 2x(\text{mod } N) - \text{четно}. \quad (14.2)$$

Поэтому

$$h(y) = p(yE(2)).$$

Если в этом равенстве y заменить на $yE(2^{-1})$, то $p(y) = h(yE(2^{-1}))$. Тем самым установлена эквивалентность первых двух утверждений теоремы.

Вычислим $x = D(y)$, если известен алгоритм для вычисления $h(y)$. Обобщая утверждение (14.2), можно показать, что

$$4x(\text{mod } N) - \text{четно} \Leftrightarrow x \in \left[0, \frac{N}{4}\right) \cup \left[\frac{N}{2}, \frac{3N}{4}\right),$$

$$8x(\text{mod } N) - \text{четно} \Leftrightarrow x \in \left[0, \frac{N}{8}\right) \cup \left[\frac{N}{4}, \frac{3N}{8}\right) \cup \left[\frac{N}{2}, \frac{5N}{8}\right) \cup \left[\frac{3N}{4}, \frac{7N}{8}\right) \text{ и т. д.}$$

Располагая несекретным преобразованием E крипtosистемы, можно легко вычислить $E(2), E(4), E(8), \dots$. Тем самым находим $E(2x), E(4x), E(8x), \dots$, поскольку $E(x_1x_2) = E(x_1)E(x_2)$. Алгоритм h при этом будет давать на k -м шаге оценку для x :

$$\frac{i}{2^{k+1}}N < x < \frac{i+1}{2^{k+1}}N.$$

Этот интервал при достаточно большом K будет содержать лишь одно целое число. \square

14.6. ТЕОРЕМА М. ВИНЕРА О МАЛОЙ СЕКРЕТНОЙ ЭКСПОНЕНТЕ

М. Винером было показано, что при небольшом секретном RSA-ключе d имеется возможность эффективно его вычислить. С учетом алгоритма, описанного в п. 9.3.2, это дает возможность факторизовать RSA-модуль. Отметим, что в RSA-криптографии простые числа p и q выбираются по возможности близкими, а это в силу теоремы Чебышева автоматически приводит к выполнению условия $q < p < 2q$.

Теорема 14.4. Пусть задана RSA-криптосистема с алгоритмами $N = q, ed \equiv 1(\text{mod } \varphi)(N), q < p < 2q, d < \frac{1}{3}N^{1/4}$. Тогда d эффективно вычислимо.

Доказательство. Оценим величину $N - \varphi(N)$.

$$\varphi(N) = (p-1)(q-1) = pq - p - q + 1 = N - p - q + 1.$$

Поэтому $N - \varphi(n) \leq p + q - 1$. Поскольку $p + q < 3\sqrt{pq} = 3\sqrt{N}$, то $N - \varphi(N) < 3\sqrt{N}$. Положим

$$ed = 1 + k\varphi(N).$$

Далее оценим сверху разность

$$\begin{aligned} \left| \frac{e}{N} - \frac{k}{d} \right| &= \left| \frac{ed - k\varphi(N) - kN + k\varphi(N)}{Nd} \right| = \\ &= \left| \frac{1 - k(N - \varphi(N))}{Nd} \right| \leq \left| \frac{3k\sqrt{N}}{N} \right| = \frac{3k}{d\sqrt{N}}. \end{aligned}$$

Воспользуемся тем, что $k\varphi(N) = ed - 1$. Следовательно,

$$k\varphi(N) < ed.$$

Примем во внимание, что $e < \varphi(N)$. Окончательно

$$\left| \frac{e}{N} - \frac{k}{d} \right| \leq \frac{1}{d\sqrt[4]{N}} < \frac{1}{2d^2}.$$

Полученное неравенство означает, что величина $\frac{k}{d}$ является подходящей дробью для несекретной дроби $\frac{e}{N}$. Как показано в п. 2.19, всего подходящих дробей не более $\log_2 N$ и все они эффективно вычислимы. \square

Существует предположение, что все секретные экспоненты вплоть до \sqrt{N} – не безопасные.

14.7. ОБ ОДНОМ ОБОБЩЕНИИ RSA-КРИПТОСИСТЕМЫ

Рассмотрим сейчас вопрос о том, как можно путем модификации RSA-криптосистемы увеличить ее стойкость. Речь пойдет о том, как это сделать путем увеличения некоторых параметров [65]. К числу таких параметров относится $\varphi(n)$, а также средняя величина периода сообщения.

Обозначим через \mathbb{Z}_N^* мультиликативную группу кольца \mathbb{Z}_N . Это как раз та группа, в которой действуют RSA-преобразования. В кольце $\mathbb{Z}[x]$ выберем некоторый полином

$$p(x) = x^m + a_1x^{m-1} + \dots + a_{m-1}x + a_m.$$

Обозначим через $\mathbb{Z}_{N,p}$ фактор-кольцо $\mathbb{Z}_N[x]/p(x)$.

Как обычно, будем обозначать через $N = pq$ RSA-модуль. Многочлен $p(x)$ будем называть специальным, если он неприводим по модулям p и q . В этом случае порядок группы $\mathbb{Z}_{N,p}^*$, который естественно обозначить через $\varphi(N, p)$, находится по формуле

$$\varphi(N, p) = (p^m - 1) \cdot (q^m - 1).$$

Это является несложным обобщением известного свойства функции Эйлера.

В общем случае нахождение $\varphi(N, p)$ является трудной задачей. Под общим случаем понимается случай, когда $p(x)$ приводим в полях \mathbb{F}_p и \mathbb{F}_q , и тогда $\varphi(N, p)$ скорее всего зависит от коэффициентов $p(x)$.

Выберем случайно $e \in \{2, 3, \dots, \varphi(N, p) - 2\}$ так, чтобы e было взаимно просто с $\varphi(N, p)$, а затем d так, чтобы

$$e \cdot d \equiv 1 \pmod{\varphi(N, p)}.$$

Заметим, что любой элемент в $\mathbb{Z}_{N,p}$ можно единственным образом представить в виде

$$y = b_1x^{m-1} + b_2x^{m-2} + \dots + b_m,$$

где $b_i \in \mathbb{Z}_N$. Важно отметить, что $y^e \pmod{p}$ можно вычислить стандартным алгоритмом Powmag в Maple. Таким образом, прямое и обратное преобразования $E(y) = y^e, D(y) = y^d$ осуществляются по известному (одному и тому же) алгоритму.

Для любого $y \in \mathbb{Z}_{N,p}$ существует наименьшее натуральное β такое, что

$$y^{e^\beta} = y \in \mathbb{Z}_{N,p}. \quad (14.3)$$

Число β называется периодом Симмонса y в кольце $\mathbb{Z}_{N,p}$ относительно e . Это как раз и есть та величина, порядок которой напрямую влияет на стойкость RSA. Имеется в виду прежде всего атака путем повторного шифрования.

В связи с этим заметим, что \mathbb{Z}_N^* является подгруппой в $\mathbb{Z}_{N,p}$, состоящей из элементов вида

$$y = b_1x^{m-1} + b_2x^{m-2} + \dots + b_m,$$

где $b_1 = b_3 = \dots = b_{m-1} = 0$. Число элементов в фактор-группе $\mathbb{Z}_{N,p}^*/\mathbb{Z}_N^*$ равно

$$\begin{aligned} & (p^m - 1)(q^m - 1) / (p - 1)(q - 1) = \\ & = (p^{m-1} + p^{m-2} + \dots + 1)(q^{m-1} + q^{m-2} + \dots + 1). \end{aligned}$$

В практическом случае при $m=2$ получаем фактор-группу порядка $(p+1)(q+1)$.

В случае RSA без модификации атака Симмонса предотвращается путем специального подбора p и q . Этого можно добиться, если $p-1$ и $q-1$ имеют большие простые делители. Теперь видно, что если $y \in \mathbb{Z}_{N,p}^*/\mathbb{Z}_N^*$, то период Симмонса будет, как правило, больше. Для обеспечения этого у нас появляются дополнительные возможности. Надо лишь найти большие простые делители у чисел $p^m + \dots + 1$ и $q^m + \dots + 1$.

На практике для увеличения стойкости RSA числа p , q и e должны удовлетворять дополнительным требованиям. Частично о них шла речь в начале главы. Теперь, когда есть возможность варьировать m , можно, сохраняя p и q , за счет выбора m удовлетворить эти дополнительные требования.

Можно также поступить следующим образом. Если взять M , кратное $p^m - 1$ и $q^m - 1$, так, чтобы выполнялось условие $ed \equiv 1 \pmod{M}$, то произойдет увеличение ключей, при этом система будет работать корректно.

Перейдем к обоснованию этих предложений. Для выбора ключей надо, чтобы $ed \equiv 1 \pmod{M}$. Это значит, что e и d – обратимые элементы кольца. Поэтому надо сформулировать и доказать некоторое условие обратимости элемента конечного кольца.

Проще всего это сделать в терминах линейной алгебры и теории модулей.

Пусть, например, A – коммутативное кольцо с единицей. $A[x]$ – кольцо многочленов над A . Для любого $p = x^m + a_1x^{m-1} + \dots + a_m \in A[x]$ любой элемент фактор-кольца $A[x]/p$ однозначно представим в виде $y_1x^{m-1} + y_2x^{m-2} + \dots + y_m$, т. е. A/p является свободным модулем с базисом $1, x, \dots, x^{m-1}$.

Определим отображение:

$$M_b : A_p \rightarrow A_p \quad (14.4)$$

по формуле $M_b(y) = by$, где by – это просто произведение в кольце A_p . Это отображение является линейным преобразованием A_p как A -модуля. Его можно представить в матричной форме:

$$\begin{pmatrix} y'_1 \\ \vdots \\ y'_m \end{pmatrix} = \begin{pmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & \ddots & \vdots \\ b_{m1} & \cdots & b_{mm} \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}, \quad (14.5)$$

где $b_{ij} \in A$. Матрицу этого преобразования будем также обозначать через M_b . Условие обратимости можно сформулировать в виде следующей теоремы.

Теорема 14.5. Элемент $b \in A_p$ будет обратим тогда и только тогда, когда обратима матрица M_b , т. е. $|M_b| \in A^*$.

Опустим стандартное доказательство этой теоремы. Покажем на простейшем примере, что это дает. Пусть, например, $p = x^2 + a_1x + a_2$, $b = b_1x + b_2$, тогда $M_b = b_2^2 - a_1b_1b_2 + a_2b_1^2$.

Возвращаясь к кольцу \mathbb{Z} , введем следующие обозначения:

$$\Delta_p = \begin{cases} 0, & \frac{(N+1)^2}{4} a_1^2 \equiv a_2 \pmod{p} \\ \frac{1}{p} \left(\frac{(N+1)^2}{4} a_1 - a_2 \right) & \text{в остальных случаях;} \end{cases} \quad (14.6)$$

$$\Delta_q = \begin{cases} 0, & \frac{(N+1)}{4} a_1^2 \equiv a_2 \pmod{q} \\ \frac{1}{q} \left(\frac{(N+1)}{4} a_1^2 - a_2 \right) & \text{в остальных случаях.} \end{cases} \quad (14.7)$$

Тогда можно показать, что

$$\varphi(n, p) = \begin{cases} (p^2 - 1)(q^2 - 1) & \Delta_p = \Delta_q = -1, \\ (p - 1)(q - 1)(pq - p - q + 3) & \Delta_p = \Delta_q = 1, \\ (p - 1)(q - 1)(pq + p - q + 1) & \Delta_p = 1, \Delta_q = -1, \\ (p - 1)(q - 1)(pq - p + q + 1) & \Delta_p = -1, \Delta_q = 1, \\ (p - 1)(q - 1)(pq - p + 3) & \Delta_p = 0, \Delta_q = 1, \\ (p - 1)(q - 1)(pq - q + 1) & \Delta_p = 0, \Delta_q = -1, \\ (p - 1)(q - 1)(pq - q + 3) & \Delta_p = 1, \Delta_q = 0, \\ (p - 1)(q - 1)(pq + q + 1) & \Delta_p = -1, \Delta_q = 0, \\ (p - 1)(q - 1)(pq + 2) & \Delta_p = 0, \Delta_q = 0. \end{cases} \quad (14.8)$$

Отметим еще один важный случай. Пусть $N=pq$, p, q – нечетные простые, $p(x) = x^m + a_1x^{m-1} + \dots + a_m$ – неприводимый полином над полями \mathbb{F}_p и \mathbb{F}_q . Тогда отображение $Z_{N,p} \rightarrow \mathbb{F}_{p^m} \times \mathbb{F}_{q^m}$ по формуле

$$b = b_1x^m + b_2x^{m-1} + \dots + b_m \rightarrow (b \pmod p, b \pmod q) \quad (14.9)$$

индуцирует групповой изоморфизм

$$\mathbb{Z}_{N,p}^* \rightarrow \mathbb{F}_{p^m}^* \times \mathbb{F}_{q^m}^*, \quad (14.10)$$

и, следовательно,

$$\varphi(N, p) = (p^m - 1)(q^m - 1). \quad (14.11)$$

Теперь корректность системы в целом легко получается из следующих фактов. Во-первых, при $y \neq 0$ можно считать $y(\pmod p) = 0 \in \mathbb{F}_{p^m}$. Тогда $y(\pmod q) \neq 0$ как элемент \mathbb{F}_{q^m} . Поскольку число элементов в \mathbb{F}_{q^m} равно $q^m - 1$, то

$$y^{q^m-1} \equiv 1 \pmod p \text{ в } \mathbb{F}_q.$$

Пусть $ed = 1 + kM$, $k \in \mathbb{Z}$, $M' = M/(q^m - 1)$. Тогда

$$y^{k(q^m-1)M'} \equiv 1 \pmod p \text{ в } \mathbb{F}_q.$$

Во-вторых, рассматривая $y = y_1x^{m-1} + \dots + y_m \in \mathbb{Z}[x]$, $y^{km} \in \mathbb{Z}[x]$, перепишем его в виде $z_1x^{m-1} + \dots + z_m + Q(x)P(x)$. Тогда

$$z_1x^{m-1} + \dots + z_m = 1 = 1 + q(u_1x^{m-1} + \dots + u_m).$$

Согласно предположению, $y = p(v_1x^{m-1} + \dots + v_m)$ для некоторых $v_i \in \mathbb{Z}$, $i = \overline{1, m}$. Это значит,

$$\begin{aligned} y^{kM+1} &= y + pq(u_1x^{m-1} + \dots + u_m)(v_1x^{m-1} + \dots + v_m) + \\ &\quad + p(v_1x^{m-1} + \dots + v_m)Q(x)P(x). \end{aligned}$$

Итак, $y^{kM+1} = y^{ed} = y$ в $\mathbb{Z}_{N,p}$.

14.8. РЮКЗАЧНЫЙ МЕТОД ШИФРОВАНИЯ

Выберем пару w, m натуральных взаимно простых чисел и будем считать их секретными. Число w назовем *множителем*, а m – *модулем*. Дополнительно выберем секретную последовательность $b = (b_1, \dots, b_n)$ положительных целых с условием

$$b_i > \sum_{j=1}^{i-1} b_j, \quad \forall i \geq 1, \quad m > \sum_{j=1}^n b_j.$$

Такую последовательность будем называть *быстрорастущей*.

Последовательность $a = (a_1, \dots, a_n)$, где $a_i \equiv wb_i \pmod{m}$, $i \geq 1$, считается несекретной.

Сообщение $x = (x_1, \dots, x_n)$, являющееся набором нулей и единиц, шифруется по правилу

$$E(x) = \sum_{i=1}^n x_i a_i. \quad (14.12)$$

Для дешифрования полученного сообщения достаточно решить уравнение вида

$$S = \sum_{i=1}^n u_i a_i. \quad (14.13)$$

Уравнение (14.13) основано на задаче о рюкзаке, которая относится к классу NP -полных задач. Тем не менее следующая теорема указывает эффективный метод ее решения легальным пользователем для быстрорастущих последовательностей.

Теорема 14.6. Пусть b_1, \dots, b_n – быстрорастущая последовательность натуральных чисел и $S > 0$. Тогда уравнение $S = \sum_{i=1}^n x_i b_i$ имеет не более одного решения $x = (x_1, \dots, x_n) \in \{0, 1\}^n$ с условием $S \leq \sum_{i=1}^n b_i$.

Доказательство. Любое решение уравнения $S = \sum_{i=1}^n x_i b_i$ должно удовлетворять условию

$$x_i = 1 \Leftrightarrow S \geq b_i + \sum_{j=i+1}^n x_j b_j, \quad i = 1, \dots, n.$$

Необходимость условия очевидна, поскольку все b_i – положительны. Пусть $x_i = 0$. Тогда

$$S = \sum_{j=1}^{i-1} x_j b_j + \sum_{j=i+1}^n x_j b_j < b_i + \sum_{j=i+1}^n x_j b_j.$$

Остается показать, что функция дешифрования эффективно вычислимa. При получении зашифрованного сообщения S пользователь вычисляет w^{-1} по модулю m и решает задачу о рюкзаке:

$$w^{-1}S = \sum_{i=1}^n x_i b_i.$$

Поскольку последовательность b_1, \dots, b_n – быстрорастущая, это легко сделать с помощью теоремы 14.6.

14.9. СТОЙКОСТЬ РЮКЗАЧНОГО ШИФРА

Пусть $w, m, b = (b_1, \dots, b_n), a = (a_1, \dots, a_n)$ – параметры рюкзачной криптосистемы. Криптоаналитик знает последовательность $\{a_i\}$, но не знает w, m, b . Он может попытаться, и небезуспешно, найти пару чисел \bar{w}, \bar{m} таких, что последовательность $\bar{a} = (\bar{a}_1, \dots, \bar{a}_n)$, определяемая условием

$$\bar{a}_i \equiv a_i \bar{w} \pmod{m}, \quad (14.14)$$

является быстрорастущей и обладает свойством

$$\sum_{i=1}^n \bar{a}_i < \bar{m}. \quad (14.15)$$

Из предыдущего пункта следует, что пару \bar{w}, \bar{m} можно использовать для дешифрования с тем же успехом, что и секретную пару w, m .

Деля формулы (14.14) и (14.15) на \bar{m} , получаем

$$\frac{\bar{a}_i}{\bar{m}} \equiv \left(a_i \frac{\bar{w}}{\bar{m}} \right) \pmod{1}, \quad (14.16)$$

$$\sum_{i=1}^n a_i \bar{r}_i \pmod{1} < 1, \quad (14.17)$$

где $\bar{r} = \bar{w}/\bar{m}$. График функции $a_i \bar{r}_i \pmod{1}$ представлен на рис. 14.1. Эту функцию принято называть «косозубой».

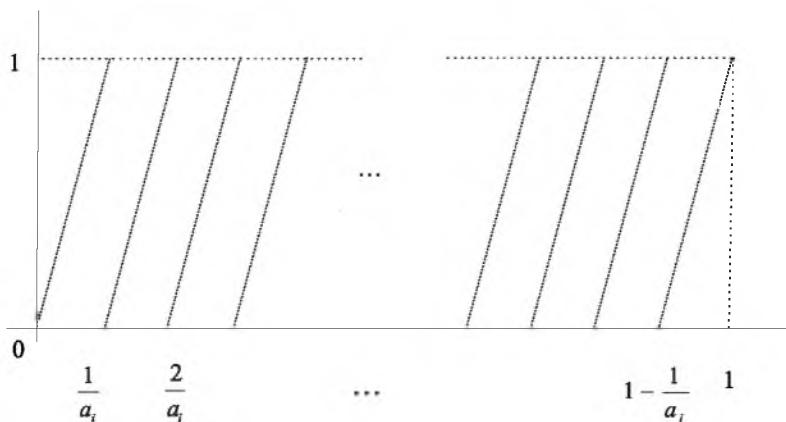


Рис. 14.1. График «косозубой» функции

Для вычисления пары \bar{w}, \bar{m} сначала надо определить точку \bar{r}_0 на оси \bar{r} , чтобы выполнялось условие (14.17). Следовательно, существует и интервал $[r_1, r_2]$, для любой точки которого неравенство (14.17) выполнено.

Пусть p_i – p_i -й минимум i -й «косозубой» функции. Имеем две системы неравенств с целыми неизвестными p_1, \dots, p_n :

$$\begin{aligned} 1 &\leq p_1 \leq a_1 - 1, & -\varepsilon_2 &\leq p_1/a_1 - p_2/a_2 \leq \varepsilon'_2, \\ 1 &\leq p_2 \leq a_2 - 1, & -\varepsilon_3 &\leq p_1/a_1 - p_3/a_3 \leq \varepsilon'_3, \\ &\vdots & &\vdots \\ 1 &\leq p_n \leq a_n - 1, & -\varepsilon_n &\leq p_1/a_1 - p_n/a_n \leq \varepsilon'_n. \end{aligned} \tag{14.18}$$

Параметры $\varepsilon_i, \varepsilon'_i$ нужно выбирать достаточно малыми, чтобы определить точку сгущения. Эту систему можно решить методом целочисленного линейного программирования.

Пусть r – одна из величин, определенная указанной процедурой, и $\bar{r}_1, \dots, \bar{r}_k$ – точки разрыва всех «косозубых» функций такие, что

$$\bar{r}_1, \dots, \bar{r}_k \in \left[\frac{p_1}{a_1}, \frac{p_1 + 1}{a_1} \right), \tag{14.19}$$

размещенные в убывающем порядке. Между двумя такими точками каждая «косозубая» функция задается отрезком прямой

$$\bar{r}a_i - q_i^t, \quad \bar{r}_t \leq \bar{r} < \bar{r}_{t+1}, \tag{14.20}$$

где q_i^t равно номеру минимума i -й «косозубой» кривой, лежащей в интервале $0, \bar{r}_t$. Таким образом, для каждого $1 \leq t \leq k$ условия (14.19) и (14.20) можно сформулировать в виде следующей системы линейных неравенств с неизвестным \bar{r} таким, что $\bar{r}_t \leq \bar{r} < \bar{r}_{t+1}$:

$$\sum_{i=1}^n (\bar{r}a_i - q_i^t) < 1, \tag{14.21}$$

$$(\bar{r}a_i - q_i^t) > \sum_{j=1}^{l-1} (\bar{r}a_j - g_j^t), \quad i = 1, \dots, n. \tag{14.22}$$

Решение последней системы дает подынтервал $[\bar{r}_t, \bar{r}_{t+1})$. Любое $\bar{r} = \bar{w}/\bar{m}$, лежащее в этом подынтервале, дает исковую пару \bar{w}, \bar{m} .

14.10. КРИПТОСИСТЕМА ЭЛЬ-ГАМАЛЯ

Криптосистема Эль-Гамаля основана на сложности вычисления дискретного логарифма в конечном поле. Подобно RSA, она также используется и для цифровой подписи. В этом случае ее называют схемой Эль-Гамаля.

Прежде всего генерируются достаточно большое простое число p и первообразный корень g по модулю p . Далее нам будет удобнее использовать вместо модулярной терминологии язык теории полей. В этом случае g – обра- зующий элемент мультиликативной группы \mathbb{F}_p^* . Выберем далее случайным образом $x \in \mathbb{F}_p^*$ и вычислим $y = g^x$.

1. Открытым ключом является тройка (p, g, y) .

2. Закрытым ключом является элемент x .

Зашифрование сообщения m , $m < p$ состоит в следующем.

1. Генерируется случайным образом одноразовый ключ k , $1 < k < p - 1$.

2. Вычисляются $a = g^k$, $b = y^k m$

3. Пара (a, b) является зашифрованным текстом.

Если $m \geq p$, то сообщение m разбивается на блоки длиной меньше p .

Расшифрование пары (a, b) , зная ключ x , можно произвести по формуле

$$m = b(a^x)^{-1}.$$

В самом деле,

$$b(a^x)^{-1} = y^k m g^{-kx} = m g^{kx} g^{-kx} = m,$$

поскольку $b = g^k m$, $a = g^k$, $y = g^x$.

Очевидным преимуществом криптосистемы Эль-Гамаля является ее вероятностный характер, что обусловлено случайным выбором ключа k . Недостатком является удвоение длины зашифрованного текста по сравнению с начальным текстом. Кроме того, для зашифрования различных сообщений необходимо использовать разные ключи. Можно показать, что если сообщения m_1 и m_2 зашифрованы с помощью одного и того же ключа k , то для соответствующих шифртекстов (a_1, b_1) и (a_2, b_2) выполняется соотношение $b_1 b_2^{-1} = m_1 m_2^{-1}$. Следовательно, если известно m_1 , то легко находится m_2 .

14.11. КРИПТОСИСТЕМА МАК-ЭЛИСА

Алгоритм шифрования, предложенный Р. Мак-Элисом, основан на сложности декодирования линейных кодов. Под бинарным линейным (n, k) -кодом понимается линейное отображение

$$Y = XG,$$

где $X \in \mathbb{F}_2^k$, $Y \in \mathbb{F}_2^n$, $G - k \times n$ – матрица над полем \mathbb{F}_2 . Иногда кодом называют образ этого отображения, т. е. множество всех кодовых слов Y . Когда первые k столбцов матрицы G образуют единичную матрицу E_k , код называется систематическим. В этом случае кодовое слово состоит из k информационных символов и $n - k$ контрольных. Если минимальное расстояние Хэмминга $d(x, y)$ между кодовыми словами равно $2t + 1$, то такой код исправляет до t ошибок.

В системе Мак-Элиса используются линейные коды, обладающие эффективным алгоритмом декодирования. К числу таких кодов относятся коды Гоппа [29].

Генерация ключей заключается в следующем.

1. Берется (n, k) линейный код, исправляющий t ошибок с порождающей $k \times n$ -матрицей G .
2. Случайным образом строится невырожденная $k \times k$ -матрица S и матрица перестановки P порядка n , т. е. матрица, полученная из E_n перестановкой строк.
3. Открытым ключом является пара $\bar{G} = SGP$, t , секретным – тройка матриц S, G, P .

Опишем процесс зашифрования.

1. Обладатель открытого ключа представляет свое сообщение m в виде бинарных блоков длиной k .
2. Шифрование одного блока выполняется по формуле $c = m\bar{G} + z$, где z – случайный бинарный вектор длины n , веса $\leq t$, т. е. в нем не более t единиц.

Зашифрованное сообщение обрабатывается следующим образом.

1. Обладатель секретного ключа (S, G, P) вычисляет $cP^{-1} = m\bar{G}P^{-1} + zP^{-1} = mSG + zP^{-1}$.
2. Используя алгоритм декодирования, можно найти mS , а затем и $m = (mS)S^{-1}$.

Для обоснования корректности расшифрования достаточно заметить, что вес слова zP^{-1} не превышает t . Поэтому алгоритм декодирования находит mSG .

Система Мак-Элиса является первой криптосистемой, использовавшей рандомизацию, т. е. добавление случайного вектора z в процессе шифрования.

14.12. КРИПТОСИСТЕМА БЛЮМА – ГОЛЬДВАССЕР

Еще одной вероятностной криптосистемой является криптосистема Блюма – Гольдвассер. Подобно RSA, она основана на сложности факторизации больших целых чисел и считается самой эффективной среди вероятностных схем шифрования с открытым ключом.

Криптосистему Блюма – Гольдвассер называют вероятностной потому, что для генерации ключа выбирается случайным образом элемент $x_0 \in \mathbb{Z}_N^*$, где $N = p q$ – RSA-модуль. Затем строится последовательность квадратов в кольце \mathbb{Z}_N^* : $x_0, x_0^2, x_0^4, \dots, x_0^{2^t}$. Последовательность p_0, p_1, \dots, p_t младших бит этих квадратов используется затем в режиме гаммирования для шифрования сообщения. Описанный способ выработки гаммы называют BBS-генератором псевдослучайных чисел.

Рассмотрим подробно способ генерации ключей.

1. Сначала вычисляют $N = p q$, где p, q – случайные большие простые числа вида $p \equiv q \equiv 3(\text{mod } 4)$.

2. Затем, используя расширенный алгоритм Евклида, находят u и v такие, что $pu + qv = 1$.

3. N считается открытым ключом, а (p, q, u, v) – секретным ключом.

Зашифрование происходит следующим образом.

1. Сообщение m представляется в виде бинарной последовательности $m = m_1 m_2 \dots m_t$.

2. Случайным образом берется x_0 квадратичный вычет по модулю N . Это можно сделать случайно, выбирая s , $(s, n) = 1$, а затем вычисляя $x_0 = s^2(\text{mod } N)$.

3. Далее последовательно вычисляются элементы x_1, x_2, \dots, x_{t+1} , где $x_i = x_{i-1}^2(\text{mod } N)$, а затем вычисленные элементы кроме последнего заменяются их младшими битами p_1, p_2, \dots, p_t .

4. Зашифрованным сообщением считается $c = (c_1, c_2, \dots, c_t), x_{t+1}, c_i = p_i \oplus m_i, i = 1, 2, \dots, t$.

Чтобы восстановить исходное сообщение, последовательно находят

$$d_1 = \left(\frac{p+1}{4} \right)^{t+1} \text{mod } (p-1);$$

$$d_2 = \left(\frac{q+1}{4} \right)^{t+1} \text{mod } (q-1);$$

$$k = x_{t+1}^{d_1} \text{mod } p;$$

$$l = x_{t+1}^{d_2} \text{mod } q;$$

$$x_0 = (lpu + kqv) (\text{mod } N).$$

Используя найденное, остается лишь применить шаг 3 алгоритма зашифрования, чтобы восстановить сообщение m :

$$m_i = p_i \oplus p_i \oplus m_i, \quad i = 1, 2, \dots, t.$$

Для доказательства корректности работы алгоритма надо всего лишь доказать, что формула $x_0 = (lpu + kqv) (\text{mod } N)$ правильно восстанавливает значение x_0 . Попутно будет объяснено, почему необходимо условие $p \equiv q \equiv 3(\text{mod } 4)$.

Теорема 14.7. *Формула $x_0 = (lpu + kqv) (\text{mod } N)$ при условии $p \equiv q \equiv 3(\text{mod } 4)$ правильно восстанавливает значение x_0 .*

Доказательство. Поскольку $p \equiv 3(\text{mod } 4)$, то число $\frac{(p+1)}{4}$ будет целым.

Поэтому

$$x_{t+1}^{(p+1)/4} \equiv (x_t^2)^{(p+1)/4} \equiv x_t^{(p+1)/2} \equiv x_t^{(p-1)/2} x_t \pmod{p}.$$

Применяя критерий Эйлера и пользуясь тем, что x_t является квадратом по модулю $N = p q$, а значит и квадратом по модулю p , имеем

$$x_{t+1}^{(p+1)/4} \equiv x_t \pmod{p}.$$

Аналогично

$$x_t^{(p+1)/4} \equiv x_{t-1} \pmod{p}.$$

Повторение этого влечет

$$k \equiv x_{t+1}^{d_1} \equiv \left(x_{t+1}^{(p+1)/4}\right)^{t+1} \equiv x_0 \pmod{p}.$$

Аналогично

$$l \equiv x_{t+1}^{d_2} \equiv x_0 \pmod{p}.$$

Окончательно, так как $ru + qv = 1$, то $lpu + kqv \equiv x_0 \pmod{p}$ и $lpu + kqv \equiv x_0 \pmod{q}$. Следовательно, $lpu + kqv \equiv x_0 \pmod{N}$, т. е. значение x_0 данной формулой восстановлено правильно.

14.13. ЗАДАНИЯ

1. Найти количество решений сравнения $x^m \equiv 1 \pmod{N}$.
2. Сколько решений имеет сравнение $x^m \equiv -1 \pmod{N}$ в случае его разрешимости?
3. Пусть E, D – взаимно обратные преобразования RSA-криптосистемы. Тогда выполняется $D(E(x)) = x$ для любого $x \in \mathbb{Z}_N^*$. Показать, что это свойство справедливо при любом x .
4. Доказать, что всевозможные преобразования RSA-криптосистемы образуют группу относительно их композиции.
5. Зашифровано сообщение по правилу $y \equiv x^k \pmod{p}$, где p – большое простое число, $1 \leq x \leq p-1$, k – целое число, $1 < k < p-1$. Показать, что если k выбрано взаимно простым с $p-1$, то алгоритм расшифрования

$$d(y) = y^d \pmod{p}$$

является корректным с $d \equiv k^{-1} \pmod{p-1}$ и $d(y) = x$.

6. Что случится с криптосистемой в предыдущей задаче, если ошибочно взять целое число k , не взаимно простое с $p-1$?
7. Показать, что в схемах RSA и Рабина шифрование открытого текста длиной n битов в зашифрованный текст длиной N битов требует $O(n^3)$ операций.

8. Предположим, что пользователь RSA в качестве модуля N по ошибке выбрал большое простое число. Показать, что в этом случае расшифровать текст легко.

9. Заданы различные простые p, q и $N = pq$. Показать, что если при заданных взаимно простых d и N сравнение

$$x^2 \equiv d \pmod{N}$$

имеет хотя бы одно решение, то оно имеет четыре решения.

10. Рассмотрим RSA-систему с модулем N . Целое число x , $1 \leq x \leq N - 1$, назовем неподвижной точкой, если оно и в зашифрованном виде тоже x . Показать, что если x – неподвижная точка, то и $N - x$ также есть неподвижная точка.

11. Показать, что в схеме RSA с параметрами p, q, e, d имеется $r + s + rs$ неподвижных точек M , $1 \leq M \leq N - 1$, где

$$r = (p - 1, e - 1), \quad s = (q - 1, e - 1).$$

Задача показывает, что в схемах шифрования с большим количеством неподвижных точек уже заложен недостаток, поэтому желательно выбирать такие p и q , для которых r и s малы.

12. Предложите способ решения сравнения $x^2 \equiv 1 \pmod{N}$. Найдите четыре решения сравнения $x^2 \equiv 1 \pmod{17 \cdot 19}$.

13. Выбрав параметры e и d RSA-криптосистемы и полагая $N = 19 \cdot 23$, примените алгоритм из доказательства теоремы 14.2 для факторизации модуля.

14. Докажите формулу $\varphi(n, p) = (p^m - 1)(q^m - 1)$ из п. 14.7 в случае, когда многочлен неприводим по модулям p и q .

15. Постройте криптосистему Мак-Элиса, используя бинарный код 00000000, 00111111, 11001011, 11110100, исправляющий две ошибки.

16. Постройте криптосистему Блюма – Гольдвассер, полагая $p = 19, q = 23$.

Г л а в а 15

ЭЛЕКТРОННАЯ ЦИФРОВАЯ ПОДПИСЬ

Имеются следующие две тесно взаимосвязанные проблемы:

- 1) сохранение информации от незаконного использования;
- 2) подтверждение авторства (принадлежность информации конкретному лицу).

Первая проблема решается с помощью шифрования. Она рассмотрена в предыдущих главах.

Проблема подтверждения авторства тесно связана с первой и возникает при следующих обстоятельствах.

1. Некоторый абонент A получает сообщение X предположительно от B . Как подтвердить, что оно получено именно от него, а не от какого-либо третьего лица (не сфальсифицировано противником)?

2. Абонент A получает от B какое-то сообщение X . Как подтвердить, что оно не было изменено кем-либо посторонним?

При обмене информацией в компьютерных сетях для подтверждения авторства были разработаны алгоритмы *электронной цифровой подписи* (ЭЦП). В основе большинства из них лежит идея криптографии с открытым ключом.

15.1. ОБОВЩЕННАЯ МОДЕЛЬ ЭЦП

В практической деятельности важно не только защищать информацию от злоумышленника, но и иметь возможность проверить авторство данного сообщения и тот факт, что оно не было изменено посторонним лицом. Именно для решения этих проблем (автентификация и целостность) разработан ряд алгоритмов ЭЦП. В основе большинства из них лежит идея использования односторонней функции с секретом.

Суть этой идеи состоит в использовании некоторой односторонней функции с секретом F_S для создания пары (x, y) , где x – сообщение; а y – решение уравнения $F_S(y) = x$.

Всякая информация, записанная в некотором алфавите, может быть представлена в виде двоичных слов, т. е. конечных последовательностей из нулей и единиц. Количество двоичных цифр в таком слове будем называть его *длиной*. Пусть X и Y – некоторые подмножества множества всех двоичных слов.

Односторонней функцией с секретом S называется функция $F_S : Y \rightarrow X$, зависящая от параметра S и обладающая следующими тремя свойствами:

- 1) при любом S существует полиномиальный алгоритм вычисления значений $F_S(y)$;
- 2) при неизвестном S не существует полиномиального алгоритма для решения уравнения $F_S(y) = x$ относительно y ;
- 3) при известном S существует полиномиальный алгоритм для решения уравнения $F_S(y) = x$ относительно y .

До настоящего времени не известно ни одного примера односторонней функции с секретом, но для практических целей используют некоторые функции, которые могут оказаться односторонними. Для них второе свойство строго не доказано, но известно, что задача инвертирования эквивалентна некоторой трудно решаемой математической задаче.

Пусть A и B – некоторые пользователи, обменивающиеся информацией по открытому каналу связи. Пусть X – совокупность всевозможных сообщений, Y – некоторое множество подписей. Пусть $F_k : Y \rightarrow X$ – функция, зависящая от параметра $k \in K$, называемого *ключом*. Будем считать, что ключ k состоит из двух частей – k_S и k_O , где k_S – секретная составляющая, известная только A ; и k_O – открытая составляющая, известная всем (не держится в секрете). Пусть F_k является сюръекцией, т. е. для любого $x \in X$ существует прообраз $y = F_k^{-1}(x)$. Функцию F также считаем общеизвестной.

Предположим, что выполняются следующие свойства:

- 1) зная k_O , функцию $F_k(y)$ можно вычислить по алгоритму полиномиальной сложности;
- 2) зная k_S , функцию $F_k(y)$ можно инвертировать по алгоритму полиномиальной сложности;
- 3) зная k_O , но не зная k_S , функцию $F_k(y)$ сложно инвертировать, т. е. неизвестен или не существует полиномиальный алгоритм нахождения $F_k^{-1}(x)$.

Прообраз $y = F_k^{-1}(x)$ некоторого сообщения x называется *подписью этого сообщения*. Пара (x, y) называется *подписаным сообщением*.

В силу первого свойства всегда легко проверить, соответствует ли подпись сообщению, а в силу третьего подделать подпись при достаточно большом ключе практически невозможно. Доказательство этого свойства позволило бы придать подписаным сообщениям юридическую силу.

Секретный и открытый ключи находятся во взаимно однозначном соответствии, и (в силу третьего требования) нет полиномиального алгоритма вычисления секретной компоненты по открытой. В общем виде алгоритм ЭЦП выглядит так.

1. Для передаваемого сообщения x отправитель A находит

$$y = F_k^{-1}(x).$$

Знание секретного ключа k_S позволяет ему сделать это за приемлемое время.

2. Далее A передает B по какому-либо каналу связи пару (x, y) , где x – сообщение; y – подпись.

3. Получив подписанное сообщение (x, y) , B находит $x' = F_k(y)$. Знание открытого ключа k_O позволяет сделать это за приемлемое время.

4. Получатель B сверяет x и x' . Если они совпадают, то полученное сообщение считаем подлинным. В противном случае либо сообщение x изменено (фальшивое), либо подпись y неверная (поддельная).

Замечание 15.1. Предложенную модель можно дополнить предварительным шифрованием пересылаемого сообщения и итоговой расшифровкой.

Замечание 15.2. Роль функции F_k иногда играет некоторая схема шифрования с открытым ключом. В силу этого многие вопросы (стойкость, выбор ключей и др.) для схем ЭЦП и соответствующих криптосистем равносильны.

В последующих пунктах этой главы излагается ряд схем ЭЦП, использующихся на практике [33].

15.2. СХЕМА ЭЦП РАБИНА

Стойкость схемы Рабина основана на трудности решения квадратичных сравнений по большому составному модулю. Как было показано в предыдущей главе, данная задача эквивалентна факторизации.

Вот одна из реализаций этой схемы.

Сначала выберем два простых числа p и q , оба сравнимые с 3 по модулю 4. Это простые секретные ключи, а их произведение $N = pq$ – открытый ключ.

Для шифрования сообщения m (m должно быть меньше, чем N) вычисляем

$$c = m^2 \pmod{N}.$$

Расшифровать сообщения можно следующим образом. Поскольку получатель знает p и q , он может решить два сравнения, используя китайскую теорему об остатках. Вычислим

$$\begin{aligned} m_1 &\equiv c^{\frac{p+1}{4}} \pmod{p}; \\ m_2 &\equiv \left(p - c^{\frac{p+1}{4}}\right) \pmod{p}; \\ m_3 &\equiv c^{\frac{q+1}{4}} \pmod{q}; \\ m_4 &\equiv \left(q - c^{\frac{q+1}{4}}\right) \pmod{q}. \end{aligned}$$

Затем выберем целое $a \equiv q(q^{-1} \pmod{p})$, целое $b = p(p^{-1} \pmod{q})$ и найдем четыре возможных решения сравнений:

$$\begin{aligned} M_1 &\equiv (am_1 + bm_3) \pmod{N}; \\ M_2 &\equiv (am_2 + bm_4) \pmod{N}; \\ M_3 &\equiv (am_2 + bm_3) \pmod{N}; \\ M_4 &\equiv (am_1 + bm_4) \pmod{N}. \end{aligned}$$

Один из четырех результатов M_1, M_2, M_3 или M_4 равен m . Если сообщение является английским текстом, то правильное сообщение M_j выбрать легко. С другой стороны, если сообщение – случайная последовательность цифр, то нет возможности определить корректное сообщение M_j . Одно из решений этой проблемы – добавить к сообщению известный текст перед шифрованием.

Покажем, как схема Рабина может быть использована для аутентификации.

В качестве функции шифрования для сообщения m возьмем функцию $m^2 \pmod{N}$, где N – открытый ключ A . Если мы хотим быть уверенными, что это сообщение послано именно A , то случайно выберем большое целое число l и пошлем A сообщение $r = l^2 \pmod{N}$. A расшифровывает его, используя знание факторизации N на простые числа p и q , находит квадратный корень l_1 из числа r и возвращает его нам. Мы поверим, что связаны именно с A тогда и только тогда, когда вернувшееся сообщение l_1 удовлетворяет сравнению $l_1^2 \equiv r \pmod{N}$.

15.3. СХЕМА ЭЦП ЭЛЬ-ГАМАЛЯ

Пусть p – большое простое число, q – большой простой делитель $p - 1$, $g \in (\mathbb{Z}/p\mathbb{Z})^*$ имеет порядок q . Пользователь A выбирает секретный ключ x ($1 < x < q$) и находит $y = g^x \pmod{p}$ – открытый ключ. Предположим, что он отправляет сообщение m ($1 \leq m \leq q$). Подписью к этому сообщению будет пара (r, s) , удовлетворяющая

$$g^m \equiv y^r \times r^s \pmod{p} \quad (15.1)$$

с условиями $0 < r < p$ и $0 < s < q$. Она строится следующим образом. Выбираем случайное $k \in [1, q - 1]$ и вычисляем

$$r = g^k \pmod{p}, \quad s = k^{-1}(m - xr) \pmod{q}.$$

Действительно, так как порядок g равен q , полученные r и s удовлетворяют (15.1):

$$y^r \times r^s \equiv g^{xr} \times g^{k \cdot k^{-1}(m - xr)} \equiv g^{xr + m - xr} \equiv g^m \pmod{p}.$$

Покажем, что при каждом k существует единственное $s < q$, удовлетворяющее (15.1). Из (15.1) имеем

$$g^m \equiv g^{xr} \times g^{ks} \pmod{p} \Rightarrow g^{m - xr - ks} \equiv 1 \pmod{p} \Rightarrow$$

$$\Rightarrow m - xr - ks \equiv 0 \pmod{q} \Rightarrow m - xr \equiv ks \pmod{q} \Rightarrow$$

$$\Rightarrow s \equiv k^{-1}(m - xr) \pmod{q}.$$

Если выбрано k , то для выработки подписи (без знания секретного ключа x) необходимо найти такое $s < q$, что верно (15.1) или

$$r^s \equiv b \pmod{p}, \quad (15.2)$$

где $b = g^{-r} \pmod{p}$. Поскольку числа k и q взаимно простые, то число $r = g^k$ имеет тот же порядок, что и g , т. е. q . Получается, что для взлома системы таким способом необходимо решить задачу дискретного логарифмирования в подгруппе G , образованной g и имеющей большой порядок q .

15.4. ЭЦП DSS

National Institute of Standards and Technology (NIST) (Национальный институт стандартов и технологий) в 1991 г. предложил для обсуждения проект стандарта ЭЦП DSS (Digital Signature Standard), использующий алгоритм DSA (Digital Signature Algorithm). Стойкость данного алгоритма основана на сложности решения задачи дискретного логарифмирования в мультиплексивной группе простого поля F_p (см. гл. 3).

Выработка подписи

Выработка подписи осуществляется по следующему алгоритму.

1. Отправитель A сообщения M предоставляет широкому кругу абонентов (получателей его сообщений) доступ к следующим параметрам:

- p – простое число, $2^{512} < p < 2^{1024}$, битовая длина p кратна 64;
- q – простое число, $2^{159} < q < 2^{160}$, и делитель $p - 1$;
- $g = h^{\frac{p-1}{q}} \pmod{p}$, где h – такое целое число, что $0 < h < p$ и $h^{\frac{p-1}{q}} \times (\pmod{p}) > 1$;
- y – открытый ключ, сформированный по правилу $y = a^x \pmod{p}$. Здесь x – секретный ключ, известный только A , причем $0 < x < q$;
- $H(M)$ – хэш-функция, которая по исходному сообщению M формирует целое число в диапазоне от 1 до q (см. п. 13.1).

2. Пользователь A генерирует случайное число k такое, что $0 < k < q$, держит его в секрете и уничтожает сразу после получения подписи.

3. A находит два числа – r и s по следующему правилу:

$$\begin{aligned} r &= (g^k \pmod{p}) \pmod{q}; \\ s &= k^{-1} (xr + H(M)) \pmod{q}. \end{aligned}$$

Подписью к сообщению M является пара (r, s) .

Проверка подписи

Пользователь B получает от A сообщение M' и подпись (r', s') к нему. B должен убедиться, что M совпадает с M' . Для этого:

- 1) если хотя бы одно из условий $0 < s' < q$, $0 < r' < q$ не выполняется, то подпись считается недействительной;
- 2) B находит $v = (s')^{-1} \pmod{q}$;
- 3) B вычисляет $z_1 = H(M')v \pmod{q}$, $z_2 = r'v \pmod{q}$;
- 4) далее вычисляется $u = (g^{z_1}y^{z_2} \pmod{p}) \pmod{q}$;
- 5) B проверяет условие $r' = u$. Если оно выполняется, то подпись считается подлинной, а сообщение – не измененным, т. е. $M' = M$.

Корректность DSA

Пусть $M = M'$, $s = s'$, $r = r'$. Покажем, что тогда $u = r$.

Итак, $v = s^{-1} \pmod{q}$, $z_1 = H(M)v \pmod{q}$, $z_2 = rv \pmod{q}$. Имеем

$$\begin{aligned} u &= g^{z_1}y^{z_2} \pmod{p} \pmod{q} = g^{H(M)s^{-1}}g^{xrs^{-1}} \pmod{p} \pmod{q} = \\ &= g^{k(xr+H(M))^{-1}(xr+H(M))} \pmod{p} \pmod{q} = g^k \pmod{p} \pmod{q} = r. \end{aligned}$$

Таким образом, $u = r$, и корректность алгоритма доказана.

Для нахождения секретных параметров ЭЦП по открытым параметрам необходимо решить следующую систему сравнений:

$$\begin{cases} y \equiv a^x \pmod{p}, \\ g^k + pn \equiv r' \pmod{p}, \\ s' \equiv k^{-1}(xr + H(M')) \pmod{p}, \end{cases}$$

где неизвестными являются x , n , k .

В работе [146] имеются некоторые замечания по стойкости алгоритма DSA.

1. В алгоритме выработки подписи есть недостаток: в редких случаях, когда $s = 0$, при проверке подписи будет сбой, поскольку в этом случае не существует s^{-1} . Эта ошибка легко устраняется при помощи дополнительной проверки, что и сделано в российском стандарте ЭЦП.

2. Алгоритм DSA медленный. В то время как скорость получения подписи сравнима со скоростью шифрования по схеме RSA, проверка подписи в большом количестве случаев примерно в 100 раз медленнее, чем RSA.

3. Тот факт, что один модуль p используется многими пользователями, ослабляет стойкость алгоритма, поскольку единственный взлом p нарушает безопасность сразу всех абонентов, пользующихся этим p . Под *взломом* понимается некое предвычисление, которое позволяет в дальнейшем легко решать проблему дискретного логарифмирования для данного p .

4. Величина 512 битов для p слишком мала. С учетом тенденции уменьшения стоимости вычислений стоимость взлома через несколько лет может сократиться до разумной величины, что для стандарта неприемлемо.

5. Существует целый класс простых чисел, для которых проблема дискретного логарифмирования решается легко. Причем построить такие числа также легко, однако затраты на проверку, является ли данное простое «слабым», превышают возможности среднего пользователя. Это значит, что тот, кто распределяет простые p , в принципе может знать секретные ключи своих клиентов.

6. Анализ алгоритма DSA показывает, что в данном случае проблема взлома подписи, вообще говоря, не сводится к проблеме дискретного логарифмирования, поскольку в алгоритме DSA g – не первообразный корень по модулю p , а лишь элемент порядка q , что намного меньше $p - 1$. Таким образом, вполне возможно, что проблема взлома алгоритма ЭЦП легче общей проблемы дискретного логарифмирования.

15.5. ЭЦП ГОСТ Р 34.10-94

Задаются следующие параметры, используемые алгоритмом: p – простое число, $2^{509} < p < 2^{512}$ либо $2^{1020} < p < 2^{1024}$; q – простое число, $2^{254} < q < 2^{256}$, такое, что $q \mid p - 1$; a – целое число в пределах $1 < a < p - 1$ такое, что $a^q \bmod p = 1$; x – целое число в пределах $0 < x < q$; y – целое число, равное $a^x \bmod p$; M – целое число в пределах $0 < M < q$.

Число x называют *секретным ключом пользователя*, y – *открытым ключом пользователя*, M – *сообщением*. В соответствии с алгоритмом проверки подписи в ГОСТ Р 34.10-94 электронную подпись можно ввести следующим образом.

Пусть дано сообщение M . Подписью к M называется пара целых чисел (r, s) таких, что

$$1 < r < q, \quad 0 < s < q, \quad (15.3)$$

$$r = \left(a^{sM^{q-2} \bmod q} \times y^{-rM^{q-2} \bmod q} \bmod p \right) \bmod q. \quad (15.4)$$

Теорема 15.1. Пусть задано целое M ($0 < M < q$). Тогда существует ровно q различных решений $(r_k, s_k)_{k=0}^{q-1}$ уравнения (15.4), причем

$$r_k = (a^k \bmod p) \bmod q, \quad (15.5)$$

$$s_k = (xr_k + kM) \bmod q. \quad (15.6)$$

Доказательство. Легко проверить путем подстановки, что все пары (r_k, s_k) , заданные по формулам (15.5) и (15.6), удовлетворяют равенству (15.4). Все эти пары различны. Действительно, если $(r_i, s_i) = (r_j, s_j)$ при некоторых $i \neq j$, то, используя (15.6), получаем $xr_i + iM \equiv xr_j + jM \pmod{q}$, откуда следует $(i - j)M \equiv 0 \pmod{q}$, что вместе с $(M, q) = 1$ дает $i \equiv j \pmod{q}$. Последнее сравнение возможно, только если $i = j$. Таким образом, чтобы доказать утверждение, достаточно показать, что сравнение (15.4) имеет не более q решений.

Будем рассматривать циклическую подгруппу группы \mathbb{Z}_p^* :

$$\langle a \rangle_p = \{a^k \pmod{p} : k = 0, \dots, q-1\},$$

порожденную элементом a . Для $m \in \{0, \dots, q-1\}$ определим функцию $\Psi(m)$ как число различных элементов группы $\langle a \rangle_p$, которые по модулю q дают остаток m . Очевидно равенство $\sum_{m=0}^{q-1} \Psi(m) = q$. Пусть Φ – количество решений (15.4), $\Phi(r)$ – количество различных s ($0 \leq s < q$) таких, что пара (r, s) есть решение (15.4). Тогда $\Phi = \sum_{r=0}^{q-1} \Phi(r)$.

Зафиксируем r . Поскольку $y = a^x \pmod{p}$, то имеем равенство

$$a^{sM^{q-2} \pmod{q}} \times y^{-rM^{q-2} \pmod{q}} \pmod{p} = a^{(s-rx)M^{q-2} \pmod{q}} \pmod{p}.$$

Когда s пробегает значения от 0 до $q-1$, то $v = (s - rx)M^{q-2} \pmod{q}$ также пробегает (в другом порядке) эти значения. Тогда $a^v \pmod{p}$ пробегает все элементы группы $\langle a \rangle_p$, когда s изменяется от 0 до $q-1$. Количество различных s таких, что

$$r = \left(a^{sM^{q-2} \pmod{q}} \times y^{-rM^{q-2} \pmod{q}} \pmod{p} \right) \pmod{q},$$

есть $\Phi(r)$ с одной стороны и $\Psi(r)$ – с другой. Тогда

$$\Phi = \sum_{r=0}^{q-1} \Phi(r) = \sum_{r=0}^{q-1} \Psi(r) = q.$$

Итак, количество различных решений сравнения (15.4) равно q , и все эти решения описаны по формулам (15.5) и (15.6). \square

Введем множества

$$\begin{aligned} R(M) &= \{(r_k, s_k) : r_k = (a^k \pmod{p}) \pmod{q}; \\ &s_k = (xr_k + kM) \pmod{q}, \quad 0 \leq k \leq q-1\}; \\ S(M) &= \{(r, s) \in R(M) : (r-1)rs \neq 0\}. \end{aligned}$$

Тогда множество $R(M)$ состоит из всех решений уравнения (15.4), а $S(M)$ – из всех подписей к M .

Чтобы сформировать подпись к сообщению M , необходимо решить сравнение (15.4) относительно (r, s) . Такую пару можно вычислить по формулам (15.5) и (15.6). Но для этого необходимо знать значение секретного ключа x и параметра k . Таким образом, получаем следующие алгоритмы.

Алгоритм генерации подписи

1. Генерируем случайное k в интервале $[0, q - 1]$ (его значение держится в секрете).
2. Находим r_k и s_k по формулам (15.5) и (15.6) (их вычисление осуществляется по алгоритмам полиномиальной сложности).
3. Значение числа k уничтожается.
4. Если $r_k \times s_k \neq 0$, то полученная пара (r_k, s_k) является подписью к M . В противном случае переходим к шагу 1.

Алгоритм проверки подписи

1. Если $r < 0$, или $r > q - 1$, или $s < 0$, или $s > q - 1$, то подпись недействительная, иначе – переходим к следующему шагу.
2. Вычисляем $z_0 = M^{q-2} \bmod q$.
3. Вычисляем $z_1 = sz_0 \bmod q$.
4. Вычисляем $z_2 = -rz_0 \bmod q$.
5. Вычисляем $z_3 = a^{z_1} \bmod p$.
6. Вычисляем $z_4 = y^{z_2} \bmod p$.
7. Вычисляем $z_5 = z_3z_4 \bmod p$.
8. Вычисляем $z_6 = z_5 \bmod q$.
9. Если $r \neq z_6$, то подпись недействительная, иначе – подпись действительная.

Эквивалентное преобразование схемы ЭЦП

Рассмотрим уравнение

$$\bar{r} = \left(a^{\bar{s}M^{q-2} \bmod q} \times y^{-\bar{r}M^{q-2} \bmod q} \right) \bmod p \quad (15.7)$$

в целых $0 < \bar{r} < p$, $0 \leq \bar{s} < q$.

Теорема 15.2. Пусть задано целое M ($0 < M < q$). Тогда существует ровно q различных решений $(\bar{r}_k, \bar{s}_k)_{k=0}^{q-1}$ уравнения (15.7), причем

$$\bar{r}_k = (a^k \bmod p); \quad (15.8)$$

$$\bar{s}_k = (x\bar{r}_k + kM) \bmod q. \quad (15.9)$$

Доказательство. Из уравнения (15.7) очевидно, что \bar{r} имеет вид (15.8) с некоторым целым k ($0 \leq k < q$), поскольку $y = a^x \bmod p$. Тогда имеем

$$a^k \equiv a^{\bar{s}M^{q-2} - \bar{r}xM^{q-2}} \pmod{p},$$

что выполнено тогда и только тогда, когда $k \equiv \bar{s}M^{q-2} - \bar{r}xM^{q-2} \pmod{q}$, откуда получаем (15.9). Итак, все решения уравнения (15.7) определяются по формулам (15.8) и (15.9). Все значения r_k различны и, следовательно, все решения различны. \square

Определим $\bar{R}(M)$ – множество решений (15.7) и

$$\bar{S}(M) = \{(\bar{r}, \bar{s}) \in \bar{R}(M) : (r-1)rs \not\equiv 0 \pmod{q}\}.$$

Теорема 15.3. Пусть задано целое M ($0 < M < q$). Тогда для всех k от 0 до $q-1$ справедливы следующие утверждения:

- 1) $r_k = \bar{r}_k \bmod q$;
- 2) $s_k = \bar{s}_k$;
- 3) $\bar{r}_k = (a^{s_k M^{q-2} \bmod q} \times y^{-r_k M^{q-2} \bmod q}) \bmod p$;
- 4) $(r_k, s_k) \in S(M) \iff (\bar{r}_k, \bar{s}_k) \in \bar{S}(M)$.

Доказательство. Первое равенство очевидно в силу определения r_k . Тогда имеем

$$\begin{aligned} s_k &= (xr_k + kM) \bmod q = (x(\bar{r}_k \bmod q) + kM) \bmod q = \\ &= (x\bar{r}_k + kM) \bmod q = \bar{s}_k. \end{aligned}$$

Из того, что y имеет порядок q и $\bar{r}_k \equiv r_k \pmod{q}$, вытекает $y^{-\bar{r}_k M^{q-2} \bmod q} = y^{-r_k M^{q-2} \bmod q}$. Подставляя вместо выражения из левой части выражение из правой части и s_k вместо \bar{s}_k в (15.7), получаем третью формулу. Четвертое утверждение вытекает из первых двух. \square

Теорема 15.4. Пусть задано целое M ($0 < M < q$). Определим отображение $F_M : \bar{R}(M) \rightarrow R(M)$ по правилу $F_M(\bar{r}, \bar{s}) = (\bar{r} \bmod q, \bar{s})$. Тогда:

- 1) F_M устанавливает взаимно однозначное соответствие между $\bar{R}(M)$ и $R(M)$;
- 2) обратное отображение $(\bar{r}, \bar{s}) = F_M^{-1}(r, s)$ имеет вид

$$\bar{s} = s, \quad \bar{r} = (a^{sM^{q-2} \bmod q} \times y^{-rM^{q-2} \bmod q}) \bmod p;$$

- 3) сужение F_M на $\bar{S}(M)$ устанавливает взаимно однозначное соответствие между $\bar{S}(M)$ и $S(M)$.

Это утверждение вытекает из предыдущего.

Вывод: в силу теоремы 15.3 задача нахождения решений уравнения (15.7) эквивалентна задаче нахождения решений уравнения (15.4) в том смысле, что по решению одной из этих задач (по алгоритмам полиномиальной сложности) легко строится решение другой.

Дискретное логарифмирование в схеме ЭЦП

Достаточным условием для формирования подписи к произвольным сообщениям является знание x , для нахождения которого достаточно решить задачу дискретного логарифмирования.

Выясним, к каким проблемам приводит задача построения подписи. В силу доказанной эквивалентности задач решения уравнений (15.4) и (15.7) будем исследовать задачу решения уравнения (15.7), которое можно переписать в виде

$$r^M y^r \equiv a^s \pmod{p}. \quad (15.10)$$

Итак, чтобы сформировать подпись, необходимо решить уравнение (15.10). Множество решений этого уравнения есть однопараметрическое семейство пар. Если зафиксировать параметр r , то приходим к уравнению

$$a^s \equiv b \pmod{p},$$

где $b = r^M y^r \pmod{p}$; s – неизвестное. Его решение – задача дискретного логарифмирования.

Зададим s_0 и предположим, что уравнение (15.10) имеет решение (r, s) такое, что $s = s_0$. Тогда, чтобы найти r , необходимо решить уравнение $r^M y^r \equiv a^{s_0} \pmod{p}$ или

$$d^r \equiv br \pmod{p},$$

где $b = a^{-s_0 M^{q-2} \pmod{q}} \pmod{p}$, $d = y^{-M^{q-2} \pmod{q}} \pmod{p}$. Для решения этой задачи алгоритмов, кроме почти полного перебора всех вариантов, на настоящее время не известно. Эту задачу можно интерпретировать иначе: необходимо найти такое r , что $rd_1^r = b_1$, где $d_1 = d^{-1} \pmod{p}$ и $b_1 = b^{-1} \pmod{p}$.

О совпадении ЭЦП для различных сообщений

Теорема 15.5. Пусть даны $M_1, M_2 \in \mathbb{Z}$ такие, что $0 < M_1, M_2 < q$, $M_1 \neq M_2$. Тогда

$$\bar{R}(M_1) \cap \bar{R}(M_2) = \{(1, x)\}; \quad (15.11)$$

$$\bar{S}(M_1) \cap \bar{S}(M_2) = \emptyset. \quad (15.12)$$

Доказательство. Пусть $(r, s) \in \bar{R}(M_1) \cap \bar{R}(M_2)$. Тогда $r = a^k \pmod{p}$ и $xr + kM_1 \equiv xr + kM_2 \pmod{q}$, откуда $k(M_1 - M_2) \equiv 0 \pmod{q}$, что возможно только при $k = 0$. Следовательно, $r = 1$ и $s = (x \times 1 + 0 \times M_1) \pmod{q} = x$. Соотношение (15.12) вытекает из (15.11). \square

Замечание 15.3. Хотя $\bar{S}(M_1)$ и $\bar{S}(M_2)$ не пересекаются при различных M_1 и M_2 , $S(M_1)$ и $S(M_2)$ могут пересекаться. Действительно, если $r_k = r_l$ при некоторых различных k и l , то пара сообщений $M_1 = (s - xr_k)k^{-1} \pmod{q}$ и $M_2 = (s - xr_l)l^{-1} \pmod{q}$, где $s \in \{1, \dots, q-1\}$ – произвольно, будет иметь одинаковую подпись, причем $M_1 \neq M_2$. Это связано с тем, что r_k может принимать одинаковые значения при различных k .

Чтобы оценить вероятность совпадения подписей, для $m \in \{0, \dots, q-1\}$ введем случайную величину η_m , равную количеству различных k ($0 \leq k < q$), для которых $m = (a^k \pmod{p}) \pmod{q}$. Также рассмотрим P_i , равные количеству различных m таких, что $\eta_m = i$. Введем вспомогательную функцию $\chi_i(j) = \delta_{ij}$, $(i, j) \in \{0, \dots, q-1\}^2$. Легко заметить, что

$$\sum_{i=0}^{q-1} \chi_i(j) = 1.$$

Тогда величину η_m можно задать следующим образом:

$$\eta_m = \sum_{k=0}^{q-1} \chi_m(r_k),$$

где $r_k = (a^k \pmod{p}) \pmod{q}$. Найдем среднее значение η :

$$\sum_{m=0}^{q-1} \eta_m = \sum_{m=0}^{q-1} \sum_{k=0}^{q-1} \chi_m(r_k) = \sum_{k=0}^{q-1} \sum_{m=0}^{q-1} \chi_m(r_k) = \sum_{k=0}^{q-1} 1 = q, \quad (15.13)$$

откуда вытекает, что математическое ожидание $E\{\eta\} = 1$.

Далее, P_i/q – вероятность события $\eta = i$, поэтому имеет место

$$\sum_{i=0}^{q-1} P_i = q. \quad (15.14)$$

Поскольку математическое ожидание η равно $\sum_{i=0}^{q-1} iP_i/q = 1$, то

$$\sum_{i=0}^{q-1} iP_i = q, \quad (15.15)$$

откуда $\sum_{i=0}^{q-1} P_i = \sum_{i=0}^{q-1} iP_i$, что эквивалентно

$$P_0 = \sum_{i=2}^{q-1} (i-1)P_i.$$

Пусть w – количество различных значений случайной величины η . В этом случае η принимает значение i тогда и только тогда, когда $P_i \neq 0$. Пусть i_1, \dots, i_w – все различные значения η , причем $i_j < i_{j+1}$. Тогда, используя (15.15), получаем

$$q = \sum_{j=1}^w i_j P_{i_j} \geq \sum_{j=1}^w i_j \geq \sum_{i=0}^{w-1} i = \frac{w(w-1)}{2} > \frac{(w-1)^2}{2}.$$

Следовательно,

$$w \leq \lceil \sqrt{2q} \rceil + 1. \quad (15.16)$$

Пример 15.1. Пусть при $p = 22643563$, $q = 1109$, имеем следующее распределение η_m : $P_0 = 425$, $P_1 = 382$, $P_2 = 202$, $P_3 = 80$, $P_4 = 17$, $P_5 = 3$ и $P_i = 0$ при $i \geq 6$, или:

η	0	1	2	3	4	5	> 5
P	425 1109	382 1109	202 1109	80 1109	17 1109	3 1109	0

Наиболее вероятно естественное предположение о том, что η принимает значения $0, 1, \dots, w-1$. Но в этом случае получаем, что $P_i = 0$ при $i \geq w$. Значит, среди значений r_k одно и то же не может встретиться более w раз. Скорее всего, вероятности P_i должны быстро убывать (как показывает пример). В этом случае на w можно получить гораздо более сильные оценки. Это будет свидетельствовать о том, что одни и те же значения r_k встречаются очень редко и r_k распределена в некотором смысле равномерно.

Замечание 15.4. В качестве подписи к некоторому сообщению M можно было бы принять пару $(\bar{r}, \bar{s}) \in \bar{S}(M)$, чтобы избежать опасности совпадения подписей на разных сообщениях или опасности искажения сообщения с сохранением подписи. Но опыт показывает, что подобные события маловероятны и практически не влияют на стойкость. Кроме того, приведение по двойному модулю обладает неоспоримым преимуществом: построенная таким образом подпись на 30 % короче ее расширенного аналога (\bar{r}, \bar{s}) и тем самым существенно экономит объем пересылаемых информационных сообщений. Чтобы дать строгое математическое обоснование, необходимо изучить распределение значений последовательности $(a^j \bmod q) \bmod p$. В следующем пункте проводится предварительный анализ этого распределения.

15.6. ЭКВИВАЛЕНТНОСТЬ ЗАДАЧ ФАЛЬСИФИКАЦИИ ПОДПИСИ В DSS И СХЕМЕ ЭЛЬ-ГАМАЛЯ

Теорема 15.6. *Существование эффективного алгоритма подписи по схеме DSS эквивалентно существованию эффективного алгоритма подписи по схеме Эль-Гамаля.*

Доказательство. Считаем, что параметры p , q и a в обеих схемах одинаковы. Пусть существует эффективный алгоритм подписи по схеме DSS и задан открытый ключ y_2 для схемы Эль-Гамаля. Положим $y_1 = y_2^{-1} \bmod p$ – открытый ключ для схемы DSS. По гипотетическому алгоритму находим подпись (\tilde{r}, \tilde{s}) заданного сообщения M . Подпись для схемы Эль-Гамаля построим по формулам

$$\begin{cases} s = \tilde{s}, \\ r = a^{Ms^{q-2} \bmod q} \times y_1^{\tilde{r}s^{q-2} \bmod q} \bmod p. \end{cases}$$

Действительно, полученная пара удовлетворяет условиям определения.

Обратно, пусть существует эффективный алгоритм подписи по схеме Эль-Гамаля и задан открытый ключ y_1 для схемы DSS. Положим $y_2 = y_1^{-1} \bmod p$ – открытый ключ для схемы Эль-Гамаля. По гипотетическому алгоритму находим подпись (\tilde{r}, \tilde{s}) заданного сообщения M . Подпись для схемы DSS построим по формулам

$$\begin{cases} s = \tilde{s}, \\ r = \tilde{r} \bmod q. \end{cases}$$

Полученная пара удовлетворяет определению. \square

Замечание 15.5. Теорема 15.6 аналогична третьему пункту теоремы 15.3, который устанавливает эквивалентность схемы цифровой подписи в стандарте РФ и описанной другой схемы, аналогичной схеме Эль-Гамаля.

В силу эквивалентности DSS и схемы Эль-Гамаля достаточно изучить проблему взлома последней. Существует аналогия со стандартом РФ (см. п. 15.5): чтобы найти корректную подпись (r, s) при заданном $r = r_0$, необходимо решить сравнение вида $r^s \equiv b \pmod{p}$, где $b = a^M y^{-r} \bmod p$ и s – неизвестное. Его решение есть задача дискретного логарифмирования. В свою очередь, чтобы корректную подпись (r, s) при заданном $s = s_0 \neq 0$, необходимо решить сравнение вида $rd^r \equiv b \pmod{p}$, где $b = a^{Ms^{q-2} \bmod q} \bmod p$, $d = y^{s^{q-2} \bmod q} \bmod p$.

Замечание 15.6. Одной из возможных задач при взломе какой-либо схемы цифровой подписи может быть задача нахождения какой-нибудь корректной пары (M, S) , где M – сообщение; S – подпись. С этой точки зрения все описанные схемы (российский стандарт, DSS, алгоритм Эль-Гамаля) эквивалентны. Действительно, в случае российского стандарта необходимо и

достаточно решить уравнение (15.10), а в случае DSS и схемы Эль-Гамаля – уравнение (15.1). Но очевидно, что эти уравнения переходят друг в друга путем перестановки переменных t и s .

Замечание 15.7. При оценке стойкости любой криптосистемы встает вопрос о влиянии количества известных корректных пар (M, S) , где M – сообщение; S – подпись, на стойкость криптографического алгоритма. Алгоритмы, построенные на операции возведения в степень в кольце \mathbb{Z}/N (типа RSA), позволяют найти любое количество таких пар. Для этого достаточно знать одну пару и преобразовывать ее, используя мультипликативную структуру операции возведения в степень, или по заданному S вычислять M . Такой прием в используемых схемах, основанных на операции экспонирования, сталкивается с непреодолимой трудностью – необходимо решать задачу дискретного логарифмирования.

15.7. ЭЦП СТБ 1176.2-99

Введенный в 1999 г. стандарт Республики Беларусь СТБ 1176.2-99 «Информационная технология. Защита информации. Процедуры выработки и проверки электронной цифровой подписи» базируется на *схеме ЭЦП Шнорра* [154], которая является модификацией схемы Эль-Гамаля.

Параметры

При выработке и проверке подписи используются l -битовое простое число p и r -битовое простое число q , $q \mid (p - 1)$. Допустимые значения параметров l и r указаны в табл. 15.1. Применяется определяемая СТБ 1176.1-99 функция хэширования h , параметр L которой устанавливается равным $r - 1$.

Таблица 15.1

Допустимые значения параметров r и l

Уровень стойкости	r	l	Уровень стойкости	r	l
1	143	638	6	208	1534
2	154	766	7	222	1790
3	175	1022	8	235	2046
4	182	1118	9	249	2334
5	195	1310	10	257	2462

Часть преобразований стандарта выполняется в группе G , определяемой множеством $B_p = \{1, 2, \dots, p - 1\}$ и операцией \circ : $u \circ v = uvR^{-1} \bmod p$, где $R = 2^{l+2}$. Использование операции \circ вместо обычного умножения по модулю p упрощает применение алгоритма Монтгомери (см. п. 9.2). Далее $u^{(m)}$ – m -я степень числа $u \in B_p$ как элемента G .

В стандарте приведены алгоритмы генерации чисел p, q и элемента $a \in B_p$, имеющего порядок q в группе G . Числа p, q, a являются долговременными параметрами СТБ 1176.2-99, единными для группы пользователей.

Входные данные

Всякое сообщение M , подпись к которому вырабатывается или проверяется, задается последовательностью байтов. Если t – n -разрядное число по основанию $2^8 = 256$, т. е.

$$t = \sum_{i=0}^{n-1} t_i (256)^i, \quad 0 \leq t_i < 256, \quad t_{n-1} \neq 0,$$

то $t \parallel M$ – сообщение, полученное вставкой байтов t_0, t_1, \dots, t_{n-1} в начало M .

Выработка подписи

При выработке подписи S к сообщению M используется личный ключ x , $0 < x < q$. Выполняются следующие шаги.

1. Выработать случайное секретное число k , $1 < k < q$.
2. $t \leftarrow a^{(k)}$.
3. $U \leftarrow h(t \parallel M)$. Если $U = 0$, то вернуться к шагу 1.
4. $V \leftarrow (k - xU) \bmod q$. Если $V = 0$, то вернуться к шагу 1.
5. $S \leftarrow U2^r + V$.

Проверка подписи

При проверке подписи S к сообщению M используется открытый ключ $y = a^{(x)}$. Алгоритм проверки подписи состоит из следующих шагов.

1. $V \leftarrow S \bmod 2^r$.
 2. $U \leftarrow (S - V)/2^r$.
 3. Проверить условия $0 < U < 2^{r-1}$, $0 < V < q$. Если хотя бы одно из условий нарушается, то подпись признается недействительной и выполнение алгоритма завершается.
 4. $t \leftarrow a^{(V)} \circ y^{(U)}$.
 5. $W \leftarrow h(t \parallel M)$.
 6. Если $W \neq U$, то подпись признается недействительной.
- Если $W = U$, то принимаются решения о том, что:

- а) подпись S была создана с помощью личного ключа x , связанного с открытым ключом y ;
- б) подпись S и сообщение M не были изменены с момента их создания.

15.8. ЦИФРОВАЯ ПОДПИСЬ НА ЭЛЛИПТИЧЕСКИХ КРИВЫХ

Криптосистемы на эллиптических кривых предложены в 1985 г. В. Миллером и Н. Коблицем. Основные преимущества, которые позволили говорить о криптографии на эллиптических кривых с практической точки зрения, – это, во-первых, большие возможности выбора группы, в которой производятся вычисления, и, во-вторых, отсутствие субэкспоненциального алгоритма дискретного логарифмирования в группе точек на эллиптической кривой (за исключением некоторых частных случаев). Большинство криптографических алгоритмов, стойкость которых базируется на сложности дискретного логарифмирования в конечном поле, достаточно легко переносятся на случай эллиптических кривых. В данном пункте рассмотрим американский алгоритм цифровой подписи на эллиптических кривых (ECDSA), принятый в качестве международного стандарта. В упомянутом стандарте также используются эллиптические кривые над полем характеристики 2. Однако криптографически стойких кривых над такими полями сравнительно мало. Поэтому ограничимся рассмотрением случая эллиптических кривых, заданных над простым полем большей характеристики.

15.8.1. Начальные параметры алгоритма цифровой подписи

Основными начальными параметрами в рассматриваемом алгоритме являются эллиптическая кривая E , определенная над конечным полем \mathbb{F}_p , характеристики p , и базовая точка $G \in E(\mathbb{F}_p)$, имеющая большой простой порядок в группе точек на данной эллиптической кривой. Эти параметры могут использоваться как совокупностью лиц, так и одним пользователем.

Эллиптическая кривая задается уравнением

$$y^2 = x^3 + ax + b.$$

Таким образом, задание кривой состоит в выборе двух элементов – a , b – из поля \mathbb{F}_p , которые определяют это уравнение. Различные пары параметров (a, b) могут определять изоморфные эллиптические кривые. При выборе эллиптической кривой можно сначала задать j -инвариант этой кривой, а затем по нему построить коэффициенты a и b .

Точка G на эллиптической кривой определяется парой элементов x_G, y_G из \mathbb{F}_p : $G = (x_G, y_G)$. Эта точка выбирается случайно. Один из способов выбора – зафиксировать случайное x и затем найти y как корень второй степени из $x^3 + ax + b$ в поле \mathbb{F}_p , если он существует. Существование корня проверяется путем вычисления символа Лежандра (см. п. 2.15)

$$\left(\frac{x^3 + ax + b}{p} \right).$$

Для получения криптографически стойкой системы цифровой подписи должны выполняться следующие условия:

1) порядок точки G должен быть равен простому числу

$$n > \max\{2^{160}, 4\sqrt{p}\};$$

2) $\#E(\mathbb{F}_p) \neq p + 1$, т. е. кривая не должна быть *суперсингулярной*;

3) $p^k \not\equiv 1 \pmod{n}$ для всех $k \in \{1, \dots, C\}$, где C настолько велико, что вычислить дискретный логарифм в \mathbb{F}_{p^C} за приемлемое время невозможно (обычно берут $C = 20$);

4) $\#E(\mathbb{F}_p) \neq p$, т. е. кривая не должна быть *аномальной*.

Заметим, что условие 3) подразумевает условие 2). Возможный способ защищаться от известных и возможных атак для специальных классов кривых, которые могут быть обнаружены в будущем, – выбирать кривую E случайным образом так, чтобы выполнялись указанные условия.

Выбор эллиптической кривой подразумевает решение ряда трудоемких вспомогательных задач. Прежде всего это подсчет количества точек на эллиптической кривой (об этом речь пойдет в конце главы). После того как порядок N кривой определен, требуется найти большой простой делитель n порядка кривой. Такой делитель может в принципе не существовать, и тогда потребуется повторять процедуру выбора кривой до тех пор, пока не выполнятся все требуемые условия. Поиск числа n может потребовать как разложения на множители числа N , так и доказательства простоты полученного множителя n .

Точку G можно выбрать следующим образом. Найдем случайную точку $G' \in E(\mathbb{F}_q)$ и вычислим $G = [N/n]G'$ (см. (15.17)). Будем повторять эту операцию до тех пор, пока точка G не станет отличной от точки \mathcal{O} .

Описанные параметры могут быть общими для совокупности пользователей. Для генерации и проверки подписи требуются еще и индивидуальные параметры пользователя – так называемые секретный и открытый ключи.

Ключ подписи (секретный ключ) – это случайное число d в интервале $0 < d < n$.

Ключ проверки подписи (открытый ключ) – это точка на эллиптической кривой $Q = [d]G$.

Алгоритм ЭЦП также использует хэш-функцию, которая обозначается h .

15.8.2. Генерация и проверка цифровой подписи

АЛГОРИТМ ГЕНЕРАЦИЯ ПОДПИСИ

Вход: сообщение m , исходные параметры и ключ подписи.

Выход: подпись (r, s) .

1. Выбрать случайное число k в интервале $1 \leq k \leq n - 1$.
 2. Вычислить $(x_1, y_1) := [k]G$.
 3. Вычислить $r := x_1 \bmod n$.
 4. Если $r = 0$, то вернуться к шагу 1.
 5. Вычислить $z := k^{-1} \bmod n$.
 6. Вычислить $e := h(m)$.
 7. Вычислить $s := z(e + dr) \bmod n$.
 8. Если $s = 0$, то вернуться к шагу 1
 9. Вывести пару (r, s) – подпись к m .
-

АЛГОРИТМ ПРОВЕРКА ПОДПИСИ

Вход: сообщение m , исходные параметры, ключ проверки подписи и подпись к m .

Выход: утверждение, что подпись действительная или фальшивая.

1. Если условия $1 \leq r, s \leq n - 1$ нарушаются, то вывести «подпись фальшивая» и завершить работу алгоритма.
 2. Вычислить $e := h(m)$.
 3. Вычислить $w := s^{-1} \bmod n$.
 4. Вычислить $u_1 := ew \bmod n$.
 5. Вычислить $u_2 := rw \bmod n$.
 6. Вычислить $X := [u_1]G + [u_2]Q = (x_1, y_1)$.
 7. Если $r = x_1 \bmod n$, то вывести «подпись действительная», иначе – «подпись фальшивая» и завершить работу алгоритма.
-

Доказательство (Корректность алгоритма генерации подписи). Докажем, что любая подпись, сгенерированная по алгоритму генерации подписи, будет «действительной» согласно алгоритму проверки подписи.

Прежде всего заметим, что параметры r и s не превосходят $n - 1$ как остатки при делении на n целых чисел. С другой стороны, выполняется проверка

того, что $r, s \neq 0$ на шагах 4 и 8 алгоритма генерации подписи. Следовательно, условия шага 1 алгоритма проверки подписи будут выполнены всякий раз, когда r, s получены по алгоритму генерации подписи.

Далее согласно шагам 5 и 7 алгоритма генерации подписи имеем

$$ks \equiv e + dr \pmod{n}.$$

Поскольку $w = s^{-1} \pmod{n}$ (шаг 3 алгоритма проверки подписи), то $k \equiv we + wrd \pmod{n}$. Поскольку точка G имеет порядок n , то

$$\begin{aligned}[k]G &= [we + wrd]G = [we]G + [wr][d]G = \\ &= [we]G + [wr]Q = [u_1]G + [u_2]Q = X.\end{aligned}$$

Таким образом, точка X , получаемая на шаге 6 алгоритма проверки подписи, совпадет с точкой $[k]G$, сгенерированной при получении подписи по алгоритму генерации. Первая координата X будет равна x_1 , и ее остаток \pmod{n} будет равен r (согласно шагу 3 алгоритма генерации подписи). Корректность доказана. \square

15.9. ОСОБЕННОСТИ СКАЛЯРНОГО УМНОЖЕНИЯ НА ЭЛЛИПТИЧЕСКИХ КРИВЫХ

Под скалярным умножением точки P некоторой эллиптической кривой на целое число $m > 0$ понимается вычисление точки

$$[m]P = \underbrace{P + \cdots + P}_m. \quad (15.17)$$

При $m < 0$ полагают $[m]P = -([-m]P)$, а при $m = 0 - [0]P = \mathcal{O}$.

Эта операция – аналог возведения в степень в мультипликативных группах. Умение эффективно выполнять скалярное умножение весьма важно, поскольку именно эта операция наиболее трудоемкая во многих криптографических алгоритмах, в том числе и в алгоритме цифровой подписи, приведенном выше.

В группе точек на эллиптической кривой скалярное умножение можно осуществить *бинарным методом*. Рассмотрим разложение m в двоичной системе счисления:

$$m = a_0 + a_1 2 + a_2 2^2 + \dots + a_{k-1} 2^{k-1}, \quad a_i \in \{0, 1\},$$

где $k = \lceil \log_2 m \rceil + 1$. Тогда для нахождения значения $[m]P$ (P – точка на эллиптической кривой) достаточно вычислить $[2^i]P$ для $i = 0, \dots, k-1$ и просуммировать лишь те $[2^i]P$, для которых $a_i = 1$. Всего нам потребуется $k-1$ раз удвоить точку, начиная с P . Затем сделаем не более $k-1$ сложений (это число достигается, когда $m = 2^k - 1$).

Однако специфика эллиптических кривых позволяет уменьшить количество сложений в бинарном методе. Идея метода состоит в том, чтобы блок из последовательных единиц в двоичном представлении m заменять на разность степеней двойки. Например, вместо вычисления $[2^i]P + [2^{i+1}]P + \dots + [2^{i+j}]P$ можно вычислить $[2^{i+j+1}]P - [2^i]P$, сэкономив $(j - 1)$ сложений, поскольку операция нахождения обратной точки вследствие специфики эллиптических кривых практически не требует времени. Еще одно улучшение касается случая, когда два блока из последовательных единиц разделены нулем:

$$m = \underbrace{11\dots 1}_a 0 \underbrace{11\dots 1}_b \dots$$

Тогда достаточно двух вычитаний:

$$[m]P = [2^{a+b+1}]P - [1]P - [2^b]P.$$

15.10. КРИПТОАНАЛИЗ АЛГОРИТМОВ ЭЦП, ОСНОВАННЫХ НА ФАКТОРИЗАЦИИ И ДИСКРЕТНОМ ЛОГАРИФМИРОВАНИИ

15.10.1. Задача факторизации

Задача факторизации состоит в разложении заданного натурального числа n в произведение $p_1^{\alpha_1} \dots p_k^{\alpha_k}$, $\alpha_i \geq 1$, степеней различных простых чисел p_1, \dots, p_k . Очевидно, такое разложение можно получить, последовательно применяя алгоритм нахождения собственного делителя заданного натурального числа.

Рассмотрим метод факторизации Ферма. Идея состоит в попытке найти представление $n = x^2 - y^2$, которое сразу дает разложение $n = (x+y)(x-y)$. Найти такое разложение можно по следующему алгоритму. Поскольку больший из множителей n больше \sqrt{n} , начинаем с $x = \lceil \sqrt{n} \rceil$, где $\lceil a \rceil$ обозначает наименьшее целое, большее либо равное a . Полагаем $z = x^2 - n$.

1. Проверяем, является ли z квадратом натурального числа. Если ответ положительный, то $y = \sqrt{z}$, и мы получаем требуемое представление. В противном случае переходим к следующему шагу.

2. Находим следующие значения для z и x :

$$z + 2x + 1 = x^2 + 2x + 1 - n = (x+1)^2 - n$$

и переходим к шагу 1.

Оценим количество циклов, необходимых для достижения результата. Если $n = ab$ – произведение двух простых (худший случай) и $a < b$, то алгоритм

прекращает работу, когда $x = (a + b)/2$. Поскольку мы начинали с $x \approx \sqrt{n}$ и $b = n/a$, то число циклов

$$W \approx \frac{1}{2} \left(a + \frac{n}{a} \right) - \sqrt{n} = \frac{(\sqrt{n} - a)^2}{2a}.$$

Если предположить, что $a = \delta\sqrt{n}$, $0 < \delta < 1$, то получаем

$$W \approx \frac{(1 - \delta)^2}{2\delta} \sqrt{n}.$$

Еще одна оценка количества циклов возникает из того, что алгоритм прекращает работу, когда y достигает $(b - a)/2$. Начальное значение y находится в пределах от 0 до $2\sqrt{n} + 1$. Поэтому

$$\frac{b - a}{2} - 2\sqrt{n} - 1 < W < \frac{b - a}{2}.$$

Это означает, что для обеспечения стойкости RSA необходимо, чтобы разность $b - a$ была велика.

Будем говорить, что число m является *B-гладким*, если все простые делители m не превосходят $B \in \mathbb{N}$. Множество S_B , составленное из всех простых чисел, не больших B , назовем *базой множителей*.

Пусть число $p - 1$ является *B-гладким* для некоторого (но не для каждого) простого делителя p нечетного числа n . *Метод p - 1*, предложенный Поллардом [141], позволяет найти собственный делитель d числа n , затратив $O(B \ln n / \ln B)$ модулярных умножений. Пусть

$$Q = \prod_{q \in S_B} q^{\alpha_q},$$

где α_q – минимальное целое такое, что

$$q^{\alpha_q} \geq \sqrt{n} > \min_{p|n} (p - 1).$$

Вычислим $2^Q \pmod{n}$ и определим $d = (2^Q - 1, n)$. Имеем:

- 1) $p | n$;
- 2) $(p - 1) | Q$;
- 3) по малой теореме Ферма $2^Q \equiv 2^{p-1} \equiv 1 \pmod{p}$ и $p | (2^Q - 1)$.

Таким образом, $p | d$ и число d является делителем n .

Метод эллиптических кривых [121] является обобщением метода $p - 1$. Для факторизации n требуется в среднем $L_{1/2, 1}(p)$ операций, где p – наименьший простой делитель n , а

$$L_{\varepsilon, c}(x) = O\left(\exp\left((c + o(1))(\ln x)^\varepsilon (\ln \ln x)^{1-\varepsilon}\right)\right), \quad 0 < \varepsilon < 1, \quad c > 0.$$

Квадратичные методы факторизации направлены на поиск решения $(x, y) \in \mathbb{Z}_n \times \mathbb{Z}_n$ сравнения

$$x^2 \equiv y^2 \pmod{n}.$$

Если $x \not\equiv \pm y \pmod{n}$, то n делит $(x - y)(x + y)$, но не делит ни $(x - y)$, ни $(x + y)$. Таким образом, $d = (x - y, n)$ является собственным делителем n .

Для решения сравнения поступают следующим образом.

1. Выбирают базу множителей $S_B = \{p_1, \dots, p_s\}$.
2. Находят T пар целых чисел (a_t, b_t) таких, что $a_t^2 \equiv b_t \pmod{n}$ и число $|b_t|$ является B -гладким, т. е.

$$b_t = (-1)^{\alpha_{t0}} p_1^{\alpha_{t1}} \cdots p_s^{\alpha_{ts}}.$$

3. Составляется $T \times (s + 1)$ матрица $A = (\alpha_{ti})$. Если сумма некоторых строк A является вектором с четными координатами $2(\beta_0, \beta_1, \dots, \beta_s)$ и \mathcal{T} – множество номеров таких строк, то искомое сравнение найдено:

$$\left(\prod_{i \in \mathcal{T}} a_i \right)^2 \equiv \left(\prod_{i=1}^s p_i^{\beta_i} \right)^2 \pmod{n}.$$

4. Если найденное сравнение не позволяет факторизовать n , то находят новые пары (a_t, b_t) , новое множество \mathcal{T} и повторяют вычисления.

Основную вычислительную трудность при таком подходе составляет нахождение подходящих пар (a_t, b_t) на этапе 2. *Метод квадратичного решета* (quadratic sieve, QS, [142]) направлен на нахождение пар вида $(x + m, Q(x))$, где $m = \lfloor \sqrt{n} \rfloor$; $Q(x) = (x + m)^2 - n$, $x = 0, \pm 1, \pm 2, \dots$. Поиск подходящих пар проводится следующим образом:

- 1) определяется массив R , индексируемый числами $x = 0, \pm 1, \pm 2, \dots, \pm M$, и заполняется значениями $R[x] \leftarrow \log |Q(x)|$;
- 2) для всякой степени $p^\alpha \leq C$ элемента базы множителей p находятся решения сравнения $Q(x) \equiv 0 \pmod{p^\alpha}$ относительно $x \in \{0, 1, \dots, p^\alpha - 1\}$;
- 3) если x^* – одно из найденных решений, то $R[x] \leftarrow R[x] - \log p$ для всех x таких, что $x \equiv x^* \pmod{p^\alpha}$;
- 4) если после обработки степеней всех чисел из базы множителей (*просеивания*) $R[x]$ равно 0, то число $|Q(x)|$ является B -гладким (доказать, почему) и $(x + m, Q(x))$ – искомая пара.

При практической реализации метода квадратичного решета значения логарифмов вычисляются приближенно. После просеивания для всех x таких, что $R[x] \approx 0$, проверяется B -гладкость числа $Q(x)$.

Доказано, что оптимальным в методе квадратичного решета является использование $s \approx L_{1/2, 1/2}(n)$ элементов базы множителей. При этом сложность алгоритма факторизации составляет $L_{1/2, 1}(n)$ операций в среднем.

Для сравнения: наиболее эффективный на сегодняшний день *обобщенный метод решета числового поля* (General Number Field Sieve, GNFS) имеет сложность $L_{1/3, c}(n)$, где $c = \left(\frac{64}{9}\right)^{1/3} \approx 1,923$.

В табл. 15.2 приведены результаты факторизации некоторых чисел RSA- k .

Таблица 15.2

Рекорды факторизации

Число	Дата	Сложность, MIPS-лет*	Алгоритм
RSA-100	Апрель 1991	7	QS
RSA-110	Апрель 1992	75	QS
RSA-120	Июнь 1993	830	QS
RSA-129	Апрель 1994	5000	QS
RSA-130	Апрель 1996	500	GNFS
RSA-140	Февраль 1999	2000	GNFS
RSA-155	Август 1999	8000	GNFS

* MIPS – миллион инструкций в секунду.

Каждое такое число n состоит из k десятичных разрядов и является произведением двух близких простых p и q , которые генерировались с использованием вероятностных тестов на простоту и уничтожались сразу после вычисления pq . Последнее достижение (на момент подготовки данного издания) – факторизация числа RSA-768, потребовавшая:

- полгода работы 80 процессоров для выполнения начальной стадии выбора многочленов для общего алгоритма решета числового поля (GNFS);
- два года работы нескольких сот компьютеров для завершения второй стадии алгоритма GNFS – стадии просеивания. На одном ядре процессора AMD Opteron с частотой 2,2 ГГц эта работа заняла бы около 1500 лет;
- нескольких суток выполнения финальной стадии на кластере из 37 одноядерных узлов, работающих на процессорах семейства Core 2 с частотой 2,26 ГГц и 16 Гб оперативной памяти.

15.10.2. Задача дискретного логарифмирования

Пусть G – конечная мультиплективная группа. Задача дискретного логарифмирования состоит в решении при заданных $a, b \in G$ уравнения $a^x = b$ относительно целого x , $0 \leq x < n = \text{ord } a$. Решение x называется *дискретным логарифмом* или *индексом* b по основанию a и обозначается $\text{ind}_a b$.

Пусть $m = \lceil \sqrt{n} \rceil$. Метод больших-малых шагов состоит в нахождении совпадения элемента последовательности $1, a, a^2, \dots, a^{m-1}$ с элементом последовательности $b, ba^{-m}, ba^{-2m}, \dots, ba^{-(m-1)m}$. Для вычисления последовательностей требуется $O(m + \log_2 m)$ умножений на элементы G . Если найдено совпадение $a^j = ba^{-im}$, то $a^{im+j} = b$ и $\text{ind}_a b = (im + j) \bmod n$.

Пусть $n = qr$, q – простое, $r > 1$. Метод Нечаева – Полягина – Хеллмана [34, 140] решения уравнения $a^x = b$ состоит в следующем.

1. По методу больших-малых шагов находится решение x_1 , $0 \leq x_1 < q = \text{ord } a^r$, уравнения

$$(a^r)^{x_1} = b^r.$$

2. Находится решение x_2 , $0 \leq x_2 < r = \text{ord } a^q$, уравнения

$$(a^q)^{x_2} = ba^{-x_1}.$$

3. Определяется решение $x = x_2q + x_1$.

Если r – простое, то уравнение на шаге 2 решается по методу больших-малых шагов. При составном r уравнение снова заменяется двумя уравнениями и т. д. Если известно разложение $n = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ порядка элемента a на простые множители, то для вычисления $\text{ind}_a b$ требуется

$$O\left(\log n + \alpha_i \sum_{i=1}^k \sqrt{p_i}\right)$$

операций умножения на элементы G .

Для нахождения дискретного логарифма в поле \mathbb{F}_q , q – простое число, по индекс-методу [61] поступают следующим образом.

1. Выбирают базу простых множителей $S_B = \{p_1, \dots, p_s\}$.
2. Находят T пар натуральных чисел (b_t, m_t) таких, что $1 \leq m_t \leq q - 2$, $b_t = a^{m_t} \pmod q$ является B -гладким, т. е. $b_t = p_1^{\alpha_{t1}} \dots p_s^{\alpha_{ts}}$.
3. Составляется система из T уравнений

$$\alpha_{t1}x_1 + \dots + \alpha_{ts}x_s \equiv m_t \pmod{(q-1)}, \quad t = 1, \dots, T,$$

относительно неизвестных $x_i = \text{ind}_a p_i$, $i = 1, \dots, s$. На шаге 2 находят такое количество пар T , что решение полученной системы определено однозначно.

4. Находят пару натуральных чисел (m', b') таких, что $1 \leq m' \leq q - 2$, $b' = ba^{m'} \pmod q$ является B -гладким, т. е. $b' = p_1^{\beta_1} \dots p_s^{\beta_s}$.

5. Искомый логарифм $x = (\beta_1x_1 + \dots + \beta_sx_s - m') \pmod{(q-1)}$.

Для нахождения дискретного логарифма по индекс-методу требуется затратить $L_{1/2, c}(q)$ операций, где c – положительная константа. Наиболее эффективная на сегодняшний день модификация метода – метод решета числового поля – позволяет вычислить дискретный логарифм за время $L_{1/3, 1, 923}(q)$.

15.11. ЗАДАНИЯ

1. В системе аутентификации, основанной на схеме RSA, пользователь A выбрал открытый ключ $e = 7$ и $n = 77$. Если он получил от B число 23, то что A должен ответить, чтобы аутентифицировать себя?
 2. В системе аутентификации, основанной на схеме Рабина, A выбирает в качестве открытых ключей $B = 2$, $N = 200$. C посыпает A число $R = 168$. Как должен должен ответить A , чтобы убедить C , что именно ему попало сообщение?
 3. В той же схеме B получает сообщение $m_1 m_{273}$. Если оно подлинное, то что представляют собой m_1 и m_2 ?
 4. В схеме подписи, основанной на RSA, пользователи A и B имеют открытые ключи $e_A = 3$, $n_A = 15$; $e_B = 7$, $n_B = 77$ соответственно. A хочет послать сообщение $M = 4$ как подпись к некоторому тексту. Какое целое число он посыпает?
 5. Доказать, что в схеме подписи, основанной на RSA, при подписи S_1 сообщения m_1 и подписи S_2 сообщения m_2 сообщение $m_1 m_2$ можно подписать $S_1 S_2$.
 6. В схеме подписи Рабина A имеет открытый ключ N_A и желает подписать сообщения $m_1 = 9$ и $m_2 = 29$. Какими будут соответствующие подписи S_1 и S_2 ?
 7. Если S – подпись к сообщению M , то как следует подписать сообщение $4M$ в схеме подписи Рабина?
 8. В схеме подписи RSA пользователь A имеет открытый ключ $e = 11$, $n = 899$. Как он подпишет сообщение 876?
 9. Предположим, что мы знаем подписи S_1 и S_2 сообщений m_1 и m_2 , используя схему Рабина
- $$S_i = E(m_i) \equiv m_i(m_i + B) \pmod{N}, \quad i = 1, 2.$$
- Показать, что для некоторых значений открытого ключа B легко подписать сообщение $m_1 m_2$.
10. Найти единицу группы G . Разработать алгоритм поиска элементов $g \in G$, имеющих порядок q (см. п. 15.7).
 11. Сравнить вычислительную сложность систем электронной цифровой подписи DSS, ГОСТ Р 34.10-94, СТБ 1176.2-99 (см. п. 15.7). Точнее, сколько требуется выполнить умножений (возведений в степень) по модулю p (или q) для выработки (проверки) подписи?
 12. Доказать, что всякое число из множества $\{0, 1, \dots, n - 1\}$ можно представить в виде $im + j$, $0 \leq i, j < m$, и совпадение $a^j = ba^{-im}$ в методе больших-малых шагов всегда будет найдено (см. п. 15.10.2).

13. Написать программу вычисления дискретного логарифма по методу больших-малых шагов. Вычислить $\text{ind}_7 5$ в группе \mathbb{Z}_{250837}^* (см. п. 15.10.2).

14. Дать верхнюю оценку количества производимых операций в модификации бинарного метода скалярного умножения, предварительно написав алгоритм скалярного умножения по этому методу (см. п. 15.9).

15. Существенно ли требование $r, s \neq 0$ в алгоритме подписи (см. п. 15.8) и почему?

16. Для метода $p - 1$ проанализировать случаи, когда d не является собственным делителем n . Доказать:

1) если $d = 1$, то $p - 1$ не является B -гладким ни для одного простого делителя p числа n ;

2) если $d = n$, то $p - 1$ является B -гладким для всякого простого делителя p числа n (см. п. 15.10.1).

17. Доказать, что в методе квадратичного решета среди делителей $Q(x)$ нет таких нечетных простых p , что $\left(\frac{n}{p}\right) = -1$, и такие p следует исключить из базы множителей S_B (см. п. 15.10.1).

18. Написать программу факторизации по методу $p - 1$. Факторизовать число $n = 1846202297$, выбрав $B = 23$ (см. п. 15.10.1).

Г л а в а 16

ПРОТОКОЛЫ ФОРМИРОВАНИЯ ОБЩЕГО КЛЮЧА

16.1. ГОЛОВОЛОМКИ МЕРКЛЯ

Пусть Алиса и Боб – абоненты некоторой информационной системы. Данные в системе передаются по открытым каналам связи, которые могут прослушиваться противником Виктором. Для защиты от Виктора Алиса и Боб используют блочную криптосистему и выполняют шифрование сообщений на секретных ключах, которые им поставляет Трент. Если в системе имеется n абонентов, то для организации взаимодействия всех возможных пар потребуется $n(n - 1)/2$ ключей. Число ключей быстро растет с ростом n . Кроме этого, ключи следует распространять по секретным каналам связи, организация и поддержка которых может быть очень трудоемкой. Возникает вопрос: можно ли обойтись без Трента и без секретных каналов?

Положительный ответ на этот вопрос дал Р. Меркль в 1972 г. Меркль предложил протокол, с помощью которого стороны могут согласовать общий секретный ключ по *аутентифицируемому каналу связи* (АКС), занимающему промежуточное положение между открытым и секретным. Виктор может перехватить сообщение, передаваемое по АКС, но не может изменить его так, чтобы это не было обнаружено Алисой или Бобом. Виктор не может также передавать по АКС свои сообщения от чужого имени. Другими словами, АКС обеспечивает контроль целостности и подлинности данных, но не обязательно обеспечивает их конфиденциальность.

ПРОТОКОЛ МЕРКЛЯ

Предназначен для формирования сторонами общего секретного ключа K_i

Стороны: Алиса, Боб.

Каналы: АКС.

Параметры: N и M .

Шаги:

1. Алиса составляет и отправляет Бобу список из N головоломок. Любой абонент (и Боб, и Виктор) может решить головоломку за время $O(M)$. Решением i -й головоломки является ключ K_i , выбранный Алисой. В качестве головоломки можно использовать результат зашифрования пары (**«головоломка»**, K_i) на ключе $\theta_i \in \Theta$, $|\Theta| = M$. Решение состоит в проведении атаки «грубой силой» по определению θ_i при известном открытом тексте **«головоломка»**.

2. Боб выбирает головоломку со случайным номером i , решает ее, определяет ключ K_i и отправляет Алисе сообщение «Привет», зашифрованное на K_i .
 3. Алиса просматривает K_1, \dots, K_N и находит среди них ключ K_i , на котором было зашифровано сообщение «Привет».
-

Обсудим надежность протокола. Боб решает всего одну головоломку и тратит на это время $O(M)$. Алисе требуется проверить не более N ключей, что можно сделать за время $O(N)$. А вот Виктору для проверки тех же ключей потребуется время $O(NM)$, поскольку для определения каждого ключа K_i ему надо решить новую головоломку.

Страна	Время
Алиса	$O(N)$
Боб	$O(M)$
Виктор	$O(NM)$

Управляя N и M , можно добиться того, что время $O(NM)$ будет неприемлемо большим, хотя вычисления за время $O(N)$ и $O(M)$ будут устраивать Алису и Боба.

16.2. ПРОТОКОЛ ДИФФИ – ХЕЛЛМАНА

Головоломки Меркля не нашли практического применения. Они так и остались концептом, демонстрирующим возможность организации принципиально новой системы криптографической связи. Тем не менее протокол Меркля натолкнул У. Диффи и М. Хеллмана на разработку похожего протокола, который впоследствии был назван в их честь. *Протокол Диффи – Хеллмана* был описан в 1976 г. в знаменитой работе «Новые направления в криптографии».

В протоколе нам потребуется циклическая группа G порядка q . Будем записывать эту группу аддитивно, подразумевая соглашения для групп точек эллиптических кривых. Пусть $\text{descr}(G)$ – описание группы G , P – ее образующий, O – ее единица: $G = \{O, P, 2P, \dots, (q - 1)P\}$.

ПРОТОКОЛ ДИФФИ – ХЕЛЛМАНА

Предназначен для формирования общего секретного ключа K

Стороны: A (Алиса), B (Боб).

Каналы: АКС.

Параметры: $\text{descr}(G)$, P .

Шаги:

1. $A: d_A \xleftarrow{R} \{1, 2, \dots, q-1\}, Q_A \leftarrow d_A P.$
2. $B: d_B \xleftarrow{R} \{1, 2, \dots, q-1\}, Q_B \leftarrow d_B P.$
3. $A \xrightarrow{\text{АКС}} B: Q_A.$
4. $A \xleftarrow{\text{АКС}} B: Q_B.$
5. $A: K \leftarrow d_A Q_B.$
6. $B: K \leftarrow d_B Q_A.$

Корректность. Формируемые сторонами ключи совпадают: $d_A Q_B = d_A d_B P = d_B Q_A.$

Далее в главе мы будем последовательно уточнять данный протокол. Попутно будем рассматривать основные атаки на протоколы формирования общего ключа и соответствующие требования к этим протоколам.

Числа d_A, d_B называются *личными* ключами. Как и секретные ключи блочных или поточных криптосистем, личные ключи выбираются случайным образом, хранятся в секрете. Элементы Q_A, Q_B группы G называются *открытыми* ключами. Личный ключ d_A однозначно определяет открытый ключ Q_A и, наоборот, Q_A однозначно определяет d_A .

Открытые ключи обращаются в информационной системе в открытом виде и доступны Виктору. Однако при надлежащем выборе G нахождение кратного $(\text{descr}(G), P, d_A) \mapsto d_A P$ является простой вычислительной задачей, которую Алиса решит за приемлемое время, а дискретное логарифмирование $\text{DL}: (\text{descr}(G), P, Q_A) \mapsto d_A$ – вычислительно трудной, с которой Виктор не справится.

Оказывается, что для определенных семейств групп G растущего порядка q времена работы сторон имеют следующий вид:

Страна	Время
Алиса	$(\log q)^{O(1)}$
Боб	$(\log q)^{O(1)}$
Виктор	$O(q^{1/2})$ (экспоненциальное) или $2^{(c+o(1))(\log q)^\alpha(\log \log q)^{1-\alpha}}, 0 < \alpha < 1$ (субэкспоненциальное)

Для атаки на протокол Виктору не обязательно находить личные ключи сторон. Достаточно по известным $(\text{descr}(G), P, d_A P, d_B P)$ определить ключ $K = d_A d_B P$. Эта задача, известная как вычислительная задача Диффи – Хеллмана (CDH , см. п. 8.2), также признается вычислительно трудной для определенных семейств групп G .

Имеется полиномиальная сводимость задачи CDH к задаче DL в той же группе. Действительно, используя алгоритм решения DL , можно найти d_A

по $(\text{descr}(G), P, d_A P)$, а затем определить $K = d_A(d_B P)$. Обратное сведение DL к CDH на сегодняшний день не доказано. Доказательство этого сведения является одной из известных нерешенных проблем теоретической криптографии.

Протокол Диффи – Хеллмана может быть преобразован в протокол выработки общего ключа тремя абонентами. Пусть C (Клара) – третий абонент. Клара генерирует личный ключ $d_C \xleftarrow{R} \{1, 2, \dots, q - 1\}$ и определяет открытый ключ $Q_C \leftarrow d_C P$. Стороны обмениваются между собой открытыми ключами, а затем выполняют следующие пересылки:

$$\begin{aligned} A &\xrightarrow{\text{АКС}} B: d_A Q_C; \\ B &\xrightarrow{\text{АКС}} C: d_B Q_A; \\ C &\xrightarrow{\text{АКС}} A: d_C Q_B. \end{aligned}$$

По окончании пересылок каждая из сторон находит один и тот же общий ключ:

$$\begin{aligned} A: K &\leftarrow d_A(d_C Q_B); \\ B: K &\leftarrow d_B(d_A Q_C); \\ C: K &\leftarrow d_C(d_B Q_A). \end{aligned}$$

16.3. АТАКА «ПРОТИВНИК ПОСЕРЕДИНЕ»

Аутентифицируемый канал связи занимает промежуточное положение между секретным и открытым. Реализовать АКС проще, чем СКС. Представим, что Боб уехал в далекую страну, все каналы связи с ним проходят по специальному кабелю, проложенному по дну океана, и потенциально прослушиваются Виктором. Алиса вполне резонно сомневается в конфиденциальности пересылаемых Бобу данных. Алиса могла бы зашифровать данные, но для этого нужен ключ, который снова требуется передать конфиденциально. С организацией СКС ничего не получается. С другой стороны, Алиса и Боб вполне могут организовать АКС. Например, Алиса и Боб могут обменяться данными по электронной почте (ОКС), а затем проверить целостность и подлинность данных по телефону (АКС). Контроль подлинности данных поддерживается голосовой (характерный тембр голоса) и вербальной (характерные жаргонизмы) аутентификацией абонентов, целостность – проверкой контрольных характеристик переданных данных. Отметим, что телефонный АКС рекомендован в системе защищенной электронной почты PGP при обмене открытыми ключами.

При выполнении протокола Диффи – Хеллмана стороны вырабатывают общий секрет K , с помощью которого могут организовать СКС. Таким образом, протокол Диффи – Хеллмана фактически позволяет преобразовать АКС в СКС.

Можно ли отказаться от АКС и передавать открытые ключи по ОКС? Оказывается, что нет. Если CDH – вычислительно трудная задача, то протокол Диффи – Хеллмана является стойким относительно атак пассивного противника Виктора. Однако ситуация кардинальным образом меняется, если у Виктора есть возможность не только перехватывать, но и менять сообщения, пересылаемые по каналу связи.

Виктор может провести следующую *атаку «противник посередине»*:

1. Виктор выбирает случайные d'_A, d'_B и меняет в канале связи Q_A на $Q'_A = d'_A P$, Q_B на $Q'_B = d'_B P$.
2. По окончании протокола Алиса сформирует ключ $K_1 = d_A Q'_B$, а Боб – ключ $K_2 = d_B Q'_A$.
3. Оба ключа известны Виктору: $K_1 = d'_B Q_A$, $K_2 = d'_A Q_B$.

Пусть по завершении протокола Алиса использует K_1 для шифрования сообщений, передаваемых Бобу по ОКС. Виктор перехватывает зашифрованные сообщения, расшифровывает их на K_1 и зашифровывает на K_2 . Виктор прочитывает все сообщения Алисы, но ни Алиса, ни Боб от этом даже не догадываются.

Атаку «противник посередине» в англоязычной литературе принято обозначать MITM (от Man-In-The-Middle). Атака является универсальной в том смысле, что может применяться к широкому классу протоколов, стороны которых не связаны АКС. Как показывает следующий пример, протоколы могут быть даже не криптографическими.

Пример 16.1 (CAPTCHA). С помощью тестов CAPTCHA (от англ. Completely Automated Public Turing test to tell Computers and Humans Apart) машина (компьютер) проверяет, что взаимодействует с человеком. Машина преобразует текст в графическую картинку, искажая начертания символов, и предлагает восстановить по картинке текст. Человек это делает легко, а вот машина (другой компьютер) – нет. Для автоматического прохождения тестов с сайта A Виктор разработал сайт B , на котором дублирует тесты. Тестирование проходят люди, их ответы пересыпаются на сайт A и засчитываются как правильные. Проделанные Виктором манипуляции по сути являются атакой «противник посередине».

16.4. СЕРТИФИКАТЫ ОТКРЫТЫХ КЛЮЧЕЙ

Для защиты от атаки «противник посередине» Алиса и Боб должны распределить свои открытые ключи так, чтобы можно было контролировать их подлинность и целостность. Первоначально вопросам распространения открытых ключей не уделялось большого внимания. Считалось, что открытые ключи будут размещаться в некотором общедоступном справочнике Трента и абоненты будут получать доступ к этому справочнику по АКС.

Со временем стало понятно, что управление открытыми ключами является ахиллесовой пятой криптографии с открытыми ключами. Возникли вопросы: как защищать справочник? как организовать доступ к нему? как организовать защиту каналов доступа? Вместо глобального общедоступного справочника были разработаны более гибкие решения. Самое распространенное из них основано на *сертификатах открытых ключей* и регламентируется международным стандартом X.509, введенным в 1988 г. Фактически X.509 предлагает способ организации *распределенного* справочника. Стандарт X.509 принят в нашей стране в виде СТБ 34.101.19.

Пусть Трент также вырабатывает личный ключ d_T и определяет по нему открытый ключ Q_T . Трент использует пару (d_T, Q_T) не в протоколах формирования общего ключа, а в алгоритмах электронной цифровой подписи. Будем для простоты считать, что в алгоритмах ЭЦП используются параметры $\text{descr}(G)$ и P , и что ключи d_T и Q_T устроены также, как и ключи протокола Диффи – Хеллмана (например, алгоритмы ЭЦП построены по схеме Шнорра). С помощью d_T Трент вырабатывает ЭЦП $S_T(X)$ сообщения $X \in \{0, 1\}^*$, а с помощью Q_T любой другой абонент проверяет соответствие подписи сообщению. Положительный результат проверки означает, что ЭЦП создана Трентом (контроль подлинности) и с момента создания подписи сообщение X не изменялось (контроль целостности).

Сертификат открытого ключа абонента – это данные об абоненте, его открытом ключе, атрибутах открытого ключа, заверенные ЭЦП Трента. В стандарте X.509 Трента называют *удостоверяющим центром*. Сертификат Алисы обозначается $\text{Cert}_T(Id_A, Q_A)$ и состоит из следующих полей:

$\text{Cert}_T(Id_A, Q_A) =$	Версия сертификата
	Номер сертификата
	Id_T (полное имя Трента)
	Срок действия сертификата
	Id_A (полное имя Алисы)
	$\text{descr}(G), P$
	Q_A
	Расширения (дополнительные атрибуты)
	S_T (объединение предыдущих полей)

Фактически, сертификат связывает имя Алисы Id_A и ее открытый ключ Q_A .

Боб, располагая сертификатом $\text{Cert}_T(Id_A, Q_A)$ и открытым ключом Q_T , может проверить подлинность и целостность Q_A . Открытый ключ Трента также может распространяться в виде сертификата $\text{Cert}_{T_1}(Id_T, Q_T)$, заверенного еще одним удостоверяющим центром T_1 . Для проверки

открытого ключа T_1 может использоваться еще один сертификат $\text{Cert}_{T_2}(Id_{T_1}, Q_{T_1})$ и т. д. В конце концов образуется *цепочка сертификатов*

$$\text{Cert}_{T_1}(Id_T, Q_T), \dots, \text{Cert}_{T_{n-1}}(Id_{T_n}, Q_{T_n}), \text{Cert}_{T_n}(Id_{T_n}, Q_{T_n}).$$

Последнее звено цепочки – это самоподписанный сертификат *корневого удостоверяющего центра* T_n . Такие сертификаты, как правило, выпускаются известными компаниями, оказывающими услуги в области криптографии с открытым ключом. Сертификаты корневых удостоверяющих центров включаются в дистрибутивы операционных систем, «пропиваются» в браузерах и т. д.

16.5. ПРОТОКОЛ С СЕРТИФИКАТАМИ

Модернизируем протокол Диффи – Хеллмана, используя в нем сертификаты. Здесь и далее при описании протоколов будем опускать параметры $\text{descr}(G)$ и P .

ПРОТОКОЛ ДИФФИ – ХЕЛЛМАНА С СЕРТИФИКАТАМИ

Стороны: A (Алиса), B (Боб), T (Трент).

Каналы: ОКС, АКС.

Генерация и распределение ключей Трента:

1. $T: d_T \xleftarrow{R} \{1, 2, \dots, q - 1\}, Q_T \leftarrow d_T P.$
2. $T \xrightarrow{\text{АКС}} A: Q_T.$
3. $T \xrightarrow{\text{АКС}} B: Q_T.$

Выдача сертификата Алисе (Бобу – аналогично):

1. $A: d_A \xleftarrow{R} \{1, 2, \dots, q - 1\}, Q_A \leftarrow d_A P.$
2. $A \xrightarrow{\text{АКС}} T: Q_A.$
3. T : формирует $\text{Cert}_T(Id_A, Q_A)$, используя d_T .
4. $A \xleftarrow{\text{АКС}} T: \text{Cert}_T(Id_A, Q_A).$

Формирование общего ключа K :

1. $A \xrightarrow{\text{ОКС}} B: \text{Cert}_T(Id_A, Q_A).$
2. $A \xleftarrow{\text{ОКС}} B: \text{Cert}_T(Id_B, Q_B).$
3. A : проверяет $\text{Cert}_T(Id_B, Q_B)$ на Q_T , определяет $K \leftarrow d_A Q_B$.
4. B : проверяет $\text{Cert}_T(Id_A, Q_A)$ на Q_T , определяет $K \leftarrow d_B Q_A$.

Здесь и далее при описании протоколов считаем, что при неверной проверке на любом шаге стороны немедленно прерывают выполнение протокола с ошибочным результатом.

В новом протоколе открытые ключи Алисы передаются в виде сертификатов по ОКС. Пересылку сертификатов предваряет распределение открытого ключа Трента и выдача им сертификатов. Подготовительные мероприятия фактически преобразуют канал передачи сертификатов в АКС.

Трент может распространять Q_T в виде сертификата или в виде цепочки сертификатов, корневой удостоверяющий центр которой признается и Алисой, и Бобом. Передача Q_A, Q_B реализуется тем, что сторона A или B обращается в удостоверяющий центр T , там проверяется подлинность стороны (например, паспортная аутентификация), а затем принимается ее открытый ключ.

Далее для простоты будем опускать предварительные этапы распределения ключей Трента и выдачи сертификатов. По умолчанию, пересылки в последующих протоколах выполняются по открытым каналам связи.

16.6. ПРОТОКОЛЫ МТИ

Ключ K , который формируется в результате выполнения последнего протокола, впоследствии используется для шифрования, имитозащиты и для решения других криптографических задач в сеансе связи между Алисой и Бобом. Поэтому ключ K называется *сеансовым*.

Существует угроза того, что ключ K будет использоваться неправильно и станет известен Виктору. Например, Алиса может просто забыть удалить K по завершении сеанса. Важно, чтобы компрометация ключа одного сеанса не сказалась на стойкости ключей других сеансов. Соответствующее требование к протоколам называется *защитой при компрометации сеансовых ключей* (KKS, Known Key Security).

В последнем протоколе ключи d_A, Q_A, d_B, Q_B целесообразно сделать долговременными или, как еще говорят, *статическими*. Действительно, процедура выдачи сертификата открытого ключа является достаточно трудоемкой, и поэтому использование сертификата только в одном сеансе – весьма неэффективное решение, лишающее применение открытых ключей практического смысла. С другой стороны, при многократном использовании одних и тех же личных и открытых ключей протокол не будет удовлетворять требованию KKS, поскольку вырабатываемые сторонами ключи K будут все время одинаковыми.

Т. Мацумото, Ю. Такашима и Х. Имаи разработали в 1986 г. протоколы, которые впоследствии получил название МТИ по первым буквам фамилий авторов. В протоколах МТИ кроме статических используются еще одноразовые или, как их еще называют, *эфемерные* ключи. Эфемерные ключи выбираются сторонами случайным образом. Их применение делает сеансовые ключи K также случайными даже при фиксированных статических ключах, что обеспечивает выполнение свойства KKS.

ПРОТОКОЛ MTI/A0

Шаги:

1. $A: u_A \xleftarrow{R} \{1, 2, \dots, q-1\}, V_A \leftarrow u_A P.$
2. $A \rightarrow B: \text{Cert}_T(Id_A, Q_A), V_A.$
3. $B: u_B \xleftarrow{R} \{1, 2, \dots, q-1\}, V_A \leftarrow u_B P.$
4. $A \leftarrow B: \text{Cert}_T(Id_B, Q_B), V_B.$
5. $A: \text{проверяет } \text{Cert}_T(Id_B, Q_B), K \leftarrow d_A V_B + u_A Q_B.$
6. $B: \text{проверяет } \text{Cert}_T(Id_A, Q_A), K \leftarrow d_B V_A + u_B Q_A.$

Корректность: $d_A V_B + u_A Q_B = (d_A u_B + d_B u_A)P = d_B V_A + u_B Q_A.$

ПРОТОКОЛ MTI/C0

Ограничения: q – простое.

Шаги:

1. $A \rightarrow B: \text{Cert}_T(Id_A, Q_A).$
2. $A \leftarrow B: \text{Cert}_T(Id_B, Q_B).$
3. $A: \text{проверяет } \text{Cert}_T(Id_B, Q_B), u_A \xleftarrow{R} \{1, 2, \dots, q-1\}, V_A \leftarrow u_A Q_B.$
4. $B: \text{проверяет } \text{Cert}_T(Id_A, Q_A), u_B \xleftarrow{R} \{1, 2, \dots, q-1\}, V_B \leftarrow u_B Q_A.$
5. $A \rightarrow B: V_A.$
6. $A \leftarrow B: V_B.$
7. $A: K \leftarrow (d_A^{-1} \bmod q)u_A V_B.$
8. $B: K \leftarrow (d_B^{-1} \bmod q)u_B V_A.$

Корректность: $d_A^{-1}u_A V_B = u_A u_B P = d_B^{-1}u_B V_A.$

В этих протоколах u_A, u_B – эфемерные личные ключи, V_A, V_B – эфемерные открытые ключи.

Протоколы MTI имеют слабости, которые мы сейчас рассмотрим, параллельно обсудив дополнительные требования к протоколам формирования общего ключа.

Задача от «чтения назад». Долговременные личные ключи d_A, d_B могут быть скомпрометированы, например, если потеряны ключевые носители, на которых они хранятся. Даже при раскрытии долговременных ключей противник не должен получать возможность определять предыдущие сеансовые ключи. Соответствующее требование к протоколам называется *защитой от «чтения назад»* (PFS, Perfect Forward Secrecy).

Рассмотрим выполнение свойства PFS для протоколов MTI.

Пусть Виктор получил оба ключа — d_A , d_B — протокола MTI/C0. Тогда он может вычислить $u_AP = (d_B^{-1} \bmod q)V_A$ и $u_BP = (d_A^{-1} \bmod q)V_B$. Для определения K Виктору надо найти u_Au_BP по тройке (P, u_AP, u_BP) . Но это трудная задача CDH и, таким образом, протокол MTI/C0 обладает свойством PFS.

Пусть Виктор получил один из ключей, например d_A , протокола MTI/C0. Определение $K = d_AV_B + u_AQ_B$ для Виктора равносильно определению $u_AQ_B = u_Ad_BP$ по известным $u_AP = V_A$ и $d_BP = Q_B$. Но это снова трудная задача CDH. Ситуация меняется, если Виктор получает два ключа. В этом случае он может найти K по данным перехвата:

$$K = d_AV_B + d_BV_A.$$

Таким образом, MTI/A0 обеспечивает защиту от «чтения назад» при компрометации одного долговременного личного ключа и не обеспечивает при компрометации двух ключей.

Атака «малая подгруппа». В протоколах MTI стороны обмениваются элементами V_A, V_B группы G . Эти элементы представляются двоичными словами. Стороны не проверяют формат присыпаемых данных. Виктор может воспользоваться этим, навязывая сторонам использование вместо G другой группы H , которая имеет малый порядок и, следовательно, элементы которой могут быть легко угаданы.

Пусть в протоколе MTI/C0 группа G представляет собой группу точек эллиптической кривой над полем \mathbb{F}_p : $G = E(\mathbb{F}_p)$. Эллиптическая кривая задается уравнением E : $y^2 = x^3 + ax + b$, $a, b \in \mathbb{F}_p$. Коэффициент b определяет строение группы, базовую точку P , но не используется непосредственно в формулах сложения точек. Виктор учитывает данный факт и проводит следующую атаку «малая подгруппа».

1. Виктор находит коэффициент $b' \in \mathbb{F}_p$, который определяет новое уравнение E' : $y^2 = x^3 + ax + b'$ и новую группу $H = E'(\mathbb{F}_p)$. Виктор подбирает b' так, чтобы группа H содержала точку W малого порядка t . Важно, что вычисления с элементами $E'(\mathbb{F}_p)$ ведутся по тем же формулам, что и вычисления с элементами $E(\mathbb{F}_p)$.

2. Виктор меняет открытые ключи V_A, V_B на произвольные точки $W_A, W_B \in \langle W \rangle$.

3. По окончании протокола Алиса и Боб определят случайные ключи $K_A, K_B \in \langle W \rangle$, причем $K_A = K_B$ с вероятностью $1/t$. Если t невелико, то ключи совпадут и стороны не обнаружат подмены, а противник сможет перебрать все элементы $\langle W \rangle$ и определить совпадшие ключи.

Как правило, существует простой способ защиты от атаки «малая подгруппа». Сторонам протокола следует всего лишь дополнительно проверять, что присланные сообщения действительно представляют элементы G .

16.7. АУТЕНТИФИКАЦИЯ

После завершения протокола Диффи – Хеллмана стороны располагают ключом K . В некоторых случаях Алиса и Боб должны убедиться, что ключ действительно общий. Для этого стороны могут выполнить дополнительные шаги, которые называются *подтверждением ключа*. Подтверждение может быть организовано с помощью системы имитозащиты (см. п. 12.15), как это сделано в следующем протоколе. Будем использовать систему $I = \{I_K : K \in G\}$, функции которой ставят в соответствие сообщениям $X \in \{0, 1\}^*$ имитовставки $Z = I_K(X)$.

ПРОТОКОЛ Подтверждение ключа

Стороны A и B проверяют, что ключ K , выработанный по протоколу Диффи – Хеллмана, действительно общий

Шаги (после шагов по формированию K):

1. $A: Z_A \leftarrow I_K('1')$.
2. $B: Z_B \leftarrow I_K('2')$.
3. $A \rightarrow B: Z_A$.
4. $A \leftarrow B: Z_B$.
5. $A: Z_B \stackrel{?}{=} I_K('2')$.
6. $B: Z_A \stackrel{?}{=} I_K('1')$.

Стороны протокола демонстрируют друг другу, что могут определять имитовставки заданных сообщений на ключе K и, таким образом, показывают знание K . Если предшествующий протокол является надежным, то знать K могут только легитимные стороны. Поэтому протокол формирования общего ключа с дополнительным подтверждением может использоваться для аутентификации сторон. Более того, в некоторых случаях формирование общего ключа является вспомогательным этапом аутентификации.

Различные сообщения ‘1’ и ‘2’ защищают от атаки «зеркализации»: если сообщения будут одинаковыми, то Виктор сможет выдать себя за Боба, ответив Алисе ее же имитовставкой.

Вместо имитозащиты можно использовать шифрование. Но так делать не рекомендуют, поскольку шифрование служит для обеспечения конфиденциальности сообщений, а не для их аутентификации (т. е. проверки подлинности).

Ошибочное разделение ключа. Если протокол МТ1/А0 используется только для аутентификации, то Виктор может провести следующую атаку:

1. Виктор выбирает $e \in \{1, 2, \dots, q - 1\}$ и получает у Трента сертификат $\text{Cert}(Id_{\tilde{A}}, Q_{\tilde{A}})$ на имя $I_{\tilde{A}}$ и на открытый ключ

$$Q_{\tilde{A}} = eQ_A.$$

2. Во время выполнения протокола Виктор перехватывает эфемерный открытый ключ V_A и отсылает его Бобу вместе с сертификатом $\text{Cert}(Id_{\tilde{A}}, Q_{\tilde{A}})$. Ответный ключ V_B Виктор меняет на eV_B и отсылает его Алисе вместе с сертификатом Боба.

3. Алиса и Боб вычислят общий секретный ключ

$$K = d_A e V_B + u_A Q_B = u_B Q_{\tilde{A}} + d_B V_A,$$

при этом Боб будет ошибочно считать, что разделяет его с Виктором (хотя Виктор даже не знает K).

Описанная атака называется «*ошибочное разделение ключа*» (UKS, Unknown Key Share). Атака является достаточно общей, известны случаи ее успешного применения к нескольким протоколам аутентификации.

Приведем пример использования атаки.

Пример 16.2. Пусть B – банк, A – его клиент. Банк и клиент формируют общий ключ K и выполняют на нем подтверждение ключа. Если подтверждение ключа завершено успешно, то банк признает клиента подлинным и переводит на его счет деньги. Виктор, выполняя описанные выше манипуляции, может вынудить банк перевести деньги на свой счет.

Атака UKS не состоится, если при выдаче сертификата Трент предложит Виктору доказать владение личным ключом $d_{\tilde{A}}$, который соответствует открытому ключу $Q_{\tilde{A}}$ (например, Трент может предложить Виктору подписать на $d_{\tilde{A}}$ запрос на выдачу сертификата). Виктор не знает

$$d_{\tilde{A}} = ed_A \bmod q$$

и поэтому проверку Трента не пройдет.

Еще одним способом защиты от атаки является усиленное подтверждение ключа, при котором стороны протокола формируют имитовставки Z_A, Z_B по новым правилам:

$$Z_A \leftarrow I_K('1' \parallel Id_A \parallel Id_B \parallel V_A \parallel V_B);$$

$$Z_B \leftarrow I_K('2' \parallel Id_B \parallel Id_A \parallel V_B \parallel V_A).$$

Имитозащита идентификаторов Id_A, Id_B указывает на пару сторон протокола. Имитозащита эфемерных ключей указывает на их сеанс.

16.8. ПРОТОКОЛ MQV

Одним из наиболее обоснованных протоколов формирования общего ключа типа Диффи – Хеллмана является протокол MQV. Этот протокол был разработан А. Менезесом (M), М. Кью (Q) и С. Ванстоуном (V) в 1995 г., а затем несколько раз дорабатывался.

Приведем протокол в редакции, которая отличается от первоначальной. Будем считать, что порядок группы G близок к 2^{2l} . Обозначим $G^* = G \setminus \{O\}$. Будем использовать в протоколе функцию хэширования $\varphi: G^* \times \{0, 1\}^* \rightarrow \{2^l, \dots, 2^{l+1} - 1\}$.

ПРОТОКОЛ MQV

Шаги:

1. $A: u_A \xleftarrow{R} \{1, 2, \dots, q - 1\}, V_A \leftarrow u_A P.$
2. $A \rightarrow B: \text{Cert}_T(Id_A, Q_A), V_A.$
3. $B: u_B \xleftarrow{R} \{1, 2, \dots, q - 1\}, V_B \leftarrow u_B P.$
4. $A \leftarrow B: \text{Cert}_T(Id_B, Q_B), V_B.$
5. $A:$
 - 5.1. Проверяет $\text{Cert}_T(Id_B, Q_B);$
 - 5.2. Проверяет, что $V_B \in G^*$;
 - 5.3. $X_A \leftarrow Id_A \parallel Id_B \parallel V_B, X_B \leftarrow Id_B \parallel Id_A \parallel V_A;$
 - 5.4. $s_A \leftarrow (u_A - \varphi(V_A, X_A)d_A) \bmod q;$
 - 5.5. $K \leftarrow s_A(V_B - \varphi(V_B, X_B)Q_B).$
6. $B:$
 - 6.1. Проверяет $\text{Cert}_T(Id_A, Q_A);$
 - 6.2. Проверяет, что $V_A \in G^*$;
 - 6.3. $X_A \leftarrow Id_A \parallel Id_B \parallel V_B, X_B \leftarrow Id_B \parallel Id_A \parallel V_A;$
 - 6.4. $s_B \leftarrow (u_B - \varphi(V_B, X_B)d_B) \bmod q;$
 - 6.5. $K \leftarrow s_B(V_A - \varphi(V_A, X_A)Q_A).$

Корректность: $s_A(V_B - \varphi(V_B, X_B)Q_B) = s_A s_B P = s_B(V_A - \varphi(V_A, X_A)Q_A).$

Используемые в протоколе числа s_A, s_B называются *одноразовыми подписями*. Название объясняется тем, что эти числа фактически являются вторыми частями подписи Шнорра:

ЭЦП Шнорра	Протокол MQV
$k \xleftarrow{R} \{1, 2, \dots, q - 1\}$	$u_A \xleftarrow{R} \{1, 2, \dots, q - 1\}$
$R \leftarrow kP$	$V_A \leftarrow u_A P$
$s_0 \leftarrow \varphi(R, X)$	$\varphi(V_A, X_A)$
$s_1 \leftarrow (k - s_0 d) \bmod q$	$s_A \leftarrow (u_A - \varphi(V_A, X_A)d_A) \bmod q$

Определение s_A без d_A (или s_B без d_B) соответствует задаче **Schnorr**, состоящей в построении поддельной подписи. Для криптографически надежных G и φ задача **Schnorr** признается трудной.

Обсудим стойкость протокола MQV.

Стойкость (пассивный противник). Пусть Виктор не вмешивается в выполнение протокола, а только перехватывает сообщения, которыми обмениваются стороны. Виктору требуется найти $K = s_A s_B P$. Используя данные перехвата, Виктор может вычислить $s_A P = V_A - \varphi(V_A, X_A) Q_A$ и $s_B P = V_B - \varphi(V_B, X_B) Q_B$. Но даже при этом для определения K ему требуется решить трудную задачу CDH.

Стойкость (активный противник). Пусть Виктор может вмешиваться в выполнение протокола и хочет выдать себя за Боба. Для этого он по V_A, Q_A, V_B и Q_B должен найти $V \in G^*$ и $K \in G$ такие, что

$$K = s s_A P, \quad sP = V - \varphi(V, X_B) Q_B, \quad s \in \{1, 2, \dots, q - 1\}.$$

Виктор может:

- 1) выбрать V , определить sP , а затем найти s с последующим определением $K = s(s_A P)$. Но нахождение s есть трудная задача DL;
- 2) выбрать V , определить sP , а затем найти K по $sP, s_A P$. Но это задача CDH;
- 3) найти (V, s) такие, что $sP = V - \varphi(V, X_B) Q_B$, а затем определить K . Но нахождение пары (V, s) есть задача **Schnorr**.

Как видим, во всех рассмотренных случаях противник сталкивается с необходимостью решения некоторой трудной задачи.

16.9. ПРОТОКОЛ TLS

Протокол TLS (Transport Layer Security) – это наиболее распространенный на сегодняшний день протокол защиты соединений между клиентом и сервером в сети Интернет. Протокол был разработан в 1996 г. компанией Netscape и первоначально назывался SSL. Протокол прошел несколько существенных модернизаций. На данный момент наиболее совершенной считается редакция 1.2, принятая в качестве RFC 5246 в 2008 г.

TLS обеспечивает взаимную аутентификацию сторон протокола, конфиденциальность и контроль целостности передаваемых между сторонами данных. TLS встраивается в стек коммуникационных протоколов поверх транспортного уровня и обеспечивает защиту данных этого уровня.

TLS представляет собой набор субпротоколов различного назначения. В частности, в TLS имеется субпротокол Handshake (рукопожатия), с помощью которого стороны формируют общий секретный ключ и параллельно проводят аутентификацию друг друга.

В начале выполнения Handshake стороны обмениваются случайными двоичными словами `client_random`, `server_random`. Затем вырабатывают предварительный общий ключ `pre_master_secret` и определяют окончательный общий ключ

$$\text{master_secret} = h(\text{pre_master_secret}, \text{client_random}, \text{server_random}),$$

где h – специальная хэш-функция.

RFC 5246 определяет следующие стандартные алгоритмы формирования общего ключа (A – клиент, B – сервер).

DH_anon (протокол Диффи – Хеллмана без сертификатов). Сервер пересыпает клиенту описание группы G и свой открытый ключ $V_B = u_B P$. Клиент пересыпает серверу свой открытый ключ $V_A = u_A P$. Стороны вычисляют общий ключ $u_A u_B P$, по которому строится `pre_master_key`. В протоколе не используются сертификаты и, таким образом, подлинность сторон не проверяется. Поэтому протокол не противостоит атаке «противник посередине» и его рекомендуется использовать только в специальных случаях.

DH_fixed (протокол Диффи – Хеллмана с сертификатами). Сервер пересыпает клиенту сертификат, который содержит описание G и открытый ключ $Q_B = d_B P$. Клиент по запросу сервера передает статический открытый ключ $Q_A = d_A P$. Если запроса от сервера нет, то клиент передает эфемерный открытый ключ $V_A = u_A P$. Стороны вычисляют общий ключ $d_A d_B P$ или $u_A d_B P$, по которому определяется `pre_master_key`.

DHE (протокол Диффи – Хеллмана с эфемерными ключами). Сервер пересыпает клиенту сертификат, открытый ключ $Q_B = d_B P$ которого можно использовать для проверки ЭЦП. Затем сервер передает описание G , свой эфемерный открытый ключ $V_B = u_B P$ и подписывает эти данные вместе с `client_random`, `server_random` на своем личном ключе d_B . Клиент проверяет ЭЦП и передает свой открытый ключ $V_A = u_A P$. Стороны вычисляют общий ключ $u_A u_B P$, по которому строится `pre_master_key`. Сертификат клиента может не передаваться.

Транспорт ключа. Сервер передает клиенту свой сертификат, открытый ключ которого можно использовать для шифрования. Клиент запишировывает на этом ключе `pre_master_secret` и передает шифртекст серверу. Сервер выполняет расшифрование `pre_master_secret` на своем личном ключе.

16.10. ЗАДАНИЯ

1. Разработать протокол типа Диффи – Хеллмана для формирования общего ключа между n абонентами.

2. Пусть $g = \begin{pmatrix} 1 & 2 \\ 1 & 0 \end{pmatrix}$ – матрица над полем \mathbb{F}_3 и $G = \langle g \rangle$ – циклическая группа, порожденная g . Доказать, что порядок G равняется 6. Стороны используют G для выработки общего ключа по протоколу Диффи – Хеллмана. Первая сторона выбирает $a = 4$, вторая сторона выбирает $b = 5$. Чему будет равняться общий ключ?

3. Пусть $g = \begin{pmatrix} 1 & 2 \\ 1 & 0 \end{pmatrix}$ – матрица над полем \mathbb{F}_5 и $G = \langle g \rangle$ – циклическая группа, порожденная g . Доказать, что порядок G равняется 4. Стороны используют G для выработки общего ключа по протоколу Диффи – Хеллмана. Первая сторона выбирает $a = 3$, вторая сторона выбирает $b = 2$. Чему будет равняться общий ключ?

4. Пусть $g = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 5 & 3 & 6 & 2 & 1 & 0 & 4 \end{pmatrix}$ – подстановка из $S(\mathbb{F}_7)$ и $G = \langle g \rangle$ – циклическая группа, порожденная g . Доказать, что порядок G равняется 10. Стороны используют G для выработки общего ключа по протоколу Диффи – Хеллмана. Первая сторона выбирает $a = 3$, вторая сторона выбирает $b = 4$. Чему будет равняться общий ключ?

5. Пусть $g = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 5 & 6 & 1 & 2 & 0 & 4 & 3 \end{pmatrix}$ – подстановка из $S(\mathbb{F}_7)$ и $G = \langle g \rangle$ – циклическая группа, порожденная g . Доказать, что порядок G равняется 12. Стороны используют G для выработки общего ключа по протоколу Диффи – Хеллмана. Первая сторона выбирает $a = 3$, вторая сторона выбирает $b = 5$. Чему будет равняться общий ключ?

6. Пусть $G = \langle g \rangle$ – группа простого нечетного порядка q . Алгоритм решения задачи CDH: $(g, g^a, g^b) \mapsto g^{ab}$ может быть преобразован в алгоритм решения задачи SqDH: $(g, g^a) \mapsto g^{a^2}$. Доказать, что верно и обратное: алгоритм решения SqDH может быть преобразован в алгоритм решения CDH.

7. Пусть d и m – взаимно простые натуральные числа, $g: x \mapsto x + d \bmod m$ – подстановка на \mathbb{Z}_m . Найти порядок g . Почему использование группы $G = \langle g \rangle$ в протоколе Диффи – Хеллмана не является безопасным?

8. Проанализировать алгоритмы формирования общего ключа TLS и установить, какие из них обеспечивают защиту от «чтения назад».

КОММЕНТАРИИ

Протоколы формирования общего ключа описываются в книгах [2, 38, 133, 153]. Имеется русскоязычный обзор [57], посвященный криптографическим протоколам в целом.

Протокол Диффи – Хеллмана введен в [88]. Протоколы МТІ определены в [130], а протокол MQV – в [134]. Схема ЭЦП Шнорра введена в [154].

Протокол TLS версии 1.2 определен в СТБ 34.101.65 «Информационные технологии и безопасность. Протокол защиты транспортного уровня (TLS)».

СТБ 34.101.66 «Информационные технологии и безопасность. Протоколы формирования общего ключа» определяет протоколы BMQV, BSTS и BPACЕ. Первый протокол уточняет MQV, второй – известную схему STS [89], третий – схему PACE [71]. Протокол BSTS обеспечивает анонимность: противник, который не вступает во взаимодействие со сторонами, а только перехватывает их сообщения, не получает информации о том, какие стороны участвуют в протоколе. Протокол BPACЕ позволяет сторонам сформировать общий ключ, используя общий пароль. Протокол BPACЕ позволяет сторонам сформировать общий ключ, используя общий пароль. Противнику, который перехватывает все сообщения протокола (вступая или не вступая во взаимодействие со сторонами), вычислительно трудно определить пароль, даже если он короткий или низкоэнтропийный. Выполняя сеанс протокола с легальной стороной, противник может проверить только один вариант пароля.

Глава 17

МЕТОДЫ И АЛГОРИТМЫ РАЗДЕЛЕНИЯ СЕКРЕТА

17.1. ОБЩАЯ ЗАДАЧА О РАЗДЕЛЕНИИ СЕКРЕТА

Основы теории разделения секрета были заложены в 1979 г. А. Шамиром и Г. Блейкли в связи со следующей задачей. Пусть имеется некоторая важная информация (секрет), которую нельзя целиком доверить отдельному лицу, а требуется так распределить ее в виде частичных секретов среди данного круга пользователей, чтобы лишь заранее определенные группы пользователей, объединяя свои частичные секреты, могли восстановить исходный секрет. Алгоритмы, решающие эту задачу, называются схемами разделения секрета (СРС).

Более точно, схема разделения секрета – это совокупность алгоритмов разделения и восстановления секрета и структуры доступа. Под структурой доступа понимается перечень заранее определенных (разрешенных) групп пользователей. Все остальные группы называются запрещенными. Для них задача восстановления секрета должна быть вычислительно сложной, а в лучшем случае – эквивалентной полному перебору. Пронумеруем всех пользователей (участников) и обозначим их множество через $P = \{1, 2, \dots, k\}$.

Простейшая структура доступа – пороговая. В такой структуре разрешенным является всякое подмножество, если число участников в нем не меньше, чем t . Если общее число участников равно k , то такая структура доступа называется (t, k) -пороговой. Число t называется порогом. Понятно, что $1 \leq t \leq k$. Соответствующая СРС называется (t, k) -пороговой схемой.

Естественным требованием для структур доступа является свойство монотонности:

$$A \in \Gamma(\mathcal{P}), A \subset B \Rightarrow B \in \Gamma(\mathcal{P}).$$

Далее оно предполагается всегда выполненным. Иными словами, если подмножество A разрешено, то всякое включающее его подмножество B также разрешено. Отсюда также следует, что всякая структура доступа задается набором своих минимальных по включению подмножеств Γ_{\min} , которое называют также базисом структуры доступа Γ .

Всякой структуре доступа соответствует структура отказа в доступе, или множество всех запрещенных подмножеств участников $\bar{\Gamma}(\mathcal{P})$. Она также обладает свойством монотонности в двойственном смысле:

$$A \in \overline{\Gamma(\mathcal{P})}, B \subset A \Rightarrow B \in \overline{\Gamma(\mathcal{P})}.$$

Иными словами, все подмножества запрещенного подмножества также запрещены. Поэтому структура отказа в доступе задается набором максимальных по включению запрещенных подмножеств $\bar{\Gamma}_{\max}$. Монотонность для разрешенных множеств означает, что при добавлении нового участника к разрешенному множеству получается еще одно разрешенное множество.

Пример 17.1. Рассмотрим структуру доступа с четырьмя участниками

$$\Gamma = \{\{1, 2\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 2, 3, 4\}, \{3, 4\}, \{1, 3, 4\}, \{2, 3, 4\}\}.$$

Тогда

$$\Gamma_{\min} = \{\{1, 2\}, \{3, 4\}\}.$$

$$\bar{\Gamma} = \{0, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}\},$$

$$\bar{\Gamma}_{\max} = \{\{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}\}.$$

При первом знакомстве с предметом напрашивается очень простое решение основной задачи, которое на поверку оказывается математически и практически несостоятельным.

Его суть сводится к следующему. Например, предлагаю 4 цифры банковского PIN-кода раздать четырем участникам с тем, чтобы они лишь с общего согласия могли пользоваться соответствующим счетом. Такой способ разделения секрета совсем неприемлем, так как уже три участника всего лишь за десять попыток угадают недостающую цифру. К счастью, есть другое решение этой задачи, лишенное указанного недостатка. Надо представить PIN-код s в виде

$$s = s_1 + s_2 + s_3 + s_4 \pmod{10^4},$$

где s_1, s_2, s_3, s_4 – четырехзначные десятичные числа. Достаточно элементарных знаний по арифметике, чтобы понять, что, зная лишь три слагаемых, сказать о сумме s ничего нельзя, так как s может быть любым элементом \mathbb{Z}_m , $m = 10^4$. Приведенный пример – лишь частный случай схемы анонимного согласия и еще более частный случай совершенной схемы разделения секрета. Эти вопросы подробно рассматриваются далее в этой главе, причем акцент делается на так называемом модулярном разделении секрета [58, 95, 96, 97]. В последних параграфах предполагается знакомство читателя с методом базисов Гребнера [24, 70].

В заключение приведем пример (t, t) -пороговой схемы.

В простом конечном поле \mathbb{F}_p , $p > t$, случайным образом выберем частичные секреты s_1, s_2, \dots, s_t , а в качестве секрета возьмем их сумму

$$s = s_1 + s_2 + \dots + s_t.$$

Очевидно, что $t - 1$ участников не обладают никакой информацией, кроме априорной, о секрете s и частичном секрете оставшегося участника. Эта схема оказывается схемой анонимного согласия. К сожалению, она пригодна лишь для этого частного случая.

В дальнейшем будем через s_1, s_2, \dots, s_k обозначать частичные секреты, через s — секрет, а через S — пространство секретов s , т. е. множество его возможных значений. В примере $S = \mathbb{F}_p$.

17.2. КРИТЕРИИ КАЧЕСТВА СХЕМ РАЗДЕЛЕНИЯ СЕКРЕТА

При разработке схем разделения секрета стараются удовлетворить некоторым естественным требованиям. К их числу в первую очередь относится требование *совершенности*, т. е. запрещенные множества участников не должны обладать никакой дополнительной информацией о значении секрета s , кроме имеющейся априорной. С другой стороны, желательно, чтобы размер частичного секрета был равен размеру основного секрета. Такие схемы называют *идеальными*.

Дадим строгие определения этих понятий. Пусть секрет s является случайной величиной на пространстве S , и пусть $P(s)$ — распределение вероятностей для секрета s . Аналогично определяются $s_i \in S_i$ и $P(s_i)$.

Определение 17.1. Потеря энтропии секрета s при наличии частичных секретов из множества A $\{s_i : i \in A\}$ называется величина

$$\Delta_s(s_i : i \in A) = H(s) - H(s|s_i : i \in A),$$

где $H(s)$ — безусловная энтропия; $H(s|s_i : i \in A)$ — условная энтропия случайного секрета s при условии, что зафиксированы частичные секреты $\{s_i : i \in A\}$.

Определение 17.2. СРС называется совершенной относительно распределения вероятностей $P(s)$, если:

- 1) $H(s \in S) \neq 0$;
- 2) $\Delta_s(s_i : i \in A) = 0$ для всех $A \in \bar{\Gamma}$.

Далее через $|s|$ будем обозначать размер в битах секретов $s \in S$, а через $|S|$ — мощность пространства секретов. Если s — просто битовая строка, то $|S| = 2^{|s|}$.

Определение 17.3. СРС называют идеальной, если

$$|s_i| = |s|, \quad i = 1, \dots, t,$$

т. е. размер основного секрета s в битах и размеры всех частичных секретов одинаковы.

Часто в это определение добавляют условие совершенности, т. е. идеальная схема в этом случае автоматически будет и совершенной.

Определение 17.4. Информационным уровнем i -го участника CPC называется величина

$$\rho_i = \frac{\log_2 |S|}{\log_2 |S_i|}.$$

Определение 17.5. Информационным уровнем CPC называется величина

$$\rho = \min\{\rho_i : 1 \leq i \leq t\}.$$

Информационный уровень совершенных идеальных схем равен единице, а в общем случае для совершенных схем верно неравенство $0 < \rho \leq 1$, которое означает, что размер частичного секрета не может быть меньше размера основного секрета. Мы не приводим здесь строгого доказательства этого важного факта, а дадим лишь его набросок.

Пусть, например, $|s_1| < |s|$, и пусть A – минимальное по включению разрешенное множество, включающее участника под номером 1. Множество $A \setminus 1$ будет уже запрещенным, однако его участники могут раскрыть секрет s за $2^{|s_1|}$ попыток, перебирая возможные значения секрета недостающего участника.

Обычно требуют, чтобы распределение секрета s на пространстве S было равномерным. Такое же требование в этом случае выдвигается и в отношении частичных секретов $s_i \in S_i$, $i = 1, 2, \dots, k$. Если частичный секрет s_i зависит от секрета s , $s_i = f(s)$, то возникает вопрос о том, когда распределение частичного секрета будет автоматически равномерным.

Здесь существенным оказывается следующее понятие. Будем называть сюръективное отображение $f : X \rightarrow Y$ сбалансированным (равновероятным), если мощности всех полных прообразов $f^{-1}(y)$, $y \in Y$ одинаковы. Ответ на поставленный вопрос дает следующее легко проверяемое свойство.

Свойство 17.1. Пусть секрет s равномерно распределен на пространстве S , а отображение $f : S \rightarrow S_i$ является сбалансированным. Тогда секрет $s_i = f(s)$ равномерно распределен на пространстве S_i .

17.3. СХЕМА ШАМИРА

В 1979 г. А. Шамир предложил первую схему разделения секрета. Он рассматривал произвольную пороговую структуру доступа. Эта схема оказалась и совершенной, и идеальной.

Пусть \mathbb{F}_q – конечное поле $q > k$. Опишем сначала распределение секретов для участников (t, k) -пороговой структуры доступа.

Сначала дилер СРС случайным образом генерирует секрет $s \in \mathbb{F}_q$ и $t - 1$ случайных величин $f_1, f_2, \dots, f_{t-1} \in \mathbb{F}_q$. Затем составляется многочлен $f(x) = s + f_1x + f_2x^2 + \dots + f_{t-1}x^{t-1}$ и выбираются попарно различные ненулевые элементы поля a_1, a_2, \dots, a_k . В качестве i -го частичного секрета берется $s_i = f(a_i)$. Любая группа, состоящая из t или более участников, сможет восстановить многочлен $f(x)$, воспользовавшись, например, интерполяционной формулой Лагранжа, так как $\deg f(x) \leq t - 1$. Одновременно будет найден и секрет $s = f(0)$.

В связи с этим напомним интерполяционную формулу Лагранжа. Для определенности будем предполагать, что секрет восстанавливают участники с номерами $1, 2, \dots, m$, $m \geq t$.

$$f(x) = \sum_{j=1}^m s_j \frac{(x - a_1) \dots (x - a_{j-1})(x - a_{j+1}) \dots (x - a_m)}{(a_j - a_1) \dots (a_j - a_{j-1})(a_j - a_{j+1}) \dots (a_j - a_m)}.$$

Имеется еще один способ восстановления секрета этими участниками. Для нахождения многочлена $f(x)$ они могут составить систему линейных алгебраических уравнений:

$$\begin{cases} s + f_1a_1 + f_2a_1^2 + \dots + f_{t-1}a_1^{t-1} = s_1, \\ s + f_1a_2 + f_2a_2^2 + \dots + f_{t-1}a_2^{t-1} = s_2, \\ \dots \\ s + f_1a_m + f_2a_m^2 + \dots + f_{t-1}a_m^{t-1} = s_m. \end{cases}$$

Уже первые t уравнений позволяют найти многочлен $f(x)$, так как определитель Вандермонда

$$\begin{vmatrix} 1 & a_1 & a_1^2 & \dots & a_1^{t-1} \\ 1 & a_2 & a_2^2 & \dots & a_2^{t-1} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & a_t & a_t^2 & \dots & a_t^{t-1} \end{vmatrix} = \prod_{i < j} (a_i - a_j)$$

отличен от нуля.

Теорема 17.1. Пороговая (t, k) схема Шамира является и совершенной, и идеальной.

Доказательство. Все секреты s, s_1, s_2, \dots, s_k — элементы поля \mathbb{F}_q , т. е. для их записи требуется одно и то же количество битов.

С другой стороны, $t - 1$ участников не могут получить никакой информации кроме априорной о значении ключа s . В самом деле, для любых $s_1 = f(a_1), s_2 = f(a_2), \dots, s_{t-1} = f(a_{t-1})$, s имеется ровно один многочлен $f(x)$, $\deg f(x) \leq t - 1$, удовлетворяющий условию $s = f(0)$. \square

Схема Шамира обладает несколькими дополнительными существенными свойствами.

1. Она открыта для присоединения новых участников. При этом многочлен $f(x)$, секрет s , частичные секреты s_i , $i = 1, 2, \dots, t$, и порог t остаются прежними. Надо лишь заранее брать достаточно большое поле \mathbb{F}_q .

2. Можно увеличивать контролирующую способность отдельных участников, предоставляя им дополнительные частичные секреты.

3. Схема Шамира не использует недоказанных математических утверждений.

Пример 17.2. В поле \mathbb{F}_{13} выберем секрет $s = 11$ и построим для него $(3, 5)$ -пороговую схему. В качестве f многочлена возьмем $f(x) = 11 + 8x + 7x^2$. Легко находим $s_1 = f(1) = 0$, $s_2 = f(2) = 3$, $s_3 = f(3) = 7$, $s_4 = f(4) = 12$, $s_5 = f(5) = 5$. Все вычисления проведены в поле \mathbb{F}_{13} . Читатель в состоянии проверить, что любые 3 и более участников правильно восстановят секрет $s = 11$.

17.4. ЛИНЕЙНОЕ РАЗДЕЛЕНИЕ СЕКРЕТА

Рассмотрим сейчас наиболее изученный метод разделения секрета, основанный на линейной алгебре. Он дает возможность строить совершенные реализации произвольных структур доступа.

Сначала выбирают r -мерный вектор-столбец s . Он играет роль вспомогательного секрета, и пусть секрет s и его «проекции» (частичные секреты) представляются как конечномерные векторы $s_i = (s_{i1}, \dots, s_{ir})$ и генерируются по формуле $s_i = SH_i$, где H_i – некоторые $r \times m_i$ -матрицы над полем \mathbb{F}_q . Сопоставим каждой матрице H_i линейное подпространство L_i , порожденное ее столбцами (т. е. состоящее из всех линейных комбинаций ее столбцов H_i). Для восстановления секрета участники СРС находят линейную оболочку своих подпространств L_i и ее базис, а затем вычисляют вектор s , решая систему линейных алгебраических уравнений. Под линейной оболочкой пространств L_1, L_2, \dots, L_s понимается их сумма $L_1 + L_2 + \dots + L_s$. Сказанное означает, что матрица, составленная из вектор-столбцов базиса суммы, и будет матрицей системы линейных уравнений. Существует и другой способ. Если A – подмножество участников, то можно решить систему

$$\{s_i = SH_i : i \in A\}.$$

Матрицы H_i стараются подобрать так, чтобы для разрешенного множества A система имела единственное решение.

Несложные рассуждения показывают, что данная конструкция дает совершенную СРС тогда и только тогда, когда семейство линейных подпространств (L_0, L_1, \dots, L_t) конечномерного векторного пространства K^r удовлетворяет свойству все или ничего. А именно, будем говорить, что

семейство подпространств (L_0, L_1, \dots, L_t) конечномерного векторного пространства L над полем K удовлетворяет свойству все или ничего, если для любого множества $A \subset \{1, 2, \dots, t\}$ линейная оболочка подпространств $\{L_a : a \in A\}$ либо содержит подпространство L_0 целиком, либо пересекается с ним только по нулевому вектору. При этом, очевидно, множество A является разрешенным ($A \in \Gamma$), если и только если линейная оболочка подпространств $\{L_a : a \in A\}$ содержит подпространство L_0 целиком. С другой стороны, множество является запрещенным ($A \notin \Gamma$), если и только если линейная оболочка подпространств $\{L_a : a \in A\}$ пересекается с подпространством L_0 только по вектору 0. Отметим, что если бы для некоторого A пересечение L_0 и линейной оболочки $\{L_a : a \in A\}$ было нетривиальным, то участники из A не могли бы восстановить секрет однозначно, но получали бы некоторую информацию о нем, т. е. схема не была бы совершенной.

Частным случаем линейного разделения секрета является схема Блейкли, одного из основоположников теории. Секретом считается точка в n -мерном пространстве. В качестве частичных секретов используются уравнения гиперплоскостей, содержащих эту точку. При специально подобранных уравнениях оказывается возможным реализовать пороговый эффект, т. е. любое заданное число участников будет обладать системой, имеющей единственное решение (секрет).

Приведем пример линейного разделения для одной структуры доступа из книги [9].

Рассмотрим структуру доступа для случая четырех участников, задаваемую $\Gamma_{\min} = \{\{1, 2\}, \{2, 3\}, \{3, 4\}\}$. Она известна как первый пример структуры, для которой не существует совершенной идеальной реализации. Было показано, что для нее $\rho \leq 2/3$. Возьмем поле \mathbb{F}_2 и матрицы

$$H_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}; \quad H_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}; \quad H_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix};$$

$$H_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}; \quad H_4 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Непосредственная проверка показывает, что в этом случае $\rho = 2/3$. Следовательно, линейное разделение секрета дает в этом случае оптимальный результат.

17.5. МОДУЛЯРНЫЙ ПОДХОД

В криптографии традиционно широко используются вычисления в колцах \mathbb{Z}_m . Видимо, этим объясняется популярность следующей (t, k) -схемы. Рассмотрим систему $m_1 < m_2 < \dots < m_k$ попарно взаимно простых модулей, для которой выполнено условие

$$M_2 = m_1 m_2 \dots m_k > m_{k-t+2} m_{k-t+3} \dots m_k = M_1. \quad (17.1)$$

Одновременно требуется, чтобы разность $M_2 - M_1$ была по возможности большой. Секрет s выбирается случайным образом из промежутка (M_1, M_2) , а частичный секрет s_i участника i , $i = 1, 2, \dots, t$, есть наименьший неотрицательный вычет s по модулю m_i .

Предполагается, что каждый участник знает не только s_i , но и модуль m_i . В основе этой схемы лежит утверждение о том, что любая система сравнений

$$\begin{cases} x \equiv s_{i_1} \pmod{m_{i_1}}, \\ x \equiv s_{i_2} \pmod{m_{i_2}}, \\ \dots \\ x \equiv s_{i_s} \pmod{m_{i_s}}. \end{cases} \quad (17.2)$$

имеет единственное решение в промежутке (M_1, M_2) , если $s \geq k$, и имеет достаточно много решений, если $s < k$ и промежуток (M_1, M_2) в некотором смысле велик. Эта схема была предложена М. Миньоттом. О подобных схемах говорят, что они основаны на китайской теореме об остатках.

Приведем пример. Пусть

$$m_1 = 97, m_2 = 98, m_3 = 99, m_4 = 101, m_5 = 103;$$

$$s_1 = 62, s_2 = 4, s_3 = 50, s_4 = 50, s_5 = 38.$$

Условие (17.1) выполнено. Нахождение основного секрета s можно выполнить с помощью китайской теоремы об остатках, $s = 500\,000$.

Основная трудность при построении этих схем заключается в подборе модулей m_1, m_2, \dots, m_t , удовлетворяющих условию (17.1).

Рассмотрим модификацию схемы Миньотта, предложенную К. Асмутом и Дж. Блумом, которая приближает ее к идеальной. Она также является (t, k) -пороговой.

Модули $m_0, m_1 < m_2 < \dots < m_t$ выбираются таким образом, чтобы были выполнены условия:

- 1) НОД $(m_i, m_j) = 1$ для $i \neq j$;
- 2) НОД $(m_0, m_i) = 1, \forall i \in I$;

- 3) $\prod_{i=1}^k m_i > m_0 \prod_{i=1}^{k-1} m_{t-i+1}$.

К. Асмут и Дж. Блум отмечают, что найти такие простые числа, которые удовлетворяют вышеперечисленным требованиям, достаточно легко.

Положим $M_1 = \prod_{i=1}^{k-1} m_{t-i+1}$, $M_2 = \prod_{i=1}^k m_i$. Пусть секрет s выбирается так, что $0 \leq s < m_0$. Пусть также $S = s + pm_0$, где p – произвольное целое из промежутка $[0, M_2/m_0)$. Число S называется *промежуточным секретом*. Тогда в качестве частичного секрета для участника i возьмем $s_i = S \pmod{m_i}$, где справа стоит результат приведения S по модулю m_i . Для восстановления секрета s применяем китайскую теорему об остатках, а затем приводим полученный результат по модулю m_0 . Очевидно, что эта схема отличается от предыдущей лишь случайным множителем p и дополнительным модулем m_0 , по которому приводится секрет. Это незначительное изменение позволяет несколько приравнять «размеры» секрета и частичных секретов, что приближает схему к идеальной.

Интересно отметить, что схема Шамира также является модулярной, но только не в кольце целых чисел, а в кольце многочленов $\mathbb{F}_q[x]$. Здесь существенным обстоятельством является то, что распределение секрета по Шамиру $s_i = f(a_i)$ можно заменить модулярным условием $f(x) = s_i \pmod{(x - a_i)}$, тогда восстановление секрета будет сведено к решению системы сравнений

$$\left\{ \begin{array}{l} f(x) \equiv s_1 \pmod{(x - a_1)} \\ f(x) \equiv s_2 \pmod{(x - a_2)} \\ \dots \\ f(x) \equiv s_l \pmod{(x - a_l)} \end{array} \right., \quad (17.3)$$

если это делают участники $1, 2, \dots, l$. Легко показать, что китайская теорема об остатках справедлива и в кольце $\mathbb{F}_q[x]$, а интерполяционный многочлен Лагранжа является решением системы (17.3).

17.6. ГЕНЕРАЦИЯ МОДУЛЕЙ ДЛЯ ПОРОГОВЫХ СХЕМ В КОЛЬЦЕ ЦЕЛЫХ ЧИСЕЛ

При построении пороговых схем самым важным является вопрос о том, как построить открытые ключи участников, т. е. модули m_1, m_2, \dots, m_k , удовлетворяющие условию (17.1). От этих модулей напрямую зависит качество СРС, т. е. ее близость к идеальной и совершенной СРС. К сожалению, в кольце целых чисел достичь этой цели в полной мере нельзя. Отметим одно простое, но важное свойство модулярных СРС.

Теорема 17.2. *Если попарно взаимно простые модули m_1, m_2, \dots, m_{2t} реализуют $(t, 2t)$ -пороговую схему, то они также реализуют и все $(l, 2t)$, $0 < l \leq 2t$ -пороговые схемы.*

Доказательство. Пусть $m_1 < m_2 < \dots < m_{2t}$. Тогда $m_1m_2 \dots m_t > m_{t+2}m_{t+3} \dots m_{2t}$ влечет за собой $m_1m_2 \dots m_tm_{t+1} > m_{t+1}m_{t+2}m_{t+3} \dots m_{2t}$ и т. д. \square

В этом разделе мы хотим обосновать возможность использования систем последовательных модулей для построения пороговых схем. Это объясняется тем, что при использовании соседних модулей размеры частичных секретов становятся максимально близкими. Отметим также, что условие (17.1) в этом случае необходимо заменить на условие

$$\begin{aligned} \max_{i_1, \dots, i_{t-1}} \text{НОК}(a + i_1, \dots, a + i_{t-1}) &< \\ &< \min_{j_1, \dots, j_t} \text{НОК}(a + j_1, \dots, a + j_t). \end{aligned} \quad (17.4)$$

Лемма 17.1. *Пусть $a, a+1, \dots, a+k-1$ – последовательные натуральные числа и пусть $a+i_1, a+i_2, \dots, a+i_s$ – некоторые $s < k$ из этих чисел. Тогда справедливо неравенство*

$$\begin{aligned} (a+i_1)(a+i_2) \cdot \dots \cdot (a+i_s) &\leqslant \\ &\leqslant (k-1)^{s(s-1)/2} \text{НОК}(a+i_1, a+i_2, \dots, a+i_s). \end{aligned} \quad (17.5)$$

Доказательство. Рассмотрим цепочку неравенств:

$$\begin{aligned} \text{НОК}(a+i_1, \dots, a+i_s) &= \frac{(a+i_1) \cdot \text{НОК}(a+i_2, \dots, a+i_s)}{\text{НОД}(a+i_1, \text{НОК}(a+i_2, \dots, a+i_s))} \geqslant \\ &\geqslant \frac{(a+i_1) \cdot \text{НОК}(a+i_2, \dots, a+i_s)}{\text{НОД}(a+i_1, \prod_{j=2}^s (a+i_j))} \geqslant \frac{(a+i_1) \cdot \text{НОК}(a+i_2, \dots, a+i_s)}{\prod_{j=2}^s \text{НОД}(a+i_1, (a+i_j))} \geqslant \\ &\geqslant \frac{(a+i_1) \cdot \text{НОК}(a+i_2, \dots, a+i_s)}{(k-1)^{s-1}} \geqslant \dots \geqslant \frac{\prod_{j=1}^s (a+i_j)}{(k-1)^{s(s-1)/2}}. \end{aligned}$$

Таким образом, лемма доказана. \square

Теперь укажем, при каких a выполняется неравенство (17.4).

Теорема 17.3. *Для того чтобы выполнялось неравенство (17.4), достаточно число a выбирать из условия*

$$a \geqslant e \cdot (k-1)^{t(t-1)/2}. \quad (17.6)$$

Доказательство. Используя лемму 17.1 и очевидное неравенство

$$\text{НОК}(a + i_1, \dots, a + i_s) \leq (a + i_1) \cdots (a + i_s),$$

усилим условие (17.4) и потребуем выполнения следующего неравенства:

$$(a) \cdots (a + t - 1) \geq (k - 1)^{t(t-1)/2} (a + k - 1) \cdots (a + k - t). \quad (17.7)$$

Тогда очевидно, что $a > (k - 1)^{t(t-1)/2}$. Положим $a = \alpha \cdot (k - 1)^{t(t-1)/2}$. Рассмотрим цепочку очевидных неравенств:

$$\begin{aligned} \frac{(a + k - 1) \cdots (a + k - t)}{(a + 1) \cdots (a + t - 1)} &\leq \left(\frac{a + k - 1}{a} \right)^{t-1} = \\ &= \left(1 + \frac{1}{\alpha(k - 1)^{t(t-1)/2-1}} \right)^{t-1} \leq \left(1 + \frac{1}{t-1} \right)^{t-1} < e. \end{aligned}$$

Таким образом, для всех $\alpha \geq e$ неравенство (17.7) выполняется, а следовательно, верно и неравенство (17.4).

Теперь покажем, что если неравенство (17.7) верно для некоторого a , то оно верно и для $a + 1$. Обозначим:

$$X(a) = a(a + 1) \cdots (a + t - 1);$$

$$Y(a) = (k - 1)^{t(t-1)/2} (a + k - 1)(a + k - 2) \cdots (a + k - t).$$

Рассмотрим отношение

$$\begin{aligned} \frac{X(a+1)}{Y(a+1)} &= \frac{X(a)}{Y(a)} \cdot \left(\frac{(a+t) \cdots (a+k-t)}{a \cdots (a+k)} \right) = \\ &= \frac{X(a)}{Y(a)} \left(1 + \frac{t(k-t)}{a(a+k)} \right) \geq \frac{X(a)}{Y(a)} > 1. \end{aligned}$$

Таким образом, теорема доказана. \square

Мы нашли условие, при котором можно построить схему Миньотта на последовательных модулях. К сожалению, мы не можем сделать то же самое для схемы Асмута – Блума, поскольку требуется выполнение более сильного условия:

$$\begin{aligned} \max_{i_1, \dots, i_{k-1}} \text{НОК}(m_0, a + i_1, \dots, a + i_{t-1}) &< \\ &< \min_{j_1, \dots, j_k} \text{НОК}(a + j_1, \dots, a + j_t). \end{aligned} \quad (17.8)$$

Очевидно, что для выполнения неравенства (17.8) достаточным условием является

$$m_0 < \frac{\min_{j_1, \dots, j_t} \text{НОК}(a + j_1, \dots, a + j_k)}{\max_{i_1, \dots, i_{t-1}} \text{НОК}(a + i_1, \dots, a + i_{k-1})}.$$

Если условие (17.8) не выполнено, то для такой схемы существуют запрещенные множества участников, которым будет известно, что некоторые значения секрета невозможны (вероятность этих значений равна 0). Такая же слабость будет и в случае, если m_0 не взаимно просто с некоторыми модулями участников. То есть в этих условиях нельзя достигнуть даже асимптотической совершенности схемы Асмута – Блума.

Тем не менее отметим ряд преимуществ последовательных модулей перед другими системами модулей. Во-первых, для генерации схемы нам достаточно выбрать a , удовлетворяющее условию (17.6). При этом мы получаем целое семейство подходящих модулей. Во-вторых, мы можем реализовать любую $(1, k), (2, k), \dots, (k, k)$ -пороговую схему при $t = k - 1$ в условиях теоремы 17.3.

17.7. ПОРОГОВЫЕ СХЕМЫ НАД КОЛЬЦОМ МНОГОЧЛЕНОВ

Рассмотрим следующую модулярную схему разделения секрета. Пусть $m_1(x), m_2(x), \dots, m_t(x) \in \mathbb{F}_q[x]$ – попарно взаимно простые модули участников. Тогда неравенство Миньотта (17.1) трансформируется в неравенство для сумм степеней:

$$M_2 = \min_{A \in \Gamma} \sum_{i \in A} \deg m_i(x) > \max_{A \in \bar{\Gamma}} \sum_{j \in A} \deg m_j(x) = M_1. \quad (17.9)$$

Очевидно, что если модули участников имеют одинаковую степень n , то неравенство (17.9) для (k, t) -пороговой схемы выполняется автоматически ($(k - 1)n < kn$). Поэтому интересной является задача оценки максимально-го числа попарно взаимно простых нормированных многочленов одинаковой степени.

Найдем максимальное число попарно взаимно простых нормированных многочленов степени n в кольце $\mathbb{F}_q[x]$. Сюда в первую очередь следует отнести неприводимые. Их число находится по известной формуле [29]:

$$N_q(n) = \frac{1}{n} \sum_{d|n} \mu(d) q^{n/d}, \quad (17.10)$$

где $\mu(d)$ – функция Мёбиуса. Таким образом, остается лишь вычислить количество приводимых попарно взаимно простых многочленов степени n . Нас в первую очередь интересуют максимальные по числу элементов, а не по включению, семейства многочленов с этим условием. Обозначим их число через $C_q(n)$.

Сначала докажем одно вспомогательное утверждение.

Теорема 17.4. *Функция $N_q(n)$ является строго возрастающей по n , кроме случаев $q = 2, 3$, когда она строго возрастает, начиная с $n = 2$.*

Доказательство. Будем исходить из двух очевидных неравенств:

$$N_q(n) \geq \frac{1}{n} \left(q^n - \frac{q^n - q}{q - 1} \right); \quad N_q(n) \leq \frac{1}{n} \left(q^n + \frac{q^n - q}{q - 1} \right).$$

Тогда

$$N_q(n+1) - N_q(n) \geq \frac{1}{n(n+1)(q-1)} (q^{n+1}(nq - 3n - 1) + (2n+1)q) > 0$$

для всякого $q > 3$, так как $nq > 3n + 1$.

Для рассмотрения оставшихся случаев будем использовать более точные оценки:

$$\frac{1}{n} \left(q^n - \frac{q^{[n/2]+1} - q}{q - 1} \right) \leq N_q(n) \leq \frac{1}{n} \left(q^n + \frac{q^{[n/2]+1} - q}{q - 1} \right).$$

Если $q = 2$, то

$$\begin{aligned} N_2(n+1) - N_2(n) &\geq \frac{1}{n(n+1)} ((n-1)2^n - n2^{[\frac{n+1}{2}]+1} - \\ &- (n+1)2^{[n/2]+1} + 2n + 2) > 0, \end{aligned}$$

начиная с $n = 5$. Для $n < 5$ условие теоремы проверяется непосредственно.

Случай $q = 3$ рассматривается аналогично. \square

Теперь мы можем найти значение $C_q(n)$.

$$\text{Теорема 17.5. } C_q(n) = \sum_{i=1}^{[n/2]} N_q(i).$$

Доказательство. Пусть сначала n нечетное. Рассмотрим максимальное по числу элементов семейство приводимых попарно взаимно простых нормированных многочленов степени n . Каждый многочлен указанного семейства содержит делитель, степень которого не превышает $[n/2]$. Поэтому $C_q(n)$ не превышает указанной суммы. Возьмем теперь произвольный многочлен из нашего семейства $f_1(x)f_2(x)$, $\deg f_1(x) < n/2$. Для максимального количества многочленов в данном семействе достаточно, чтобы все дополнительные многочлены $f_2(x)$ были неприводимыми. Поскольку $\deg f_1(x)f_2(x) = n$, $\deg f_1(x) = l \leq [n/2]$, то $\deg f_2(x) = n - l$. Это означает, что для существования дополнительного многочлена $f_2(x)$ достаточно выполнения условия $N_q(l) \leq N_q(n - l)$, и что в этом случае указанная верхняя граница достижима. Но согласно предыдущей теореме это условие выполняется всегда.

В случае четного n отличие состоит в том, что в наше семейство следует включить многочлены вида $f_1(x)^2$, где $f_1(x)$ – неприводимый нормированный многочлен степени $n/2$.

Таким образом, нами получена точная формула для максимального числа попарно взаимно простых нормированных многочленов степени n над полем \mathbb{F}_q :

$$S_q(n) = N_q(n) + C_q(n) = N_q(n) + \sum_{i=1}^{[n/2]} N_q(i). \quad (17.11)$$

Теперь мы можем указать, какие (t, k) -пороговые СРС могут быть реализованы с помощью попарно взаимно простых модулей одинаковой степени.

Теорема 17.6. *Любое семейство попарно взаимно простых многочленов степени n пригодно для реализации любой (t, k) -пороговой структуры доступа в смысле Миньотта и Асмута – Блума с числом участников k таким, что*

$$k < S_q(n).$$

Доказательство. Для всякой (t, k) -пороговой структуры доступа нам понадобится k попарно взаимно простых многочленов для реализации в смысле Миньотта или $k + 1$ многочленов для реализации в смысле Асмута – Блума. Откуда $k < S_q(n)$. \square

Отметим ряд полезных свойств построенной (t, k) -пороговой схемы разделения секрета.

Свойство 17.2. Пусть n – степень модулей участников (t, k) -пороговой СРС, $k < S_q(n) - 1$. Тогда добавление нового участника $k + 1$ к пороговой модулярной схеме разделения секрета в кольце $\mathbb{F}_q[x]$ не потребует изменения уже распределенных частичных секретов и модулей.

Свойство 17.3. Пусть имеется полиномиальная модулярная (t, k) -пороговая СРС, построенная указанным выше способом. Тогда переход к любой (t', k) -пороговой схеме потребует изменения не модулей участников, а лишь их частичных секретов.

В схеме Шамира открытыми параметрами являются точки, в которых вычисляются значения полинома. Поэтому изменение порога приводит к изменению и полинома, что более трудоемко, чем генерация нового промежуточного секрета в модулярной схеме.

Свойство 17.4. Построенные модули (t, k) -пороговой схемы пригодны для реализации схем с различными секретами того же размера.

Отметим, что в схеме Шамира для каждого случая необходимо генерировать новый полином, несмотря на то, что (t, k) прежние, разные лишь секреты. Это так, поскольку участник, имея два частичных секрета при одном и том же полиноме, но с различными свободными членами, получает как минимум разность секретов

$$f_1(i) - f_2(i) = c_1 - c_2.$$

Запрещенное подмножество получает еще больше возможностей для восстановления секретов. В этом смысле модулярный подход более надежен.

Приведем пример. Рассмотрим $(2, 3)$ -пороговую схему разделения секрета по Миньотту (т. е. без дополнительного модуля). В качестве модулей участников выберем 3 попарно взаимно простых многочлена из кольца $\mathbb{F}_2[x]$. Пусть $m_1(x) = x^2$, $m_2(x) = x^2 + 1$, $m_3(x) = x^2 + x + 1$. Степень секрета может быть равна 3. Положим $c(x) = x^3 + 1$. Тогда частичными секретами будут многочлены $s_1(x) = 1$, $s_2(x) = x + 1$ и $s_3(x) = 0$ соответственно.

17.8. СОВЕРШЕННЫЕ МОДУЛЯРНЫЕ СХЕМЫ

Главными характеристиками СРС являются совершенность и идеальность. Рассмотрим случай схемы Асмута – Блума в кольце $\mathbb{F}_q[x]$ с дополнительным модулем $m_0(x)$, $\deg m_0(x) \leq \deg m_i(x)$, $\forall i \in I$. Необходимым и достаточным условием модулярной реализации структуры доступа Γ является выполнение неравенства

$$M_2 = \min_{A \in \Gamma} \deg \text{НОК}(m_i(x), i \in A) > \max_{B \in \Gamma} \deg \text{НОК}(m_j(x), j \in B) = M_1.$$

Секрет $s(x)$ случайным образом выбирается на множестве многочленов, степень которых меньше $\deg m_0(x)$. Будем считать, что секрет $s(x)$ равномерно распределен на указанном множестве. Затем случайным образом генерируется многочлен $p(x)$, степень которого меньше $M_2 - \deg m_0(x)$. Он является дополнительным секретом и не известен ни одному из участников. В результате формируется промежуточный секрет $S(x) = m_0(x)p(x) + s(x)$, $\deg S(x) < M_2$. Отметим, что все полиномы $S(x)$ равновероятны. Частичные секреты вычисляются по формуле $s_i(x) = S(x) \pmod{m_i(x)}$. Найдем условия, при которых построенная схема Асмута – Блума будет совершенной.

Теорема 17.7. *Реализация схемы Асмута – Блума в кольце многочленов $\mathbb{F}_q[x]$ совершенна тогда и только тогда, когда:*

- 1) $\text{НОД}(m_0(x), m_i(x)) = 1$, $\forall i \in I$;
- 2) $\deg m_0(x) \leq M_2 - M_1$.

Доказательство. Необходимость первого условия очевидна, так как в противном случае участник i , вычислив $\text{НОД}(m_0(x), m_i(x)) = d_i(x)$, найдет вычет секрета $s(x)$ по этому модулю:

$$r_i(x) = s_i(x) \pmod{d_i(x)}.$$

Тем самым уменьшается перебор значений секрета, который может быть представлен в виде $s(x) = d_i(x)g(x) + r_i(x)$. Остается перебрать многочлены $g(x)$, степень которых меньше степени $s(x)$.

Докажем необходимость второго условия. Рассмотрим запрещенное подмножество участников A , степень наименьшего кратного модулей которого равна M_1 . Считаем, что первое необходимое условие выполнено, а второе – нет. Рассмотрим систему сравнений

$$\begin{cases} S(x) \equiv s_A(x) \pmod{m_A(x)}, \\ S(x) \equiv \bar{s}(x) \pmod{m_0(x)}, \end{cases}$$

где $(s_A(x), m_A(x))$ – общие частичный секрет и модуль соответственно, полученные участниками из подмножества A , $\bar{s}(x)$ – произвольное возможное значение секрета.

Поскольку $\text{НОД}(m_0(x), m_A(x)) = 1$, то решение данной системы единственно по модулю $m_A(x)m_0(x)$. Выберем $\bar{s}(x) = (m_A(x)x^{M_2-M_1} + s_A(x)) \pmod{m_0(x)}$. Очевидно, что решением системы будет $S(x) = m_A(x)x^{M_2-M_1} + s_A(x)$, так как $\deg S(x) = M_2 < \deg m_0(x)m_A(x)$. Следовательно, полином $\bar{c}(x)$ не может быть секретом, а значит, $H(s(x)|s_i(x) : i \in A) < H(s(x))$. Получили противоречие.

Обратно, пусть построенная реализация схемы Асмута – Блума удовлетворяет условиям теоремы. Во-первых, $M_2 \geq M_1 + \deg m_0(x) \geq 2\deg m_0(x)$. Тогда $H(p(x)) \geq H(s(x))$. Во-вторых, покажем, что значение секрета остается равномерно распределенным, когда известны частичные секреты запрещенного подмножества участников.

Мощность множества возможных секретов равна $q^{\deg m_0(x)}$. Энтропия секрета при равномерном распределении находится по формуле

$$H(c(x)) = \log q^{\deg m_0(x)} = \deg m_0(x) \log q.$$

Покажем, что распределение остается равномерным и при известном наборе частичных секретов запрещенного подмножества участников $A \in \bar{\Gamma}$. Положим $d_A = \deg \text{НОК}(m_i(x), i \in A)$. Имеется $q^{M_2-d_A}$ полиномов $y(x)$ таких, что $y(x) \equiv s_A(x) \pmod{m_A(x)}$ и $\deg y(x) \leq M_2 - 1$. Обозначим множество этих многочленов через Y .

Далее, в множестве Y имеется в точности $q^{M_2-d_A-\deg m_0(x)}$ многочленов, удовлетворяющих сравнению $y(x) \equiv \bar{s}(x) \pmod{m_0(x)}$ для фиксированного $\bar{s}(x) \in S$. Следовательно,

$$P(s(x) = \bar{s}(x) | s_i(x) : i \in A) = \frac{q^{M_2-d_A-\deg m_0(x)}}{q^{M_2-d_A}} = \frac{1}{q^{\deg m_0(x)}}.$$

По формуле для условной энтропии $H(s(x)|s_i(x) : i \in A) = H(s(x))$. \square

Следствие 17.1. *Любая структура доступа имеет совершенную реализацию в смысле Асмута – Блума в кольце $\mathbb{F}_q[x]$.*

Следствие 17.2. *Максимальный информационный уровень совершенной схемы Асмута – Блума в кольце полиномов над полем Галуа достигается, если $\deg m_0(x) = M_2 - M_1$. В этом случае он вычисляется по формуле*

$$\rho = \frac{M_2 - M_1}{\max_{i \in I} \deg m_i(x)}. \quad (17.12)$$

Следствие 17.3. *Построенная в разделе 17.7 (t, k) -пороговая схема разделения секрета является совершенной и идеальной.*

Доказательство. Действительно, $M_2 - M_1 = n$, $n = \deg m_i(x) = \deg m_0(x)$, $|S| = q^n = |S_i|$. \square

Таким образом, построена совершенная и идеальная пороговая схема в рамках модулярного подхода. Теперь охарактеризуем все структуры доступа, которые могут быть реализованы идеально в кольце $\mathbb{F}_q[x]$.

Определение 17.6. *Два участника – i и j – из множества I называются взаимозаменяемыми, если для любого подмножества $A \in \bar{\Gamma}_{\max}$ справедливо:*

$$i \in A \Leftrightarrow j \in A.$$

Взаимозаменяемым участникам можно давать одинаковые модули и, соответственно, частичные секреты. Поэтому без потери общности можно рассматривать ту же структуру доступа, заменив этих участников одним участником. Верно и обратное: в любую структуру доступа можно добавить участника, который будет взаимозаменяемым с каким-то из уже присутствующих, без изменения модулей и частичных секретов остальных участников. Далее рассматриваем структуры доступа, в которых отсутствуют взаимозаменяемые участники.

Теорема 17.8. *Идеальной модулярной реализацией в кольце полиномов над полем Галуа обладает только пороговая структура доступа.*

Доказательство. Предположим, что у нас есть идеальная реализация Асмута – Блума. В этом случае модули участников и дополнительный модуль $m_0(x)$ имеют одинаковую степень n . Считаем, что в структуре доступа отсутствуют взаимозаменяемые участники. В противном случае заменим их одним участником.

Рассмотрим произвольное запрещенное подмножество $A \in \bar{\Gamma}_{\max}$. Для него справедливо

$$i \notin A \Rightarrow A \cup \{i\} \in \Gamma.$$

Тогда, с одной стороны, очевидно, что $M_2 - M_1 \leq n$. Но по теореме 17.7 и сделанному предположению об идеальности следует, что $M_2 - M_1 = n$. Поэтому $\text{НОД}(m_i(x), m_j(x)) = 1$ для всех $j \in A$ и $i \notin A$. Это справедливо для всех запрещенных подмножеств $A \in \bar{\Gamma}_{\max}$.

Осталось показать, что $\text{НОД}(m_i(x), m_j(x)) = 1$ для любой пары $i, j \in I$. Это вместе с равенством степеней гарантирует, что структура доступа – пороговая. Для этого достаточно заметить, что из отсутствия взаимозаменяемых участников следует $\forall i, j \in I, \exists A \in \bar{\Gamma}_{\max} : i \in A, j \notin A$.

Таким образом, $\text{НОД}(m_i(x), m_j(x)) = 1$ для любой пары $i, j \in I$ и степени всех модулей одинаковы. \square

17.9. МОДУЛЯРНАЯ РЕАЛИЗАЦИЯ ПРОИЗВОЛЬНЫХ СТРУКТУР ДОСТУПА

Сейчас мы покажем, что произвольная структура доступа допускает модулярную реализацию в кольце многочленов от одной переменной x над любым конечным полем. Соответствующая теорема была доказана в работе [96]. Это утверждение легко обобщается на случай любого евклидова кольца.

Теорема 17.9. *Произвольная структура доступа может быть реализована модулярно в кольце многочленов $\mathbb{F}_q[x]$ как в смысле Миньотта, так и в смысле Асмута – Блума.*

Доказательство. Нам необходимо реализовать структуру доступа Γ по Миньотту.

На первом шаге алгоритма $m_1(x) = m_2(x) = \dots = m_t(x) = 1$. Возьмем затем какое-нибудь максимальное по включению запрещенное подмножество $A \notin \Gamma$. Все модули, не входящие в A , домножим на неприводимый многочлен $p_1(x) \in \mathbb{F}_q[x]$. В результате такого домножения окажется, что степень НОК многочленов – участников из множества A меньше степени НОК многочленов – участников из любого разрешенного множества. Это условие не нарушается и в дальнейшем, если будет взят другой неприводимый многочлен $p_2(x) \neq p_1(x)$ и другое максимальное по включению запрещенное подмножество. Всего таких домножений потребуется не более количества всех максимальных запрещенных подмножеств.

Секрет $s(x)$ выбирается так, чтобы

$$M_1 \leq \deg s(x) < M_2,$$

где M_1 – максимальная степень НОК многочленов запрещенных подмножеств; M_2 – минимальная степень НОК многочленов разрешенных подмножеств.

Для реализации в смысле Асмута – Блума требуется дополнительный неприводимый многочлен $m_0(x)$ и выполнение неравенства

$$\deg m_0(x)M_1 < \deg Y(x) < M_2.$$

С этой целью первоначально модули выбранного запрещенного подмножества домножаются на такой многочлен p_1 , что $\deg p_1(x) > \deg m_0(x)$. Далее алгоритм работает так же, как в случае схемы Миньотта. \square

Замечание 17.1. С помощью небольшой модификации можно доказать, что теорема 17.9 остается справедливой при переходе к любому евклидову кольцу. При этом необходимо лишь заменить неравенства вида $\deg f(x) < \deg g(x)$ на неравенства вида $\nu(a) < \nu(b)$, где ν – евклидова норма.

Пример 17.3. Рассмотрим следующую структуру отказа для четырех участников:

$$\bar{\Gamma} = \{1, 2\}, \{2, 3\}, \{1, 3, 4\}.$$

Для реализации схемы Миньотта нам потребуется три неприводимых многочлена:

$$p_1 = x^3 + x + 1, \quad p_2 = x^3 + x^2 + 1, \quad p_3 = x^2 + x + 1.$$

Модулями участников будут многочлены

$$m_1 = x^3 + x^2 + 1, \quad m_2 = x^2 + x + 1, \quad m_3 = x^3 + x + 1, \quad m_4 = (x^3 + x^2 + 1)(x^3 + x + 1).$$

Легко проверить что НОК многочленов всякого разрешенного подмножества равен

$$(x^3 + x^2 + 1)(x^3 + x + 1)(x^2 + x + 1).$$

Для реализации в смысле Асмута – Блума необходимо выбрать дополнительный модуль степени меньше 3, например $m_0 = x^2$.

17.10. РАЗДЕЛЕНИЕ СЕКРЕТА В КОЛЬЦЕ МНОГОЧЛЕНОВ ОТ НЕСКОЛЬКИХ ПЕРЕМЕННЫХ

Рассмотрим обобщение предыдущих результатов на случай кольца многочленов от нескольких переменных $\mathbb{F}_q[X]$, $X = (x_1, x_2, \dots, x_m)$ над полем \mathbb{F}_q . Необходимым условием для такого обобщения является наличие развитой теории идеалов и аналога китайской теоремы об остатках в данном кольце. В соответствии с методом базисов Гребнера можно корректно определить приведение произвольного многочлена (секрета) $s(X)$ по модулю любого идеала [24].

В кольце $\mathbb{F}_q[X]$ остатки и сравнения берутся по модулю идеала, иными словами, $f(X) \equiv r(X) (\text{mod } I)$ означает, что f приводится к r по модулю идеала I , т. е. $f - r \in I$.

Любому моному $X^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \dots x_m^{\alpha_m}$ из $\mathbb{F}_q[X]$ можно поставить в соответствие m -вектор степеней $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m) \in \mathbb{Z}_{\geq 0}^m$. Для однозначного приведения многочлена по модулю идеала необходимо выбрать мономиальное упорядочение. Результатом приведения является остаток от деления многочлена на соответствующий базис Гребнера [24, 70].

Мономиальным упорядочением на $\mathbb{F}_q[X]$ называется бинарное отношение $>$ на $\mathbb{Z}_{\geq 0}^m$ такое, что:

- оно является линейным;
- если $\alpha > \beta, \gamma \in \mathbb{Z}_{\geq 0}^m$, то $\alpha + \gamma > \beta + \gamma$;
- в любом непустом подмножестве $\mathbb{Z}_{\geq 0}^m$ имеется минимальный элемент.

Алгоритм Бухбергера позволяет построить базис Гребнера $G_I = (g_1, g_2, \dots, g_k)$ относительно данного мономиального упорядочения для всякого идеала I кольца $\mathbb{F}_q[X]$. Поскольку всякий полином f приводится по модулю I с помощью базиса G_I единственным образом, то идеалы удобно задавать их базисами Гребнера.

Идеалы $I_1, I_2 \subset \mathbb{F}_q[X]$ называются *взаимно простыми* или *комаксимальными*, если $I_1 + I_2 = (I_1, I_2) = 1$, где (I_1, I_2) – НОД идеалов. Отметим, что для взаимно простых идеалов их пересечение совпадает с произведением $I_1 I_2 = I_1 \cap I_2$.

Понятие *степени идеала* обобщает понятие степени полинома. Она определяется как размерность фактор-кольца $\mathbb{F}_q[X]/I$ как векторного пространства над полем \mathbb{F}_q , т. е. $\deg I = \dim_{\mathbb{F}_q} \mathbb{F}_q[X]/I$.

Мультистепенью multideg f полинома f называется максимальный вектор степеней его мономов относительно выбранного упорядочения, а старшим членом $LT(f)$ называется член, соответствующий моному с максимальной степенью.

В следующей теореме сформулирован один из вариантов CRT в кольце $\mathbb{F}_q[X]$, который используется для алгоритма восстановления секрета.

Теорема 17.10. *Пусть задано мономиальное упорядочение. Система сравнений*

$$\left\{ \begin{array}{l} f(X) \equiv s_1(X) \pmod{I_1}, \\ f(X) \equiv s_2(X) \pmod{I_2}, \\ \dots \\ f(X) \equiv s_k(X) \pmod{I_k} \end{array} \right. \quad (17.13)$$

при попарно взаимно простых идеалах I_1, \dots, I_k имеет единственное решение по модулю произведения $I_1 \dots I_k$ относительно данного мономиального упорядочения. Оно представимо в виде

$$F(X) \equiv \sum_{i=1}^k s_i(X) b_i(X) \pmod{I_1 \dots I_k},$$

где $b_i(X) \equiv 1 \pmod{I_i}$, $b_i(X) \equiv 0 \pmod{I_j}$, $i \neq j$.

Замечание 17.2. Указанное решение строится с использованием базисов Гребнера идеалов I_1, \dots, I_k . В самом деле, минимальный базис Гребнера суммы идеалов (I_i, I_j) содержит 1, а приведение объединенного базиса пары

взаимно простых идеалов к 1 позволяют указать такие $a_i(X) \in I_i$, $a_j(X) \in I_j$, что $a_i(X) + a_j(X) = 1$. Далее, $b_i(X) \equiv \prod_{j=1}^k a_j(X)$, $i \neq j$. Полученное решение $F(X)$ является единственным по модулю произведения всех идеалов $I_1 I_2 \dots I_k$ относительно данного мономиального упорядочения, поэтому *CRT*-алгоритм применим для модулярного восстановления секрета в кольце $\mathbb{F}_q[X]$.

17.11. РЕАЛИЗАЦИЯ ПРОИЗВОЛЬНЫХ СТРУКТУР ДОСТУПА

Сейчас мы займемся реализацией произвольных структур доступа в кольце $\mathbb{F}_q[X]$. Для схемы Миньотта необходимо, чтобы модули участников I_1, I_2, \dots, I_t и секрет $s(X) \in \mathbb{F}_q[X]$ удовлетворяли условию: секрет $s(X)$ должен быть приведен по модулю произведения идеалов всех разрешенных подмножеств и не являться таковым для запрещенных. Для всякого идеала $I \in \mathbb{F}_q[X]$ приведенными полиномами являются лишь линейные комбинации мономов из $RT(I)$ [24].

Модулярная схема Асмута – Блума отличается от приведенной конструкции лишь тем, что секрет выбирается из множества приведенных мономов по модулю дополнительного идеала P_0 .

Теорема 17.11. *Любая структура доступа Γ имеет модулярные реализации в любом кольце $\mathbb{F}_q[X]$ по Миньотту и Асмуту – Блуму.*

Доказательство. Рассмотрим случай реализации по Миньотту. Пусть P_0, P_1, \dots, P_l – попарно взаимно простые радикальные нульмерные идеалы одной и той же степени, где l – число максимальных по включению запрещенных подмножеств. Первоначально присвоим каждому участнику единичный идеал. Берем затем какое-нибудь максимальное по включению запрещенное множество B и модули всех участников, не входящих в B , умножаем на идеал P_1 .

Поступаем так с каждым максимальным запрещенным подмножеством. В результате всех умножений имеем следующее.

Для всякого разрешенного множества участников A пересечение (произведение) всех их идеалов будет равно произведению $P_1 P_2 \dots P_l$, а для всякого запрещенного B соответствующее пересечение будет собственным делителем этого произведения. Игнорируя индексы, можно сказать, что $\bigcap_{i \in B} P_i = P_1 P_2 \dots P_{l_1}$, где $l_1 < l$. Следовательно,

$$RT(P_1 P_2 \dots P_{l_1}) \subset RT(P_1 P_2 \dots P_l).$$

Для каждого максимального запрещенного множества B выберем по одному моному $s_B(X)$ из дополнения $RT(P_1 P_2 \dots P_l) \setminus RT(P_1 P_2 \dots P_{l-1})$, а в качестве самого секрета возьмем какую-нибудь линейную комбинацию мономов $s_B(X)$.

Таким образом, этот полином будет приведенным по $\text{mod} \left(\bigcap_{i \in A} P_i \right)$ и неприведенным по $\text{mod} \left(\bigcap_{i \in B} P_i \right)$.

Для случая схемы Асмута – Блума построенный секрет $S(X)$ считается вспомогательным. В качестве разделяемого секрета берется $s(X) = S(X) \text{ mod } P_0$. \square

Замечание 17.3. Если один из участников входит во все максимальные запрещенные множества, то он не получит никакого дополнительного модуля P_i и не будет обладать никакой дополнительной информацией о секрете. Один из мономов $s_B(X)$ может подходить для нескольких запрещенных множеств B . Поэтому общее число мономов секрета $s(X)$ не превосходит l .

Пример 17.4. Рассмотрим следующую структуру отказа для четырех участников:

$$\bar{\Gamma} = \{1, 2\}, \{2, 3\}, \{1, 3, 4\}.$$

Для реализации схемы по Миньотту нам потребуется три попарно взаимно простых идеала:

$$I_1 = \{x_1, x_2^3 + x_2 + 1\}, I_2 = \{x_1, x_2^3 + x_2^2 + 1\}, I_3 = \{x_1, x_2^3 + x_2\}.$$

Модули участников будут следующими:

$$m_1 = I_2, m_2 = I_3, m_3 = I_1, m_4 = I_1 I_2.$$

Легко проверить что общий модуль разрешенных подмножеств равен $I_1 I_2 I_3$. Для реализации в смысле Асмута – Блума необходимо выбрать дополнительный модуль.

17.12. МАКСИМАЛЬНЫЕ ИДЕАЛЫ ОДИНАКОВЫХ СТЕПЕНЕЙ

Вычислим количество максимальных идеалов одной и той же степени в кольце $\mathbb{F}_q[X]$. Их можно использовать при реализации общих структур доступа, как это сделано в предыдущем параграфе. Одновременно полученная формула обобщает классическую формулу для числа неприводимых многочленов данной степени в кольце $\mathbb{F}_q[x]$.

Теорема 17.12. Пусть I_1, I_2 – нульмерны и взаимно просты. Тогда $\deg I_1 I_2 = \deg I_1 + \deg I_2$.

Доказательство. Наше утверждение следует из известного варианта CRT:

$$\mathbb{F}_q[X]/I_1 I_2 \cong \mathbb{F}_q[X]/I_1 \oplus \mathbb{F}_q[X]/I_2.$$

В самом деле, все факторкольца являются еще и векторными пространствами над полем \mathbb{F}_q . \square

Обозначим через $\overline{N}_q(n)$ число максимальных идеалов степени n в кольце $\mathbb{F}_q[X]$.

Теорема 17.13.

$$\overline{N}_q(n) = N_{q^n}(n). \quad (17.14)$$

Доказательство. В классическом случае для многочленов одной переменной формула для $N_q(n)$ выводится из того, что $q^n = \deg \prod_{\deg f|n} f(x)$, где произведение распространено на все неприводимые полиномы $f(x)$, $\deg(f(x))|n$ [29].

Докажем аналог этой формулы для идеалов в кольце $\mathbb{F}_q[X]$, т. е. что $\deg \prod_{\deg I|n} I = q^{mn}$, а затем применим обращение Мёбиуса. С учетом теоремы

17.12 достаточно показать, что нули всех максимальных идеалов с условием $\deg I|n$ заполняют пространство $\mathbb{F}_{q^n}^m$ однократно без пересечений. В самом деле, у максимального идеала I в точности $\deg I$ нулей, а в силу попарной взаимной простоты степени таких идеалов складываются при умножении.

Если $\deg I = n$, то

$$\mathbb{F}_q[X]/I \cong \mathbb{F}_q[\alpha_1, \alpha_2, \dots, \alpha_m] = \mathbb{F}_{q^n},$$

где $\alpha_1, \alpha_2, \dots, \alpha_m$ – вычеты переменных по модулю идеала. Это следует из классической теоремы 3.22 о гомоморфизмах колец. Все нули идеала I получаются с помощью автоморфизма Фробениуса

$$\begin{aligned} (\alpha_1, \alpha_2, \dots, \alpha_m) &\rightarrow (\alpha_1^q, \alpha_2^q, \dots, \alpha_m^q) \rightarrow \dots \rightarrow \\ &\rightarrow (\alpha_1^{q^{n-1}}, \alpha_2^{q^{n-1}}, \dots, \alpha_m^{q^{n-1}}). \end{aligned}$$

В цепочке нет повторений. Противное означало бы, что

$$\mathbb{F}_q[\alpha_1, \alpha_2, \dots, \alpha_m] = \mathbb{F}_q[\alpha_1^{q^k}, \alpha_2^{q^k}, \dots, \alpha_m^{q^k}], \text{ где } k < n - 1,$$

а значит, поле $\mathbb{F}_q(\alpha_1, \dots, \alpha_m) = \mathbb{F}_{q^k}$ было бы собственным подполем \mathbb{F}_q^n , т. е. $\deg I < n$. Таким образом, с одной стороны, на каждый максимальный идеал I , $\deg I = n$ приходится в точности n различных элементов пространства $\mathbb{F}_{q^n}^m$.

С другой стороны, взаимно простые идеалы не имеют общих нулей. Остается только заметить, что любой вектор $(\alpha_1, \alpha_2, \dots, \alpha_m) \in \mathbb{F}_{q^n}^m$ является нулем ядра гомоморфизма специализации $\mathbb{F}_q[x_1, \dots, x_m] \rightarrow \mathbb{F}_q[\alpha_1, \alpha_2, \dots, \alpha_m]$. В самом деле, поскольку образом гомоморфизма является поле $\mathbb{F}_q[\alpha_1, \alpha_2, \dots, \alpha_m]$, то его ядро — максимальный идеал. \square

17.13. НУЛЬМЕРНЫЕ РАДИКАЛЬНЫЕ ИДЕАЛЫ

На основе полученной формулы для числа максимальных идеалов можно подсчитать $\bar{C}_q(n)$ — максимальное число попарно взаимно простых нульмерных радикальных идеалов одной степени n в кольце $\mathbb{F}_q[X]$. Именно они используются в п. 17.11.

Теорема 17.14.

$$\bar{C}_q(n) = \sum_{l \leq \frac{n}{2}} N_{q^n}(l) + N_{q^n}(n)$$

при нечетном n .

$$\bar{C}_q(m) = \sum_{l \leq \frac{m-2}{2}} N_{q^m}(l) + \left[\frac{1}{2} N_{q^m} \left(\frac{m}{2} \right) \right] + N_{q^m}(m)$$

при четном n .

Доказательство. В случае $m = 1$ это утверждение доказано нами в предыдущей главе. В общем случае, рассуждения легко обобщаются. В самом деле, рассмотрим радикальный нульмерный идеал I , $\deg I = m$, где m — нечетно. Любой радикальный идеал I обладает примерной компонентой I_1 , которая является максимальным идеалом. Пусть $l = \deg I_1 \leq (m-1)/2$. В предыдущей главе мы показали что для всякого q и $l \leq (m-1)/2$ выполняется условие $N_q(l) \leq N_q(m-l)$. Следовательно, $\bar{N}_q(l) \leq \bar{N}_q(m-l)$. Поэтому существует дополнительный идеал I_2 такой, что $\deg I_1 I_2 = m$.

Теперь рассмотрим радикальный нульмерный идеал I , $\deg I = m$, где m — четно. В этом случае дополнительный идеал I_2 для всякого такого I_1 , $\deg I_1 = m/2$, обладает той же степенью $m/2$. Идеалы I_1 , I_2 — различны, поскольку I является радикальным. Следовательно, число таких идеалов с делителями степени $m/2$ в максимальном семействе равно $[1/2 \bar{N}_q(m/2)]$. \square

Соответствующие идеалы пригодны для построения общих схем разделения секрета в кольце $\mathbb{F}_q[X]$. Тем самым получена теорема об оценке числа участников в общей и пороговой схеме.

Теорема 17.15. Пусть имеется максимальное по числу элементов семейство нульмерных радикальных идеалов степени n . Тогда данные идеалы пригодны для реализации $t < \overline{C}_q(n)$ структур доступа с числом участников t таким, что $t < \log_2 C_q(n)$.

Доказательство. Чтобы реализовать произвольную структуру доступа согласно теореме 17.11, нам потребуется не более чем $l + 1$ модулей, где l – количество максимальных запрещенных подмножеств в данной структуре. Очевидно, что $s \leq 2^t$, откуда и следует утверждение теоремы. \square

17.14. ОБ ИДЕАЛЬНЫХ СХЕМАХ В КОЛЬЦЕ МНОГОЧЛЕНОВ ОТ НЕСКОЛЬКИХ ПЕРЕМЕННЫХ

Исследуем схему Асмута – Блума в кольце $\mathbb{F}_q[X]$. Обозначим через $RT(I)$ множество мономов, приведенных по модулю I , а через $RP(I)$ – линейную оболочку $RT(I)$. Пусть также $I^B = \text{НОК}[I_i, i \in B]$, а $M_2 = \bigcap_{B \in \Gamma} RT(I^B)$ – множество мономов, лежащих в пересечении RT идеалов всех разрешенных подмножеств. Отметим, что промежуточный секрет $C(X) \in RP(M_2)$.

Теорема 17.16. Схема Асмута – Блума в кольце $\mathbb{F}_q[X]$ совершенна тогда и только тогда, когда выполнены следующие условия:

- 1) $I_0 + I_i = 1, \forall i \in J$;
- 2) $\forall A \notin \Gamma : RT(I^A I_0) \subseteq M_2$.

Доказательство. Сначала докажем необходимость. Пусть есть совершенная схема Асмута – Блума, но первое условие теоремы не выполнено, т. е. существует $I_i : I_i + I_0 = D \neq 1$. Тогда множество возможных значений секрета для такого участника можно сузить: $c(X) = s_i(X) \bmod D$. Следовательно, схема несовершена: получили противоречие.

Пусть первое условие выполнено, но не выполнено второе, т. е. существует запрещенное подмножество A такое, что $RT(I^A I_0) \not\subseteq M_2$. Иными словами, существует моном $m \in RT(I^A I_0) \setminus M_2$. Рассмотрим многочлен $g(X) = s_A(X) + (m - m \bmod I^A) = s^A(X) + f_m(X)$, где $s^A(X)$ – общий частичный секрет, восстановленный участниками из подмножества A . Заметим, что многочлен $f_m(X) \in I^A$, $f_m(X) \in P(I^A I_0)$ и содержит моном m . Следовательно, $g(X) \in RP(I^A I_0)$. Положим $c' = g(X) \bmod I_0$. Согласно CRT для системы

$$\begin{cases} C(X) \equiv s^A(X) \bmod I^A, \\ C(X) \equiv c'(X) \bmod I_0 \end{cases}$$

существует единственное решение в $RP(I^A I_0)$, но по построению этим решением является многочлен $g(X)$. С другой стороны, $m \in g(X) \notin$

$\notin RP(M_2)$, т. е. значение $c'(X)$ для секрета невозможно: опять получили противоречие.

Докажем достаточность. Пусть условия теоремы выполнены. Покажем, что секрет остается равномерно распределенным и при наличии частичных секретов из запрещенного подмножества. Рассмотрим произвольное запрещенное подмножество $A \in \Gamma$ и множество многочленов $V = \{s^A(X) + f(X) | f(X) \in I^A, f(X) \in RP(M_2)\}$ – множество возможных значений промежуточного секрета. Зафиксируем некоторое значение секрета $c^j(X) \in RP(I_0)$. Тогда существует единственный многочлен $g^j(X) \in RP(I^A I_0)$ такой, что $g_j(X) \equiv s^A(X) \pmod{I^A}, g^j(X) \equiv c^j(X) \pmod{I_0}$ (согласно CRT).

Если $RT(I^A I_0) = M_2$, то каждому значению секрета соответствует единственный промежуточный секрет из множества V , т. е. секрет остается равномерно распределенным при наличии частичных секретов из подмножества A .

Пусть $RT(I^A I_0) \subset M_2$. Каждому многочлену $f(X) \in RP(M_2)$, содержащему хотя бы один моном из $M_2 \setminus RT(I^A I_0)$, поставим в соответствие многочлен $\bar{f}(X) = f(X) - f(X) \pmod{I^A I_0} \neq 0$. Очевидно, что $\bar{f}(X) \in I^A I_0$. Тогда каждому значению секрета $c^j(X) \in RP(I_0)$ соответствует множество промежуточных секретов

$$\begin{aligned} C^j &= \{g^j(X), g^j(X) + \bar{f}(X) | \bar{f}(X) \in I^A I_0, \\ &\quad \bar{f}(X) \in RP(M_2)\} \subset V. \end{aligned}$$

Очевидно, что множества C^j равномощные. Следовательно, в множестве V для каждого значения секрета $s(X)$ существует одинаковое число возможных промежуточных значений $s(X) \in V$, что влечет равномерное распределение $s(X)$ и при наличии частичных секретов из запрещенного подмножества. \square

Два участника называются взаимозаменяемыми, если для любого максимального запрещенного множества A верно $i \in A \Leftrightarrow j \in A$. Далее рассматриваем структуры доступа, в которых отсутствуют взаимозаменяемые участники.

Теорема 17.17. *Идеальной модулярной реализацией в кольце $\mathbb{F}_q[X]$ обладает только пороговая структура доступа.*

Доказательство. Пусть есть совершенная и идеальная схема Асмута – Блума. Это значит, что выполнены условия теоремы 17.16 и, кроме того, $\deg I_i = \deg I_j = \deg I_0, \forall i, j$. Покажем, что $I_i + I_j = 1, \forall i \neq j$, откуда будет следовать, что структура доступа – пороговая, так как в этом случае $\deg I_i I_j = \deg I_i + \deg I_j$ и выполняется второе условие теоремы 17.16.

Рассмотрим произвольное максимальное по включению запрещенное подмножество участников $A \in \bar{\Gamma}_{\max}$, т. е. $\forall i \notin A \Rightarrow A \cup \{i\} \in \Gamma$. Тогда, с одной

стороны, $\deg \text{HOK}[I^A I_i] \geq |M_2|$, так как $M_2 \subseteq RT(\text{HOK}[I^A I_i])$. С другой стороны, $RT(I^A I_0) \subseteq M_2$, $\deg I^A + \deg I_0 \leq |M_2|$ и $\deg I_i = \deg I_0$, откуда следует, что $I^A + I_i = 1$, $RT(I^A I_i) = RT(I^A I_0) = M_2$. Таким образом, модуль участника, не входящего в максимальное запрещенное подмножество, взаимно прост с модулями участников из этого подмножества. Заметим также, что из-за отсутствия в структуре доступа взаимозаменяемых участников выполняется $\forall i, j \in J, \exists A \in \bar{\Gamma}_{\max} : i \in A, j \notin A$. \square

17.15. ЗАДАНИЯ

1. Описать все структуры доступа при $k = 2, 3, 4, 5$. Указать базисы структур доступа и отказа.
2. Построить модулярные реализации всех непороговых структур доступа при числе участников $k = 4$.
3. Построить какие-нибудь СРС, используя доступный аппарат алгебры и теории чисел.
4. Показать, что схема анонимного согласия является и совершенной, и идеальной.
5. Построить системы подходящих модулей в промежутке от 100 до 500 для схем Миньотта и Асмута – Блума для $(3, 5)$ -структур доступа.
6. Рассмотрим $(3, 5)$ -схему Шамира в поле \mathbb{F}_{17} , причем $s_1 = f(1) = 8$, $s_2 = f(2) = 75$, $s_3 = f(3) = 10$. Требуется определить секрет s .
7. Проиллюстрировать на конкретном примере, как в $(3, 5)$ -схеме можно добавить еще одного участника так, чтобы порог и секрет s остались прежними.
8. Дать подробное обоснование того, что схема Шамира является линейной.
9. Определить уровень информации пороговой $(2, 3)$ -схемы в кольце $\mathbb{F}_2[x]$, если $m_1(x) = x^2$, $m_2(x) = x^2 + 1$, $m_3(x) = x^2 + x + 1$.
10. Построить идеальную $(3, 5)$ -схему Асмута – Блума в кольце $\mathbb{F}_2[x]$.
11. Какой дополнительный модуль лучше взять для схемы разделения секрета из задания 9.

Г л а в а 18

О НОВЫХ НАПРАВЛЕНИЯХ В КРИПТОЛОГИИ

18.1. О ВОЗМОЖНОСТЯХ КВАНТОВОЙ КРИПТОГРАФИИ

Криптология является многогранным направлением прикладной математики, интенсивно использующим такие фундаментальные понятия современной математической науки, как информация, случайность, алгоритм, сложность. Поскольку неопределенность и стохастичность окружающего мира имеют квантовую природу, то криптология связана с квантовой механикой и с ее основным постулатом – принципом неопределенности Гейзенберга.

Впервые идея шифрования с использованием квантового канала сформулирована в 1970 г. С. Уинснером и развита в 80-е гг. Ч. Беннетом, Ж. Брассаром и С. Брейдбардом [75].

Для пересылки последовательности двоичных битов в квантовом канале связи используются специальным образом поляризованные фотоны. Фотон – элементарная квантовая система, которая характеризуется определенным направлением поляризации $r = (r_1, r_2)' = (\cos \alpha, \sin \alpha)'$ (рис. 18.1).

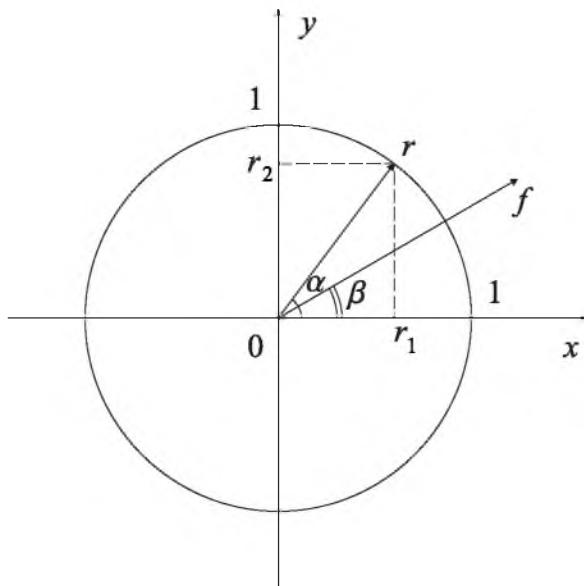


Рис. 18.1. Поляризация фотона

На приведенном рис. 18.1 r – единичный вектор, указывающий поляризацию фотона и направленный под углом α ; $r_1 = \cos \alpha$, $r_2 = \sin \alpha$ – проекции; f – направление поляризации фильтра-анализатора. Из законов квантовой механики следует, что при попадании в такой фильтр-анализатор с параметром β фотон ведет себя «дихотомическим и совершенно непредсказуемым образом» [6], проходя через фильтр без изменения с вероятностью $p = p(\alpha - \beta) = \cos^2(\alpha - \beta)$ и поглощаясь в нем с дополнительной вероятностью $q = q(\alpha - \beta) = 1 - p = \sin^2(\alpha - \beta)$. Это квантово-механическое явление проявляется и на макроуровне для интенсивности пучка поляризованного света при прохождении через фильтр в виде известного физического закона Малюса. Отметим, что если разность угла поляризации и угла ориентации фильтра-анализатора принимает значение $\gamma = \alpha - \beta \in \left\{ \frac{\pi}{4}, \frac{3\pi}{4} \right\}$, то $p(\gamma) = q(\gamma) = 1/2$, и фотон проходит фильтр или поглощается им с одной и той же вероятностью – $1/2$. Этот факт используется при выборе осей поляризации для квантового шифрования.

Принято говорить, что *прямоугольная поляризация* фотона имеет место, если его вектор поляризации

$$r = r^{(1)} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{или} \quad r = r^{(2)} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Прямоугольная поляризация обозначается символом «+». Если же

$$r = r^{(3)} = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \quad \text{или} \quad r = r^{(4)} = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix},$$

то говорят о *диагональной поляризации* и обозначают ее символом «×».

Проиллюстрируем применение квантовых сигналов в криптографии на примере передачи секретного ключа по открытому квантовому каналу. Рассмотрим классическую схему криптографической передачи информации, в которой участвует три персонажа:

- 1) Алиса, желающая передать Бобу секретный ключ, который в дальнейшем будет использоваться, например, для обмена информацией с помощью некоторой симметричной крипtosистемы;
- 2) Боб, который нуждается в секретном ключе;
- 3) Виктор – криptoаналитик, которая хотела бы скрытно завладеть копией передаваемого секретного ключа.

Алиса генерирует чисто случайную секретную двоичную ключевую последовательность длиной N $K = (K_1, \dots, K_N) \in V_N$ и чисто случайную секретную индексную последовательность той же длины $J = (J_1, \dots, J_N) \in V_N$.

Затем Алиса генерирует в квантовый канал связи последовательность N фотонов: i -му фотону F_i , несущему сигнал K_i , дается поляризация «+», если $J_i = 0$, и поляризация « \times », если $J_i = 1$.

Боб, имея два приемника фотонов (с поляризациями «+» и « \times »), принимает поток фотонов. Если бы он знал индексную последовательность J , то, переключая приемники, добился бы того, что $\gamma_i \equiv 0$, $p_i \equiv 1$, $q_i \equiv 0$, т. е., что последовательность фотонов была бы воспринята безошибочно, и, следовательно, получен ключ K . Поскольку Боб не знает J , то использует свою, наудачу выбранную индексную последовательность $J' \in V_N$. Очевидно, что в среднем у J и J' совпадет $N/2$ символов. Следовательно, лишь половина (в среднем) фотонов будет зарегистрирована безошибочно, а остальные фотоны не несут никакой информации о K :

$$\begin{aligned} K' &= (K'_1, \dots, K'_N) \in V_N, \\ K'_{i_l} &= K_{i_l}, \\ l &= 1, 2, \dots, n, \quad \mathbf{E}\{n\} = \frac{n}{2}. \end{aligned}$$

Заметим, что если Виктор извлечет некоторые фотоны из последовательности для измерения своими приемниками, то Боб сразу же обнаружит потерю фотонов.

Следующие шаги протокола выполняются в обычном канале связи. Прежде всего через этот канал Алиса и Боб определяют посредством открытого обмена сообщениями, какие фотоны зарегистрированы и какие из них соответствуют истинной поляризации $J'_{i_l} = J_{i_l}$. Если Виктор не нарушал квантовой передачи (т. е. получено столько же фотонов, сколько отправлено), то биты K_{i_1}, \dots, K_{i_n} , значения которых не обсуждались по открытому каналу, становятся общим секретным ключом для Алисы и Боба:

$$k = (k_1, \dots, k_n) = (K_{i_1}, \dots, K_{i_n}).$$

Из-за возможных действий Виктора (в том числе связанных со «вставкой» фотонов) Алиса и Боб должны убедиться, что их получившиеся битовые строки идентичны. Простое решение состоит в том, чтобы Алиса и Боб открыто сравнили некоторые m из n битов ($m \ll n$), относительно которых, как они думают, необходимо прийти к соглашению. Позиции этих «открыто сверяемых» битов должны быть выбраны после того, как квантовая передача завершена, чтобы лишить Виктора информации о том, какие фотоны она может измерять без опаски. Оставшиеся $n - m$ битов могут использоваться в качестве секретного ключа для последующей связи по открытому каналу с помощью одного из симметричных криптоалгоритмов. На рис. 18.2 аналогично [6] приведен протокол выработки секретного ключа с помощью квантового канала связи при $N = 15$, $n = 6$, $m = 2$, состоящий из 11 шагов.

Шаг	Информация																
	0	1	1	0	1	1	0	0	1	0	1	1	1	0	0	1	
1	0	1	1	0	1	1	0	0	1	0	1	1	1	0	0	1	
2	x	+	x	+	+	+	+	+	x	x	+	x	x	x	x	+	
3	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}		
4	+	x	x	+	+	x	x	+	x	+	x	x	x	x	x	+	
5	1		1		1	0	0	0		1	1	1		0	1		
6	+		x		+	x	x	+		+	x	x		x	+		
7			✓			✓			✓				✓		✓	✓	
8			1			1			0				1		0	1	
9						1								0			
10						✓								✓			
11						1							1			1	

Рис. 18.2. Протокол выработки секретного ключа с помощью квантового канала связи

1. Передача по квантовому каналу.
 - 1.1. Случайная битовая строка K , посылаемая Алисой.
 - 1.2. Последовательность поляризаций, задаваемая J .
 - 1.3. Посланная Алисой последовательность фотонов.
 - 1.4. Последовательность поляризаций J' , использованная Бобом.
 - 1.5. Битовая строка, зарегистрированная Бобом.
 2. Обсуждение по открытому каналу.
 - 2.1. Боб сообщает поляризацию зарегистрированных фотонов.
 - 2.2. Алиса отмечает, какие поляризации были угаданы правильно.
 - 2.3. Последовательность $n = 6$ секретных битов, которую можно использовать в качестве секретного ключа (если квантовая передача не была нарушена).
 - 2.4. Боб указывает номера $m = 2$ «открыто сверяемых» битов ключа.
 - 2.5. Алиса подтверждает эти биты.
 3. Результат: оставшиеся $n - m = 4$ секретных бита.
- Заметим в заключение, что техническая (аппаратная реализация представленного выше протокола выработки общего секретного ключа с помощью квантового канала связи встречается с определенными трудностями.

18.2. СТЕГАНОГРАФИЯ И ЕЕ ПРИМЕНЕНИЕ

Если криптологию трактовать в широком смысле, то стеганография (*steganography*) – направление криптологии, имеющее целью скрытие (*hiding*) сообщения в потоке передаваемой информации. Интересно проследить историю возникновения терминов «криптография» (от греч. *kryptos* –

скрытый) и стеганография (от греч. *steganos* – прикрытый). Термин «криптография», обозначающий «секретное письмо», впервые использован в 1641 г. Дж. Уилкинсом – одним из основателей Королевского Общества в Великобритании. При этом имелось в виду открытое секретное письмо (*overt secret writing*): открытое в том смысле, что очевидна зашифрованность письма. Термин «стеганография» введен в 1665 г. К. Шоттом. Здесь имелось в виду «скрытое секретное письмо» (*covert secret writing*): скрытое в том смысле, что тайной является сам факт наличия секретного сообщения в потоке передаваемой информации.

В докомпьютерной стеганографии Ф. Л. Бауэр [68] выделяет два направления: лингвистическую и техническую стеганографию. Первая использует, например, неопределенности написания букв, промежутков между словами для сокрытия сообщения. Классическим примером технической стеганографии является написание скрытого сообщения молоком на белой бумаге.

Современная стеганография применяется для тайной передачи сообщений внутри «безобидных» данных, для скрытой маркировки данных с помощью «водяных знаков» (*watermarking*) [147]. Отметим, что «водяные знаки» используются для защиты электронных произведений (книг, музыки, видео) от пиратского копирования.

Качество стеганографии характеризуется тремя факторами:

- 1) обнаруживаемость (*detectability*);
- 2) робастность (*robustness*);
- 3) информационная емкость (*bitrate*).

Обнаруживаемость – главная проблема тайной передачи информации, и поэтому передаваемая информация обычно предварительно шифруется. Робастность (устойчивость) ко всем ожидаемым типам обработки переданной информации (например, к фильтрации, усечению, масштабированию) является важнейшим требованием при изготовлении «водяных знаков». Информационная емкость – это максимальное количество битов скрываемого сообщения, при котором передаваемый сигнал еще не получает заметных искажений. В качестве сигналов-носителей тайных сообщений, называемых контейнерами, в современной стеганографии обычно используются аудиосигналы (музыка, речь), а также изображения и видео.

Типичная схема стеганографической передачи сообщения представлена на рис. 18.3, где X – исходный «безобидный» сигнал-носитель, который иногда называют контейнером; M – исходное сообщение, которое надо «спрятать» в сигнале X ; K – ключ для шифрования сообщения. Сначала осуществляют шифрование:

$$m = E_k(M).$$

Затем зашифрованное сообщение m «незаметно» внедряется («вкрапляет-

ся) в сигнал X некоторыми специальными способами, которые будут рассмотрены ниже: $X_m = f(X, m)$. После этого «нагруженный секретом» сигнал X_m пересыпается по каналу связи, в котором этот сигнал может быть подвергнут различным атакам. В результате получатель принимает искаженный сигнал \hat{X}_m и пытается выделить «спрятанное» сообщение M или хотя бы обнаружить его наличие. Использование ключа K является условным.

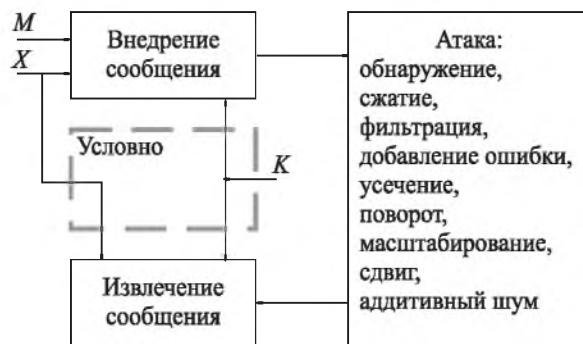


Рис. 18.3. Схема стеганографической передачи сообщения

Остановимся на способах задания функции $f(\cdot)$, т. е. на способах «внедрения» сообщения m в сигнал-контейнер X . Выделяются два класса этих способов: пространственные (временные); частотные (спектральные). Примером пространственного способа «внедрения» является способ, в котором m используется для изменения пикселей, номера которых определяет ключ K [72]. Другой способ состоит в модификации всех пикселей на «незаметную» величину [107]. Например, в способе LSB изменениям подвергаются младшие биты (Least Significant Bits).

Оказывается, что пространственные способы не позволяют достичь требуемого уровня «неприметности» искажений сигнала-контейнера X , поэтому наиболее широко используются частотные способы. При этом X вначале подвергается одному из спектральных преобразований (быстрое преобразование Фурье, дискретное косинус-преобразование, преобразование Уолша):

$$Y = g(X).$$

Затем сообщение m «внедряется» в массив коэффициентов преобразования Y одним из указанных выше пространственных способов:

$$Y_m = f(Y, m),$$

после чего осуществляет обратное спектральное преобразование:

$$X_m = g^{-1}(Y_m).$$

18.3. АКТИВНЫЙ КРИПТОАНАЛИЗ

Криптографические алгоритмы реализуются в *устройствах* – интеллектуальных карточках, аппаратных комплексах защиты информации, программных криптографических модулях. Такие устройства выполняют криптографические преобразования с использованием секретных параметров, хранящихся в защищенной памяти. Устройства вполне могут быть доступны злоумышленнику для экспериментов и манипуляций, не санкционированных разработчиками.

Методы *инженерного криптоанализа* направлены на определение архитектуры устройства, содержимого защищенной памяти, реализованных в устройстве секретных алгоритмов. При этом исследователи пользуются дорогостоящими физическими приборами и инструментами: специальными микроскопами и микрозондами, устройствами лазерной резки, источниками ионных и электронных пучков. Методами инженерной криптографии был восстановлен, например, алгоритм поточного шифрования A5, реализованный в SIM-картах GSM [103].

Методы *активного криптоанализа* менее дорогостоящие и занимают промежуточное место между математическими и инженерными методами. Криptoаналитик осуществляет следующее:

- *наблюдает* за выполнением алгоритма в аппаратном или программном устройстве: замеряет временные показатели и уровни напряжения, фиксирует флаги переноса и количество обращений к кэш-памяти, запоминает содержимое регистров в момент переключения между задачами процессора и др.;
- *кратковременно воздействует* на выполнение алгоритма: облучает устройство для изменения содержимого регистров, меняет тактовую частоту с целью внесения ошибок в порядок выполнения инструкций процессором;
- *разрушающее воздействует* на выполнение алгоритма: обнуляет ультрафиолетовыми лучами ячейки постоянной памяти, отрезает проводники, ведущие к ячейкам рабочих регистров.

Собранная информация и полученные при внесенных сбоях результаты выполнения алгоритма позволяют упростить решение математических задач криптоанализа.

Далее считаем, что в устройстве реализован следующий алгоритм возведения чисел $u \in \mathbb{Z}_n$, $\lceil \log_2 n \rceil = l$, в степень $d = (d_0, \dots, d_{l-1})_2$.

1. Установить $w \leftarrow 1$, $v \leftarrow u$.
2. Для $i = 0, 1, \dots, l - 1$ выполнить:
 - а) если $d_i = 1$, то $w \leftarrow (wv) \bmod n$;
 - б) $v \leftarrow v^2 \bmod n$.
3. Вернуть w .

Предположим, что устройство используется для подписи сообщений $x \in \mathbb{Z}_n$ по RSA. При этом n – составной модуль RSA, d – секретная экспонента, которая хранится в защищенной памяти устройства.

В классической постановке задачи криптоанализа злоумышленнику известны:

- 1) открытая экспонента e ;
- 2) подписываемые сообщения x ;
- 3) подписи $y = x^d \bmod n$.

Требуется определить секретный ключ d . Рассмотрим некоторые активные методы решения данной задачи.

Использование замеров времени

Пусть криптоаналитик располагает возможностью измерять время $R(x, d)$ вычисления подписи $x^d \bmod n$, а также оценивать время $R(x, \hat{d})$ для произвольной выбираемой экспоненты \hat{d} .

Для случайного $x \in \mathbb{Z}_n$ можно предположить, что время вычисления $R(x, d)$ является суммой независимых случайных величин

$$r_i = d_i \xi_i + \zeta_i, \quad i = 0, \dots, l - 1,$$

где ξ_i, ζ_i – время выполнения шагов 2,а) и 2,б) алгоритма соответственно.

Пусть криптоаналитик подбирает оценку \hat{d}_0 бита d_0 . Обозначим

$$\Delta(x, \hat{d}_0) = R(x, d) - R(x, \hat{d}_0).$$

Если $\hat{d}_0 = d_0$, то

$$\Delta(x, \hat{d}_0) = \sum_{i=1}^{l-1} r_i, \quad \mathbf{D}\{\Delta(x, \hat{d}_0)\} = \sum_{i=1}^{l-1} \mathbf{D}\{r_i\}.$$

Если же $\hat{d}_0 \neq d_0$, то

$$\Delta(x, \hat{d}_0) = \sum_{i=1}^{l-1} r_i \pm \xi_0, \quad \mathbf{D}\{\Delta(x, \hat{d}_0)\} = \sum_{i=1}^{l-1} \mathbf{D}\{r_i\} + \mathbf{D}\{\xi_0\}.$$

Таким образом, при изменении бита \hat{d}_0 с истинного значения на ложное дисперсия случайной величины $\Delta(x, \hat{d}_0)$ увеличивается на $\mathbf{D}\{\xi_0\}$.

Используя данные соображения, криптоаналитик проводит атаку по определению бита d_0 следующим образом.

1. Перехватываются сообщения x_t , $t = 1, \dots, T$, и фиксируется время $R(x_t, d)$ их возведения в степень.

2. Для $b = 0, 1$ формируются выборки $\Delta(x_1, b), \dots, \Delta(x_T, b)$ и определяются их выборочные дисперсии s_b^2 .

3. Если $s_0^2 > s_1^2$, то искомая оценка $\hat{d}_0 = 0$. Если же $s_0^2 < s_1^2$, то $\hat{d}_0 = 1$.

Определив d_0 , далее таким же образом можно определить бит d_1 , затем d_2, \dots, d_{l-1} .

Использование временных сбоев

Пусть при вычислении подписи $x^d \bmod n$ в устройстве происходит единичный сбой – на i -й итерации алгоритма в регистре $w = (w_0, \dots, w_{l-1})_2$ изменяется значение бита w_j . Будем считать, что i и j – случайные величины с равномерным распределением на множестве $\{0, 1, \dots, l - 1\}$.

Обозначим $d[i] = (d_0, \dots, d_{i-1})_2$, $D[i] = (0, \dots, 0, d_i, \dots, d_{l-1})_2$ (считаем, что $d[0] = D[l] = 0$). Тогда истинная подпись есть $y = x^{d[i]+D[i]} \bmod n$, а подпись, полученная при сбое, –

$$y' = \left(x^{d[i]} \pm 2^j \right) x^{D[i]} \bmod n = \left(y \pm \pm 2^j x^{D[i]} \right) \bmod n.$$

Криптоанализ проводится следующим образом.

1. Фиксируются сообщения x_t , $t = 1, \dots, T$, и соответствующие им сбоевые подписи y'_t .

2. Для $\hat{i} = l - 1, l - 2, \dots$:

а) выбираются все возможные номера \hat{j} бита ошибки;

б) выбираются все возможные значения $\hat{D}[\hat{i}]$;

в) для всех $t = 1, \dots, T$ проверяется совпадение $(y'_t \mp 2^{\hat{j}} x_t^{\hat{D}[\hat{i}]})^e \bmod n = x_t$.

3. Если совпадение найдено, то с большой вероятностью $\hat{D}[\hat{i}] = D_t[\hat{i}]$ и определены $l - \hat{i}$ битов секретной экспоненты d .

Пусть i_1, \dots, i_T – неизвестные номера сбойных итераций, упорядоченные по неубыванию. Тогда $\mathbf{P}\{i_T < l - m\} = 1 - \left(1 - \frac{m}{l}\right)^T$, $0 \leq m < l$, и на шаге 2 с такой вероятностью потребуется перебрать не более 2^m значений $\hat{D}[\hat{i}]$. Ясно, что атаку можно продолжить и восстановить следующую порцию битов d .

Использование необратимых сбоев

Пусть секретная экспонента хранится в защищенной памяти EEPROM (Electrically Erasable Programmable Read-Only Memory) как бинарное слово θ_0 . Такая память является асимметричной – при воздействии на нее ультрафиолетовым излучением заряд (бит ключа) скорее обнулится, чем примет значение 1.

Криптоаналитик действует следующим образом.

1. Задает произвольное сообщение $x \in \mathbb{Z}_n^*$ и получает значение подписи $y = x^d \bmod n$.

2. Для $t = 1, 2, \dots$ криптоаналитик облучает устройство, формируя таким образом ключ θ_t , отличающийся от θ_{t-1} на небольшое количество битов. Затем для сообщения x снова вырабатывается подпись y_t , но уже на ключе θ_t . Вычисления проводятся до тех пор, пока не встретится подпись $y_T = y_{T+1} = \dots = 1$.

3. Веса Хэмминга (количество ненулевых битов) ключей удовлетворяют цепочке равенств $w(\theta_0) \geq w(\theta_1) \geq \dots \geq w(\theta_{T-1}) > w(\theta_T) = 0$. Для $t = T - 1, \dots, 0$ ключ θ_t ищется в малой окрестности ключа θ_{t+1} . Для каждого кандидата $\hat{\theta}_t$ проверяется совпадение соответствующей подписи \hat{y}_t со значением y_t .

БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ

1. Агиеевич С. В., Афоненко А. А. О свойствах экспоненциальных подстановок // Вести НАН Беларуси. 2005. № 1. С. 106–112.
2. Основы криптографии / А. П. Алферов [и др.]. М., 2005.
3. Андерсон Т. В. Статистический анализ временных рядов. М., 1984.
4. Большев Л. Н., Смирнов Н. В. Таблицы математической статистики. М., 1983.
5. Боровков А. А. Математическая статистика. М., 1988.
6. Бассар Ж. Современная криптология. М., 1999.
7. Van der Варден Б. Л. Алгебра. М., 1976.
8. Варфоломеев А. А., Жуков А. Е., Пудовкина М. А. Поточные криптосистемы. Основные свойства и методы анализа стойкости. М., 2000.
9. Введение в криптографию / под ред. В. В. Ященко. М., 2001.
10. Верн Ж. Жангада. Минск, 1992.
11. Габасов Р. Ф. Основы динамического программирования. Минск, 1982.
12. Герасименко В. А., Малюк А. А. Основы защиты информации : учеб. пособие. М., 1997.
13. Дуб Дж. Вероятностные процессы. М., 1956.
14. Анализ биологических последовательностей / Р. Дурбин [и др.]. М., 2006.
15. Духин А. А. Теория информации. М., 2007.
16. Ивченко Г. И., Медведев Ю. И. Введение в математическую статистику : учебник. М., 2010.
17. Жельников В. Криптография от папируса до компьютера. М., 1996.
18. Кейперс Л., Нидеррайтер Г. Равномерное распределение последовательностей. М., 1985.
19. Кемени Дж., Снелл Дж. Конечные цепи Маркова. М., 1970.
20. Килин С. Я., Хорошко Д. Б., Низовцев А. П. Квантовая криптография: идеи и практика. Минск, 2007.
21. Китаев А., Шень А., Вялый М. Классические и квантовые вычисления. М., 1999.
22. Кнут Д. Искусство программирования : в 3 т. М., 2000. Т. 1 : Основные алгоритмы.

23. Кнут Д. Искусство программирования : в 3 т. М., 2000. Т. 2 : Получисленные алгоритмы.
24. Кокс Д., Литтл Дж., О'Ши Д. Идеалы многообразия и алгоритмы. М., 2000.
25. Колмогоров А. Н., Успенский В. А. Алгоритмы и случайность // Теория вероятностей и ее применения. 1987. Т. 32, вып. 3. С. 425–455.
26. Королюк В. С. Справочник по теории вероятностей и математической статистике. М., 1985.
27. Кузюрин Н. Н., Фомин С. А. Эффективные алгоритмы и сложность вычислений. М., 2007.
28. Левин Л. А. Универсальные задачи перебора // Проблемы передачи информации. 1973. Вып. 3. С. 115–116.
29. Лидл Р., Нидеррайтер Г. Конечные поля : в 2 т. М., 1988.
30. Булевы функции в теории кодирования и криптологии / О. А. Логачев [и др.]. М., 2012.
31. Максимов Ю. И. О цепях Маркова, связанных с двоичными регистрами сдвига со случайными элементами // Тр. по дискретной математике. 1997. Т. 1. С. 203–220.
32. Молдовян Н. А. Проблематика и методы криптологии. СПб., 1998.
33. Молдовян Н. А. Теоретический минимум и алгоритмы цифровой подписи. СПб., 2010.
34. Нечаев В. И. Элементы криптографии. Основы теории защиты информации. М., 1999.
35. Сачков В. Н. Введение в комбинаторные методы дискретной математики. М., 1982.
36. Соболева Т. Л. История шифровального дела в России. М., 2002.
37. Соловей О. В., Харин Ю. С. Математические методы генераторов двоичных последовательностей с неравномерным движением регистров: обзор // Управление защищкой информации. 2002, Т. 6, № 32. С. 77–83.
38. Столлингс В. Криптография и защита сетей. Принципы и практика. М., 2002.
39. Стратонович Р. Л. Теория информации. М., 1975.
40. Тарасенко Ф. П. Введение в курс теории информации. Томск, 1973.
41. Токарева Н. Н. Симметричная криптография. Краткий курс : учеб. пособие. Новосибирск, 2012.
42. Феллер В. Введение в теорию вероятностей и ее применения : в 2 т. М., 1984. Т. 1.
43. Фомичев В. М. Дискретная математика и криптология. М., 2003.

44. Харин Ю. С. Вероятностно-статистический анализ цепей Маркова высокого порядка // Вестник БГУ. Сер. 1. 2006. № 3. С. 80–86.
45. Харин Ю. С. Оптимальность и робастность в статистическом прогнозировании : монография. Минск, 2008.
46. Харин Ю. С. Цепи Маркова с r -частичными связями и их статистическое оценивание // Доклады НАН Беларуси. 2004. Т. 48, № 1. С. 40–44.
47. Эконометрическое моделирование / Ю. С. Харин [и др.]. Минск, 2003.
48. Харин Ю. С., Агееевич С. В. Компьютерный практикум по математическим методам защиты информации. Минск, 2001.
49. Харин Ю. С., Берник В. И., Матвеев Г. В. Математические основы криптологии. М., 1999.
50. Математические и компьютерные основы криптологии / Ю. С. Харин [и др.]. Минск, 2003.
51. Харин Ю. С., Гурин А. С. Статистические оценки параметров векторных авторегрессионных временных рядов при наличии пропущенных значений и их асимптотические свойства // Доклады НАН Беларуси. 2006. Т. 50, № 1. С. 18–24.
52. Харин Ю. С., Зуев Н. М., Жук Е. Е. Теория вероятностей, математическая и прикладная статистика. Минск, 2011.
53. Харин Ю. С., Мартиневский А. В. Обнаружение n -мерной равномерности в двоичных последовательностях с использованием экстремальных статистик // Pattern Recognition and Information Processing. 1999. Vol. 2. P. 358–362.
54. Харин Ю. С., Петличкий А. И. Цепь Маркова s -го порядка с r частичными связями и их статистическое оценивание // Дискретная математика. 2007. Т. 12, вып. 2. С. 109–130.
55. Харин Ю. С., Степанова М. Д. Практикум на ЭВМ по математической статистике. Минск, 1987.
56. Харин Ю. С., Ярмола А. Н. Статистическое оценивание параметров МТД-модели дискретных временных рядов // Известия НАН Беларуси : сер. физ.-мат. наук. 2006. № 2. С. 20–25.
57. Черемушкин А. В. Криптографические протоколы: основные свойства и уязвимости // Прикладная дискретная математика. 2009. С. 115–150.
58. Шенец Н. Об информационном уровне модулярных схем разделения секрета // Доклады НАН Беларуси : сер. физ.-мат. наук. 2010. Т. 54. С. 9–12.
59. Ширяев А. Н. Вероятность. М., 1980.
60. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы и исходные тексты на языке С. М., 2003.
61. Adleman L. A subexponential algorithm for the discrete logarithm problem with applications to cryptography // IEEE 18th Ann. Symp. on Found. of Comp. Sc. 1979. P. 55–60.

62. Agievich S. Two-stage allocations and the double Q -function // Electronic Journal of Combinatorics. № 10. 2003. P. 21.
63. Arora S., Barak B. Computational Complexity. A Modern Approach. Cambridge, 2007.
64. Bach E., Shallit J. Algorithmic Number Theory. Massachusetts, 1997. Vol. I : Efficient Algorithms.
65. Banghe Li. Generalizations of RSA Public Key Cryptosystem // IACR, Cryptology ePrint Arc. 2005.
66. Barrett P. Implementing the Rivest, Shamir and Adleman public key encryption algorithm on a standard digital signal processor // Adv. in Cryptology: Proc. of Crypto'86. 1987. LNCS 263. P. 311–323.
67. Baum L. E., Petrie T. Statistical inference for probabilistic functions of finite state Markov chains // The Annals of Mathematical Statistics. 2000. Vol. 52, № 2. P. 287–315.
68. Bauer F. Decrypted Secrets: methods and maxims of cryptology. N.Y., 1997.
69. Basawa I. V., Rao B. L. C. Statistical inference for stochastic processes. N.Y., 1980.
70. Becker T., Weispfenning V. Grebner Bases. A Computational Approach to Commutative Algebra. N.Y., 1993.
71. Bender J., Fischlin M., Kuegler D. Security Analysis of the PACE Key-Agreement Protocol // Cryptology ePrint Archive. Report 2009/624. 2009.
72. Techniques for data hiding / W. Bender [and others] // IBM Syst. J. 1996. Vol. 35, № 3–4. P. 313–336.
73. Beth T., Ding C. On Almost Perfect Nonlinear Permutations // Advances in Cryptology: EUROCRYPT'93 Proceedings. N.Y., 1994. P. 65–76.
74. Biham E., Shamir A. Differential Cryptanalysis of DES-like Cryptosystems // Journal of Cryptology. 1991. Vol. 4. P. 3–72.
75. Quantum cryptography, or unforgeable subway tokens / C. H. Bennett [and others] // Adv. in Cryptology: Proc. of Crypto'82. 1982. P. 267–275.
76. Berchtold A. Estimation of the Mixture Transition Distribution Model // J. of Time Ser. Anal. 2001. Vol. 22, № 4. P. 379–397.
77. Beth T., Piper F. C. The stop-and-go generator // Adv. in Cryptology: Proc. of Eurocrypt'84. 1985. LNCS 209. P. 88–92.
78. High-speed quantum random number generation by measuring phase noise of a single-mode laser / Qi Bing [and others] // Optics Letters. 2010. Vol. 35. Issue 3. P. 312–314.
79. Blake I., Seroussi G., Smart N. Elliptic Curves in Cryptography. Cambridge, 1999.
80. Brüer J. O. On pseudo random sequences as crypto generators // Proc. of the International Zurich Seminar on Dig. Com. Zurich, 1984.

81. *Buhmann P., Wyner A.* Variable length Markov chains // The Annals of Statistics. 1999. Vol. 27, № 2. P. 480–513.
82. *Chambers W. G., Gollmann D.* Lock-in effect in cascades of clock-controlled shift-registers // Adv. in Cryptology: Proc. of Eurocrypt'88. 1988. LNCS 330. P. 331–343.
83. *Chassaing P.* An optimal random number generator on \mathbb{Z}_p // Statistics & Probability Letters. 1989. Vol. 7. P. 307–309.
84. *Cohen H.* A Course in Computational Algebraic Number Theory. N.Y., 1993.
85. *Cook S.* The complexity of theorem-proving procedures. Proc. 3rd STOC. N.Y., 1971. P. 151–158.
86. *Coppersmith D., Krawchuk Y., Mansour Y.* The shrinking generator // Adv. in Cryptology: Proc. of Crypto'93. 1994. LNCS 773. P. 22–39.
87. *Davis D., Ihaka R., Fenstermacher P. R.* Cryptographic Randomness from Air Turbulence in Disk Drives // Advances in Cryptology: CRYPTO '94 Conference Proceedings ; edited by Yvo G. Desmedt. 1994. LNCS 839. P. 114–120.
88. *Diffie W., Hellman M.* New Directions in Cryptography // IEEE Transactions on Information Theory. 1976. IT-22. P. 644–654.
89. *Diffie W., Oorschot P., Wiener M.* Authentication and Authenticated Key Exchanges. Designs, Codes and Cryptography. N.Y., 1992. 2(2). P. 107–125.
90. *Eastlake D., Schiller J., Crocker S.* RFC 4086 // Randomness Requirements for Security. Massachusetts, 2005.
91. *Eichenauer J., Lehn J., Topuzoglu A.* A nonlinear congruential pseudorandom number generator with power of two modulus // Mathematics of Comp. 1988. Vol. 51. P. 757–759.
92. *Elias P.* The efficient construction of an unbiased random sequence // The Math. Statistics. 1972. Vol. 43. P. 865–870.
93. *Elias P.* Universal Codeword Sets and Representations of the Integers // IEEE Trans. Information Theory. 1975. 21(2). P. 194–203.
94. *Gaines H. F.* Cryptanalysis. N.Y., 1939.
95. *Galibus T., Matveev G.* Finite Fields, Grobner Bases and Modular Secret Sharing // Journal of Discr. Math. Science and Cryptography. 2012. Vol. 15. Issue 6. P. 339–348.
96. *Galibus T., Matveev G.* Generalized Mignotte's sequences over polynomial rings // ENTCS. 2007. Vol. 186. P. 43–48.
97. *Galibus T., Matveev G., Shenets N.* Some Structural and Security Properties of the Modular Secret Sharing // SYNASC. 2008 ; IEEE CPC Los Alamitos, California. 2009. P. 197–200.
98. *Games R. A., Chan A. H.* A fast algorithm for determining the complexity of a binary sequence with period 2^n // IEEE Trans. on Inf. Theory. 1983. Vol. 29. P. 144–146.
99. *Geffe P.* How to protect data with ciphers that are really hard to break // Electronics. 1973. Vol. 46. P. 99–101.

100. *Goldreich O.* Computational Complexity. A Conceptual Perspective. Cambridge, 2008.
101. *Goldreich O.* Foundations of Cryptography. Basic Tools. Cambridge, 2004.
102. *Goldwasser S., Bellare M.* Lecture Notes on Cryptography. URL : <http://cseweb.ucsd.edu/mihir/papers/gb.pdf>. Date of access : 2008.
103. *Golić J. D.* Cryptanalysis of alleged A5 stream cipher // Adv. in Cryptology: Proc. of Eurocrypt'97. 1997. LNCS 1233. P. 239–255.
104. *Golomb S. W.* Shift Register Sequences. San-Francisco, 1967.
105. *Günther C. G.* Alternating steps generators controlled by de Bruijn sequences // Adv. in Cryptology: Proc. of Eurocrypt'87. 1988. LNCS 304. P. 5–14.
106. *Guterman Z., Pinkas B., Reinman T.* Analysis of the Linux Random Number Generator. URL : <http://eprint.iacr.org/2006/086.pdf>. Date of access : 2006.
107. *Hartung F., Girod B.* Watermarking of compressed and uncompressed video // Signal Processing. 1998. Vol. 66, № 3. P. 283–301.
108. *Hellman M.* A Cryptanalytic Time-Memory Tradeoff // IEEE Trans. on Inf. Theory. 1980. Vol. IT-26 (4). P. 401–406.
109. *Iwata T., Kurosawa K.* OMAC: One-Key CBC MAC // LNCS. 2003. Vol. 2887. P. 129–153.
110. *Jacobs P. A., Lewis P. A. W.* Discrete time series generated by mixtures // J. Royal Statist. Soc. B. 1978. Vol. 40, № 1. P. 94–105.
111. *Jakobsen T.* A fast method of cryptanalysis of substitution ciphers // Cryptologia. 1995. Vol. 19, № 3. P. 265–274.
112. *Johnson D.* A Catalog of Complexity Classes // Handbook of Theoretical Computer Science. Massachusetts, 1990. Vol. A. P. 67–161.
113. *Joux A.* Multicollisions in iterated hash functions, application to cascaded constructions // CRYPTO. 2004. LNCS 3152. P. 306–316.
114. *Karp R.* Reducibility among combinatorial problems // Complexity of Computer Computations. N.Y., 1972. P. 85–103.
115. *Kahn D.* The Codebreakers. N.Y., 1967.
116. *Kelsey J., Schneier B., Ferguson N.* Yarrow-160: Notes on the design and analysis of the Yarrow cryptographic pseudorandom number generator // 6th Ann. Workshop on Selected Areas in Cryptography. N.Y., 1999.
117. *Klein A.* Stream Ciphers. London, 2013.
118. *Kranakis E.* Primality and Cryptography. N.Y., 1983.
119. *Kullback S.* Statistical Methods in Cryptanalysis. Laguna Hills, CA, 1976.
120. *Lachaud G., Wolfmann J.* The Weights of the Orthogonals of the Extended Quadratic Binary Goppa Codes // IEEE Trans. Inform. Theory. 1990. Vol. 36. P. 686–692.
121. *Lenstra H. W. Jr.* Factoring integers with elliptic curves // Ann. of Math. 1987. Vol. 126. P. 649–673.

122. *MacLaren M., Marsaglia G.* Uniform random number generators // J. ACM 12. 1965. № 1. P. 83–89.
123. *Marsaglia G.* DIEHARD: a battery of tests of randomness. URL : <http://stst.fsu.edu/geo/diehard.html>. Date of access : 2012.
124. *Marsaglia G.* Marsaglia random number CD-ROM. N.Y., 1996.
125. *Marsaglia G.* The mathematics of random number generators // Proc. of Symp. on Appl. Math. 1992. Vol. 46. P. 73–89.
126. *Marsaglia G., Bray T.A.* One line random number generators and their use in combinatorics // Comm. ACM. 1968. Vol. 11. P. 757–759.
127. *Marsaglia G., Tsay L.* Matrices and the structure of random number sequences // Linear Algebra and Its Appl. 1985. Vol. 67, № 1. P. 147–156.
128. *Marsaglia G., Zaman A.* A new class of random number generators // Ann. of Appl. Prob. 1991. Vol. 1, № 3. P. 462–480.
129. *Matsui M.* Linear Cryptanalysis Method for DES Cipher // Adv. in Cryptology: Proc. of Eurocrypt'93. 1994. LNCS 765. P. 386–397.
130. *Matsumoto T., Takashima Y., Imai H.* On Seeking Smart Public-key Distribution Systems // Transactions of the IECE of Japan. 1986. Vol. 69. P. 99–106.
131. *Maurer U.* A universal statistical test for random bit generators // Adv. in Cryptology: Proc. of Crypto'90. 1991. LNCS 537. P. 409–420.
132. *Meier W., Staffelbach O.* Fast correlation attacks on stream ciphers // Adv. in Cryptology: Proc. of Eurocrypt'88. 1988. LNCS 330. P. 301–314.
133. *Menezes A., Oorschot P., Vanstone S.* Handbook of Applied Cryptology. N.Y., 1997.
134. *Menezes A., Qu M., Vanstone S.* Some New Key Agreement Protocols Providing Mutual Implicit Authentication // Workshop on Selected Areas in Cryptography (SAC '95). 1995. P. 22–32.
135. *Montgomery P.* Modular multiplication without trial division // Math. of Comp. 1985. Vol. 44. P. 519–521.
136. *Montgomery P.* Speeding the Pollard and elliptic curve methods of factorization // Mathematics of Computation. № 48. 1987. P. 243–264.
137. *Nyberg K.* Differentially Uniform Mapping for Cryptography // Adv. in Cryptology: Proc. of Eurocrypt'93. N.Y., 1994. P. 55–64.
138. *Pieprzyk J., Hardjono T., Seberry J.* Fundamentals of Computer Security. N.Y., 2003.
139. *Pincus S., Singer B. H.* Randomness and degrees of irregularity // Proc. Nat. Acad. Sci. N.Y., 1993. P. 2083–2088.

140. *Pohlig S. C., Hellman M. E.* An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance // IEEE Trans. Inf. Theory. 1985. Vol. 24. P. 106–110.
141. *Pollard J. M.* Theorems on factorization and primality testing // Proc. of the Cambridge Phil. Soc. 1974. Vol. 76. P. 521–528.
142. *Pomerance C.* The quadratic sieve factoring algorithm // Adv. in Cryptology: Proc. of Eurocrypt'84. 1985. LNCS 209. P. 169–182.
143. *Poularikas A. D.* The Transforms and Applications Handbook. Boca Raton, 2000.
144. *Pratt V.* Every Prime Has a Succint Certificate // SIAM J. Comput. 4(3). 1974. P. 214–220.
145. *Raftery A. E.* A model for high-order Markov chains // Journal of the Royal Statistical Society. Ser. B. 1985. Vol. 47, № 3. P. 528–539.
146. *Rivest R. L.* On NIST's Proposed Digital Signature Standard // LNCS 739. 1993. P. 481–484.
147. *Robie D. L., Merserean R. M.* Video error correction using steganography // Eur. J. Appl. Sign. Processing. 2002. Vol. 2002, № 2. P. 164–173.
148. *Rabin M.* Digitalized signatures and public-key functions as intractable as factorization // Technical Report 212. Massachusetts, 1979.
149. *Rueppel R. A.* Analysis and Design of Stream Ciphers. Berlin, 1986.
150. *Rueppel R. A., Staffelbach O. J.* Products of linear recurring sequences with maximum complexity // IEEE Trans. Inf. Theory. 1987. Vol. 33. P. 124–131.
151. *Rukhin A. L.* Approximate Entropy for Testing Randomness. URL : <http://www.math.umbc.edu/ftp/tr98-07.html>. Date of access : 2012.
152. *Rukhin A. L.* A statistical test suite for random and pseudorandom number generators for cryptographic applications / NIST. URL : <http://csrc.nist.gov/rng/SP800-22b.pdf>. Date of access : 2001.
153. *Schneier B.* Applied Cryptography: Protocols, Algorithms, and Source Code in C. N.Y., 1996.
154. *Schnorr C. P.* Efficient signature generation by Smart Cards // J. of Cryptology. 1991. Vol. 4. P. 161–174.
155. *Siegenthaler T.* Decrypting a class of stream ciphers using ciphertext only // IEEE Trans. Comp. 1985. Vol. C-34. P. 81–85.
156. *Silverman J. H.* The Arithmetic of Elliptic Curves. N.Y., 1986.
157. *Sinkov A.* Elementary Cryptanalysis. Mathematical Approach. N.Y., 1956.
158. *Stinson D. R.* Cryptography. Theory and Practice. N.Y., 1995.
159. *Symul T., Assad S. M., Lam P. K.* Real time demonstration of high bitrate quantum random number generation with coherent laser light // Appl. Phys. Lett. 2011. № 98. P. 231–233.

160. *Wegman M., Carter J.* New hash functions and their use in authentication and set equality // Journal of Computer and System Sciences. 1981. Vol. 22. P. 265–279.
161. *Willems F.* Universal data compression and repetition times // IEEE Trans. Inf. Theory. 1989. Vol. 35. P. 54–58.
162. *Young A. L.* Mathematical Ciphers from Caesar to RSA. N.Y., 2006.
163. *Zeng K., Huang M.* An improved linear syndrome algorithm in cryptanalysis with applications // Adv. in Cryptology: Proc. of Crypto'90. 1990. LNCS 537. P. 34–47.
164. *Ziv J., Lempel A.* On the complexity of finite sequences // IEEE Trans. Inf. Theory. 1976. Vol. 22. P. 75–81.
165. Information Security Institute / Crypt-X. URL : www.isi.qut.edu.au. Data of access : 1998.
166. List of General NESSIE Test Tools. URL : www.cryptonessie.org. Data of access : 2010.
167. The Intel Random Number Generator // Cryptography Research. Inc. San Francisco, 1999.

ПРИЛОЖЕНИЯ

1. АРХИВ ДИСКРЕТНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Архив содержит 18 файлов дискретных последовательностей (модельных и реальных), которые используются при выполнении заданий данного практикума. Архив размещен на Интернет-сайте: <http://apmi.bsu.by>.

1. **File00** – «эталонная чисто случайная» двоичная последовательность Марсальи $x_t \in \{0, 1\}$ длиной 65536 элементов.

2. **File01** – двоичная псевдослучайная последовательность, порожденная С-генератором `rand`, длиной 65536 элементов.

3. **File02.1** – двоичная стационарная цепь Маркова длиной 65536 элементов с матрицей вероятностей одношаговых переходов $P(0, 2)$, где

$$P(\varepsilon) = \begin{pmatrix} 0,5 + \varepsilon & 0,5 - \varepsilon \\ 0,5 - \varepsilon & 0,5 + \varepsilon \end{pmatrix}, \quad -0,5 \leq \varepsilon \leq 0,5.$$

4. **File02.2** – двоичная стационарная цепь Маркова длиной 65536 элементов с матрицей переходов $P(0, 1)$.

5. **File02.3** – двоичная стационарная цепь Маркова длиной 65536 элементов с матрицей переходов $P(0, 05)$.

6. **File02.4** – двоичная стационарная цепь Маркова длиной 65536 элементов с матрицей переходов $P(0, 01)$.

7. **File02.5** – двоичная стационарная цепь Маркова длиной 65536 элементов с матрицей переходов $P(0, 001)$.

8. **File02.6** – двоичная стационарная цепь Маркова длиной 65536 элементов с матрицей переходов $P(0, 0001)$.

9. **File03.1** – двоичная линейная рекуррента порядка $s = 7$: $x_t = x_{t-3} \oplus x_{t-7}$, $t = 1, 2, \dots$, длиной 65536 элементов.

10. **File03.2** – двоичная линейная рекуррента порядка $s = 15$: $x_t = x_{t-7} \oplus x_{t-15}$, $t = 1, 2, \dots$, длиной 65536 элементов.

11. **File03.3** – двоичная линейная рекуррента порядка $s = 31$: $x_t = x_{t-13} \oplus x_{t-31}$, $t = 1, 2, \dots$, длиной 65536 элементов.

12. **File04.1** – бинарная авторегрессия BAR(7) длиной 65536 элементов порядка $s = 7$: $x_t = x_{t-3} \oplus x_{t-7} \oplus \xi_t$, $t = 1, 2, \dots$, где $\xi_t \in \{0, 1\}$ – последовательность независимых одинаково распределенных случайных величин Бернулли: $\mathbf{P}\{\xi_t = 0\} = 0,6$; $\mathbf{P}\{\xi_t = 1\} = 0,4$.

13. File04.2 – бинарная авторегрессия BAR(15) длиной 65536 элементов порядка $s = 15$: $x_t = x_{t-7} \oplus x_{t-15} \oplus \xi_t$, $t = 1, 2, \dots$, где $\xi_t \in \{0, 1\}$ – последовательность независимых одинаково распределенных случайных величин Бернулли: $P\{\xi_t = 0\} = 0,6$, $P\{\xi_t = 1\} = 0,4$.

14. File04.3 – бинарная авторегрессия BAR(31) длиной 65536 элементов порядка $s = 31$: $x_t = x_{t-13} \oplus x_{t-31} \oplus \xi_t$, $t = 1, 2, \dots$, где $\xi_t \in \{0, 1\}$ – последовательность независимых одинаково распределенных случайных величин Бернулли: $P\{\xi_t = 0\} = 0,6$, $P\{\xi_t = 1\} = 0,4$.

15. File05 – двоичная последовательность длиной 65536 элементов, представляющая фрагмент гимна кафедры ММАД.

16. File06 – двоичная последовательность длиной 65536 элементов, представляющая фрагмент оцифрованного изображения эмблемы БГУ.

17. File07 – «эталонная чисто случайная» N -значная последовательность $x_t \in \{0, 1, \dots, N - 1\}$ длиной 65536 элементов при $N = 8$.

18. File08 – стационарная цепь Маркова с $N = 8$ состояниями $V = 0, 1, \dots, 7$ длиной 65536 элементов с матрицей вероятностей одношаговых переходов

$$P = (p_{ij}), \quad p_{ii} = \frac{1}{2}, \quad p_{ij} = \frac{1}{14} \quad (i \neq j), \quad i, j = 1, \dots, 8.$$

2. ТАБЛИЦЫ

Таблица П.1
**Большие числа
и астрономические величины**

Астрономическая величина	Число
Возраст Земли, лет	2^{30}
Возраст Вселенной, лет	2^{34}
Число атомов на Земле	2^{170}
Число атомов на Солнце	2^{190}
Число атомов во Вселенной	2^{265}
Объем Вселенной, см ³	2^{280}

Таблица П.2
Большие числа

n	2^n	n	2^n
10	1024	120	$1,32923 \cdot 10^{36}$
20	$1,04858 \cdot 10^6$	140	$1,3938 \cdot 10^{42}$
30	$1,07374 \cdot 10^9$	160	$1,4615 \cdot 10^{48}$
40	$1,09951 \cdot 10^{12}$	180	$1,5325 \cdot 10^{54}$
50	$1,1259 \cdot 10^{15}$	200	$1,60694 \cdot 10^{60}$
60	$1,15292 \cdot 10^{18}$	220	$1,685 \cdot 10^{66}$
70	$1,18059 \cdot 10^{21}$	240	$1,76685 \cdot 10^{72}$
80	$1,20893 \cdot 10^{24}$	260	$1,85267 \cdot 10^{78}$
90	$1,23794 \cdot 10^{27}$	280	$1,94267 \cdot 10^{84}$
100	$1,26765 \cdot 10^{30}$	300	$2,03704 \cdot 10^{90}$

Таблица П.3
Точные значения больших чисел

<i>n</i>	2^n	<i>n</i>	2^n
1	2	33	8589934592
2	4	34	17179869184
3	8	35	34359738368
4	16	36	68719476736
5	32	37	137438953472
6	64	38	274877906944
7	128	39	549755813888
8	256	40	1099511627776
9	512	41	2199023255552
10	1024	42	4398046511104
11	2048	43	8796093022208
12	4096	44	17592186044416
13	8192	45	35184372088832
14	16384	46	70368744177664
15	32768	47	140737488355328
16	65536	48	281474976710656
17	131072	49	562949953421312
18	262144	50	1125899906842624
19	524288	51	2251799813685248
20	1048576	52	4503599627370496
21	2097152	53	9007199254740992
22	4194304	54	18014398509481984
23	8388608	55	36028797018963968
24	16777216	56	72057594037927936
25	33554432	57	144115188075855872
26	67108864	58	288230376151711744
27	134217728	59	576460752303423488
28	268435456	60	1152921504606846976
29	536870912	61	2305843009213693952
30	1073741824	62	4611686018427387904
31	2147483648	63	9223372036854775808
32	4294967296	64	18446744073709551616

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

- абелева группа, 42
автоморфизм, 46
активный криptoанализ, 483
алгебраические методы криptoанализа поточных криптосистем, 306
алгоритм DSA, 410
алгоритм Берлекэмпа – Месси, 129, 294
алгоритм Бухбергера, 469
алгоритм Евклида, 20
алгоритмическое определение случайности, 136
алфавит, 14
алфавит сообщений, 150
аннулирующий многочлен, 58
аномальная эллиптическая кривая, 423
асимметричная криптосистема, 286
асимптотические свойства стационарного ИДС, 184
асимптотический размер теста, 113
атака «дней рождения», 377
аутентифицируемый канал связи, 433
- базис Гребнера, 469
батареи статистических тестов, 109
батарея тестов «DIEHARD», 109
батарея тестов CRYPT-X, 110
батарея тестов NESSIE, 110
батарея тестов NIST, 109
батарея тестов Д. Кнута, 109
биграммная (N -граммная) подстановка, 204
биграммное шифрование, 16
бинарный линейный (n, k)-код, 401
блочная криптосистема, 286, 309
блочно-итерационная криптосистема, 313
блочно-итерационная функция хэширования, 373
- вероятностная машина Тьюринга, 222
вероятностная модель, 89
вычет по модулю m , 24
вычислительная задача, 214
- «гамма», 287
генератор ANSI X9.17, 272
генератор FIPS-186, 273
- генератор Yarrow-160, 274
генератор Беса – Пайпера, 300
генератор Геффе, 296
генератор РРСП, 260
генератор Таусворта, 269
генератор Фибоначчи, 270
генератор Эйхенауэра – Лёна с обращением, 267
генератор методом Макларена – Марсальи, 279
генератор с неравномерным движением, 278
генераторы с дополнительной памятью, 278
гомоморфизмы групп, 46
гомоморфизмы колец, 54
гомофоническое шифрование, 15
группа, 42
группа классов вычетов, 43
- двоичный временной ряд, 88
дискретная авторегрессия порядка s , 106
дискретная случайная последовательность, 88
дискретное преобразование Фурье, 91, 263
дискретный временной ряд (ДВР), 88
дискретный временной ряд стационарный в узком смысле, 90
дискретный временной ряд стационарный в широком смысле, 90
дискретный логарифм, 30
дискриминант эллиптической кривой, 75, 79
дифференциальная энтропия, 171
- евклидовы кольца, 55
- задача дискретного логарифмирования, 429
задача поиска, 215
задача распознавания, 215
задача факторизации, 392, 426
зашифрованный текст, 14
защита информации, 12
защита от «чтения назад», 441
- идеальная СРС, 452
идеальная модульная реализация, 475
ИДС стационарный, 151
изогения, 83

- индекс подгруппы, 44
индекс числа по модулю m при основании, 30
информационный уровень участника СРС, 453
информационный уровень СРС, 453
источник дискретных сообщений (ИДС), 150
источник непрерывных сообщений (ИНС), 150
- каскад Голлманна, 297
квадратичный вычет, 30
квадратичный закон взаимности, 31, 32
квадратичный конгруэнтный генератор, 266
квадратичный невычет, 30
квантовая криптография, 477
китайская теорема об остатках, 27, 457
класс вычетов, 24
класс стохастических последовательностей по Колмогорову, 139
класс стохастических последовательностей по Чёрчу, 139
класс хаотических последовательностей, 138
классификация состояний ОЦМ, 97
классы вычетов по модулю идеала, 53
ключ, 14
количество информации по Шеннону, 194
кольцо, 51
комбинирование LFSR-генераторов, 280
комбинирование последовательностей, 296
комбинирование с помощью прореживания, 282
комбинирующие генераторы, 278
конгруэнтный генератор с переносом, 267
конгруэнтный генератор со случайными параметрами, 283
конечное поле, 61
корректор Неймана, 263
корреляционный критоанализ, 300
критоанализ, 12
критоанализ на основе повторного использования гаммы, 301
критоанализ на основе стохастических аналогов, 306
критоанализ поточных шифров, 300
критоаналитические задачи, 290
криптография, 12
криптология, 12
криптообразование Вернама, 203
- криптосистема, 12, 14
криптосистема Блюма – Гольдвассера, 402
криптосистема Мак-Элиса, 401
криптосистема Эль-Гамала, 400
криптосистема подстановки-перестановки, 332
криптосистема совершенно криптостойкая, 205
криптостойкий генератор на основе проблемы теории чисел, 275
криптостойкий генератор на основе односторонних функций, 271
криптостойкость, 205
критерий Эйлера, 31
критерий взаимной простоты чисел, 20
критерий возвратности, 96
критерий эргодичности, 98
кручение кривой, 86
- левые смежные классы, 44
линейная рекуррента, 65
линейная рекуррентная последовательность (ЛРП), 65, 292
линейная сложность, 129
линейное разделение секрета, 455
линейное разложение, 21
линейный конгруэнтный генератор, 264
- максимальный идеал, 56
малопараметрические модели цепей Маркова высокого порядка, 106
маргинальное распределение, 89
матрица вероятностей одношаговых переходов, 91, 101
машина Тьюринга, 216
метод квадратичного решета, 428
метод повторного шифрования, 389
метод шифрования RSA, 386
метод эллиптических кривых, 427
минимальный многочлен, 58
многоалфавитное подстановочное шифрование, 17
многочлен деления, 80
модель Джекобса – Льюиса, 102
модель дискретного скользящего среднего порядка q , 106
модель дискретной авторегрессии и скользящего среднего (DARMA(s, q)), 106
модулярный шифр, 15

- мономиальное упорядочение, 469
 мощность теста, 113
 мультиплекативный конгруэнтный генератор, 264
- наибольший общий делитель, 19
 наилучшее приближение второго рода, 38
 наилучшее приближение первого рода, 37
 наименьшее общее кратное, 22
 начальное распределение вероятностей состояний, 94
 незашифрованный текст, 14
 нелинейная рекуррента, 268
 нелинейный конгруэнтный генератор, 266
 необходимое и достаточное условие совершенной криптостойкости, 206
 неприводимый нормированный многочлен, 64
 нечетная подстановка, 49
 нульмерные радикальные идеалы, 473
- обобщенное марковское свойство, 100
 обобщенный метод решета числового поля, 429
 обобщенный покер-тест, 115
 обратный элемент, 42
 общее кратное, 22
 общий делитель, 19
 однородная цепь Маркова (ОЦМ), 95, 165
 однородная цепь Маркова *s*-го порядка (ОЦМ(*s*)), 101
 односторонняя функция, 231
 односторонняя функция с секретом, 406
 операция Монтгомери, 241
 оптимизация функционала энтропии, 179
 открытый канал связи, 309
 открытый ключ, 386
 ОЦМ неразложимая, 96
 ОЦМ(*s*) эргодическая, 101
 ОЦМ(*s, r*) эргодическая, 105
- первообразный корень, 28
 перестановка символов с периодом, 201
 перестановочный (транспозиционный) шифр, 17
 подгруппы групп, 43
 подполе, 58
 подстановка символов алфавита, 201
- подтверждение ключа, 443
 подходящие дроби, 36
 поле, 52
 поле разложения, 60
 полиграммное шифрование, 16
 полиномиальное комбинирование, 280
 полная система вычетов, 24
 пороговая схема, 461
 порядки неприводимых многочленов, 63
 порядок группы, 43
 порядок многочлена, 63
 порядок элемента, 45
 последовательность максимального периода, 67
 поточные криптосистемы, 286, 287
 правые смежные классы, 44
 преобразование Фурье, 119
 приведенная система вычетов, 25
 примитивные триномы, 293
 примитивный элемент расширения, 59
 программный генератор РРСП, 260
 произвольная структура доступа, 467
 промежуточный секрет, 458
 прореживающий генератор, 298
 простая цепь Маркова, 100
 простое алгебраическое расширение, 59
 простое число, 22
 простой идеал, 56
 пространство зашифрованных сообщений, 14
 пространство ключей, 14
 пространство сообщений, 14
 протокол Диффи – Хеллмана, 434
 псевдослучайная последовательность, 108
- равномерно распределенная случайная последовательность (РРСП), 92, 258
 разложение группы по подгруппе, 44
 расписание ключей, 314
 расширенный алгоритм Евклида, 21
 регистр сдвига с линейной обратной связью, 268
 регистр сдвига с обратной связью, 66
 регистр сдвига с одной обратной связью, 268
 редукция Барретта, 241
 режим «Счетчик», 290
 режим CFB, 290
 режим OFB, 290

- рекуррентная последовательность порядка n , 291
рюкзачный метод шифрования, 397
- самосинхронизирующаяся поточная криптосистема, 289
сбалансированное отображение, 453
свойства количества информации, 194
свойства обобщенной энтропии, 172
свойства функционала энтропии, 152
свойство иерархической аддитивности, 155
секретный канал связи, 310
сертификат открытого ключа, 438
символ Лежандра, 30
символ Якоби, 33, 255
симметрическая группа, 48
симметричная криптосистема, 286
синхронная поточная криптосистема, 289
система сравнений первой степени, 27
случайная последовательность по Мартин-Лёфу, 138
совершенная CPC, 452
совершенные модулярные схемы, 464
спектральная плотность, 91
спектральный тест, 119
статистическое оценивание начального состояния ЛРП, 298
стационарное распределение вероятностей ОЦМ, 97
стеганография, 480
структура доступа, 450
структура отказа, 450, 468
суперсингулярная эллиптическая кри- вая, 423
схема Асмута – Блума, 464
схема Миньотта, 457
схема Шамира, 453
схема ЭЦП Рабина, 408
схема ЭЦП Эль-Гамала, 409
схема независимых испытаний, 100
схема разделения секрета (CPC), 450
- таблица Кэли, 43
тактовая подстановка, 314
текст, 14
теорема Стратоновича, 189
теорема о высоковероятном подмножестве, 185
- теорема о циклических подклассах, 97
теорема солидарности, 97
теоретико-информационные оценки стойкости симметричных криптосистем, 205
теория разделения секрета, 450
тест n -серий, 113
тест «собирателя купонов», 116
тест Миллера – Рабина, 245
тест Соловая – Штассена, 245
тест выявления марковской зависимости, 140
тест интервалов, 114
тест максимального дельта-произведения, 136
тест максимального скалярного произведения, 132
тест на основе MTD-модели, 142
тест на основе алгоритма сжатия Лемпеля – Зива, 128
тест на основе линейной сложности, 129
тест на основе модели Джекобса – Льюиса, 140
тест на основе приращений энтропии, 126
тест на основе цепей Маркова с частичными связями, 143
тест пересекающихся n -грамм, 117
тест перестановок, 116
тест случайного блуждания, 123
тест, основанный на рангах двоичных матриц, 119
точка эллиптической кривой, 74
- удельная энтропия, 160
удельная энтропия гауссского стационарного случайного процесса, 178
универсальный алгоритм статистического тестирования, 111
универсальный статистический тест Маурера, 124
условная дифференциальная энтропия, 172
условная энтропия, 154
- факторизация модуля, 390
факторкольцо, 53
физический генератор случайных последовательностей, 260
фильтрующий генератор, 278
формула Колмогорова – Чепмена, 95
функция Эйлера, 25

- функция с лазейкой, 233
функция хэширования, 369
- характеристика области целостности, 53
характеристический многочлен ЛРП, 292
- цепная дробь, 31
цепь Маркова (ЦМ), 94
цепь Маркова переменного порядка, 106
цепь Маркова порядка s (ЦМ(s)), 100
цепь Маркова с частичными связями
 ЦМ(s, r), 105
циклическая группа, 43
циклическая подгруппа, 45
- четная подстановка, 49
число Мерсенна, 256
число Кармайкла, 244
- шаговая функция хэширования, 373
шенноновская модель крипtosистемы, 199
шифр Бофора, 203
шифр Вернама, 18
шифр Виженера, 17
шифр Виженера и его модификации, 202
шифр Плейфера, 16
шифр Юлия Цезаря, 14, 15
шифр модульного гаммирования, 289
- шифр подстановки, 15
шифр простой подстановки, 15
шифр табличного гаммирования, 288
- электронная цифровая подпись (ЭЦП), 406
эллиптическая кривая, 74
эндоморфизм Фробениуса, 86
эндоморфизм кривой, 84
энтропийная устойчивость случайных
 символьных последовательностей, 189
энтропия ИДС, 152
энтропия ИНС, 170
энтропия Хартли, 153
энтропия Шеннона, 153
энтропия марковского ИДС, 167
- ядро гомоморфизма, 47
- MTD-модель Рафтери, 104
- RSA-алгоритм генерации псевдослучайных
 последовательностей, 275
- j -инвариант эллиптической кривой, 75, 79, 422
- m -последовательность, 292
- S*-блок, 318

ОГЛАВЛЕНИЕ

Предисловие.....	7
Основные обозначения	9

Часть I. МАТЕМАТИЧЕСКИЕ И КОМПЬЮТЕРНЫЕ ОСНОВЫ КРИПТОЛОГИИ

Глава 1. Введение в криптологию	12
1.1. Предмет криптологии	12
1.2. История развития криптологии	13
1.3. Задачи криптографии и криптоанализа	14
1.4. Задания	18
Глава 2. Арифметические основы	19
2.1. Алгоритм деления с остатком	19
2.2. Наибольший общий делитель	19
2.3. Взаимно простые числа	20
2.4. Расширенный алгоритм Евклида	21
2.5. Наименьшее общее кратное	22
2.6. Простые числа	22
2.7. Сравнения.....	23
2.8. Классы вычетов	24
2.9. Функция Эйлера	25
2.10. Сравнения первой степени	26
2.11. Система сравнений первой степени	27
2.12. Первообразные корни	28
2.13. Существование первообразных корней	29
2.14. Индексы по модулям p^k и $2p^k$	30
2.15. Символ Лежандра.....	30
2.16. Квадратичный закон взаимности.....	31
2.17. Символ Якоби.....	33
2.18. Цепные дроби	34
2.19. Подходящие дроби.....	35
2.20. Подходящие дроби в качестве наилучших приближений.....	37
2.21. Задания	39
Глава 3. Алгебраические основы.....	42
3.1. Понятие группы	42
3.2. Подгруппы групп.....	43

3.3. Циклические группы	45
3.4. Гомоморфизмы групп	46
3.5. Группы подстановок	48
3.6. Действие группы на множестве	50
3.7. Кольца и поля	51
3.8. Подкольца	52
3.9. Гомоморфизмы колец	54
3.10. Евклидовы кольца	55
3.11. Простые и максимальные идеалы	56
3.12. Конечные расширения полей	58
3.13. Поле разложения	60
3.14. Конечные поля	61
3.15. Порядки неприводимых многочленов	63
3.16. Число неприводимых многочленов	64
3.17. Линейные рекуррентные последовательности	65
3.18. Последовательности максимального периода	67
3.19. Задания	68
 Глава 4. Эллиптические кривые	74
4.1. Уравнение Вейерштрасса эллиптической кривой	74
4.2. j -инвариант и дискриминант эллиптической кривой	75
4.3. Сложение точек эллиптической кривой	76
4.4. Эллиптические кривые над конечными полями нечетной характеристики	78
4.5. Эллиптические кривые над конечными полями характеристики 2	79
4.6. Многочлены деления	79
4.7. Изогении и эндоморфизм Фробениуса	82
4.8. Вычисление порядка группы точек эллиптической кривой	84
4.9. Задания	87
 Глава 5. Вероятностные модели случайных последовательностей	88
5.1. Дискретные временные ряды, их модели и вероятностные характеристики	88
5.2. Равномерно распределенная случайная последовательность и ее свойства	92
5.3. Цепь Маркова и ее свойства	94
5.4. Цепь Маркова порядка s	100
5.5. Модель Джекобса – Льюиса	102
5.6. MTD-модель Рафтери	104
5.7. Цепь Маркова с частичными связями $\text{ЦМ}(s, r)$	105
5.8. Другие малопараметрические модели цепей Маркова высокого порядка	106
5.9. Задания	107

Глава 6. Статистическое тестирование случайных и псевдослучайных последовательностей	108
6.1. Проблема статистического тестирования в криптологии и батареи тестов...	108
6.2. Универсальный алгоритм статистического тестирования случайных и псевдослучайных последовательностей	111
6.3. Тест n -серий	113
6.4. Тест интервалов	114
6.5. Обобщенный покер-тест	115
6.6. Тест «собирателя купонов»	116
6.7. Тест перестановок	116
6.8. Тест пересекающихся n -грамм	117
6.9. Тест, основанный на рангах двоичных матриц	118
6.10. Спектральные тесты.....	119
6.11. Тесты случайного блуждания	123
6.12. Универсальный статистический тест Маурера	124
6.13. Тесты на основе приращений энтропии	126
6.14. Тест, основанный на алгоритме сжатия Лемпеля – Зива	128
6.15. Тест, основанный на линейной сложности.....	129
6.16. Тест на основе экстремальной статистики скалярного произведения	131
6.17. Тест на основе экстремальной статистики дельта-произведения.....	134
6.18. Об алгоритмическом определении случайности.....	136
6.19. Тест выявления марковской зависимости	140
6.20. Тест на основе модели Джекобса – Льюиса	140
6.21. Тест на основе MTD-модели	142
6.22. Тест на основе цепей Маркова с частичными связями.....	143
6.23. Задания	145
Глава 7. Методы теории информации в криптологии	150
7.1. Источники дискретных сообщений и их вероятностные модели	150
7.2. Функционал энтропии и его свойства	152
7.3. Условная энтропия и ее свойства.....	154
7.4. Удельная энтропия стационарной символьной последовательности.....	160
7.5. Энтропийные характеристики марковских символьных последовательностей.....	165
7.6. Источники непрерывных сообщений и их энтропийные свойства	170
7.7. Оптимизация функционала энтропии на классе вероятностных распределений	179
7.8. Асимптотические свойства стационарного источника дискретных сообщений.....	184
7.9. Энтропийная устойчивость случайных символьных последовательностей	189
7.10. Количество информации по Шенону и его свойства	193
7.11. Шеноновские модели криптосистем	199
7.12. Теоретико-информационные оценки стойкости симметричных криптосистем	205
7.13. Задания	210

Глава 8. Элементы теории сложности вычислений	214
8.1. Вычислительные задачи	214
8.2. Задачи распознавания и поиска	215
8.3. Машина Тьюринга	216
8.4. Разрешимые и неразрешимые задачи.....	218
8.5. Ресурсы.....	219
8.6. Вероятностные машины	221
8.7. Алгоритмы Лас-Вегас и Монте-Карло	223
8.8. Сведение.....	225
8.9. Классы сложности.....	227
8.10. Язык PRIMES.....	229
8.11. Односторонние функции	231
8.12. Функции с лазейкой.....	233
8.13. Функция Рабина	234
8.14. Задания.....	235

Глава 9. Базовые алгоритмы	239
9.1. Алгоритмы арифметики больших чисел	239
9.2. Операция Монтгомери и редукция Барретта	241
9.3. Вероятностные и детерминированные алгоритмы тестирования на простоту	242
9.4. Построение больших простых чисел.....	247
9.5. Алгоритмы сложения точек эллиптической кривой	249
9.6. Вычисление кратной точки эллиптической кривой.....	253
9.7. Задания	254

Часть II. КРИПТОГРАФИЧЕСКИЕ СИСТЕМЫ И МЕТОДЫ КРИПТОАНАЛИЗА

Глава 10. Генерация случайных и псевдослучайных последовательностей ...	258
10.1. Классификация алгоритмов генерации	258
10.2. Физические генераторы случайных последовательностей	260
10.3. Линейные и мультипликативные конгруэнтные генераторы	264
10.4. Нелинейные конгруэнтные генераторы	266
10.5. Генераторы на основе регистров сдвига с линейной, нелинейной и случайной обратной связью	267
10.6. Криптостойкие генераторы на основе односторонних функций.....	271
10.7. Криптостойкие генераторы, основанные на проблемах теории чисел.	275
10.8. Методы «улучшения» свойств элементарных псевдослучайных последовательностей	276
10.9. Фильтрующие генераторы	278
10.10. Комбинирование алгоритмов генерации методом Макларена – Марсальи	279

10.11. Комбинирование LFSR-генераторов	280
10.12. Конгруэнтный генератор со случайными параметрами	283
10.13. Задания	283
Глава 11. Поточные криптосистемы	286
11.1. Основные понятия и классификация поточных криптосистем.....	286
11.2. Рекуррентные последовательности и регистры сдвига	291
11.3. Алгоритмы Берлекэмпа – Месси и линейная сложность.....	294
11.4. Комбинирование последовательностей	296
11.5. Статистическое оценивание начального состояния ЛРП	298
11.6. Криptoанализ поточных шифров.....	300
11.7. Задания	306
Глава 12. Блочные криптосистемы.....	309
12.1. Блочное шифрование	309
12.2. Задачи криptoанализа	311
12.3. Блочно-итерационные криптосистемы	313
12.4. Операции над двоичными словами.....	315
12.5. Булевы функции и отображения	318
12.6. Криптосистемы подстановки-перестановки	332
12.7. Криптосистема AES	334
12.8. τ -инволютивные подстановки	336
12.9. Криптосистемы Фейстеля.....	338
12.10. Криптосистема Belt	340
12.11. Атака «грубой силой».....	342
12.12. Разностная атака	346
12.13. Линейная атака	351
12.14. Режимы шифрования	355
12.15. Имитозащита	357
12.16. Задания	360
Глава 13. Функции хэширования	369
13.1. Определение и использование	369
13.2. Задачи криptoанализа	371
13.3. Блочно-итерационные функции хэширования	373
13.4. Шаговые функции хэширования.....	375
13.5. Атака «дней рождения»	377
13.6. Модернизированная атака «дней рождения»	380
13.7. Задания	383
Глава 14. Криптосистемы с открытым ключом.....	386
14.1. RSA-криптосистема	386
14.2. Возможные атаки на криптосистему RSA	388

14.3. Стойкость RSA против атаки повторного шифрования	389
14.4. Поиск секретного ключа d и факторизация модуля N	390
14.5. Биты ключей в RSA-крипtosистеме	392
14.6. Теорема М. Винера о малой секретной экспоненте	393
14.7. Об одном обобщении RSA-крипtosистемы	394
14.8. Рюзачный метод шифрования	397
14.9. Стойкость рюзачного шифра	399
14.10. Крипtosистема Эль-Гамаля	400
14.11. Крипtosистема Мак-Элиса	401
14.12. Крипtosистема Блюма – Гольдвассер	402
14.13. Задания	404
 Глава 15. Электронная цифровая подпись.....	 406
15.1. Обобщенная модель ЭЦП.....	406
15.2. Схема ЭЦП Рабина.....	408
15.3. Схема ЭЦП Эль-Гамаля.....	409
15.4. ЭЦП DSS	410
15.5. ЭЦП ГОСТ Р 34.10-94.....	412
15.6. Эквивалентность задач фальсификации подписи в DSS и схеме Эль-Гамаля	419
15.7. ЭЦП СТБ 1176.2-99	420
15.8. Цифровая подпись на эллиптических кривых	422
15.9. Особенности скалярного умножения на эллиптических кривых	425
15.10. Критоанализ алгоритмов ЭЦП, основанных на факторизации и дискретном логарифмировании	426
15.11. Задания	431
 Глава 16. Протоколы формирования общего ключа	 433
16.1. Головоломки Меркля	433
16.2. Протокол Диффи – Хеллмана	434
16.3. Атака «противник посередине»	436
16.4. Сертификаты открытых ключей	437
16.5. Протокол с сертификатами	439
16.6. Протоколы МТІ	440
16.7. Аутентификация	443
16.8. Протокол MQV	445
16.9. Протокол TLS	446
16.10. Задания	448
 Глава 17. Методы и алгоритмы разделения секрета.....	 450
17.1. Общая задача о разделении секрета	450
17.2. Критерии качества схем разделения секрета.....	452
17.3. Схема Шамира	453

17.4. Линейное разделение секрета	455
17.5. Модулярный подход	457
17.6. Генерация модулей для пороговых схем в кольце целых чисел.....	458
17.7. Пороговые схемы над кольцом многочленов	461
17.8. Совершенные модулярные схемы	464
17.9. Модулярная реализация произвольных структур доступа	467
17.10. Разделение секрета в кольце многочленов от нескольких переменных.....	468
17.11. Реализация произвольных структур доступа.....	470
17.12. Максимальные идеалы одинаковых степеней.....	471
17.13. Нульмерные радикальные идеалы.....	473
17.14. Об идеальных схемах в кольце многочленов от нескольких переменных ...	474
17.15. Задания	476
 Глава 18. О новых направлениях в криптологии	 477
18.1. О возможностях квантовой криптографии	477
18.2. Стеганография и ее применение	480
18.3. Активный криптоанализ	483
 Библиографические ссылки	 487
 Приложения	 496
1. Архив дискретных последовательностей	496
2. Таблицы.....	498
 Предметный указатель	 500

Учебное издание

Классическое университетское издание

**Харин Юрий Семенович
Агиевич Сергей Валерьевич
Васильев Денис Владимирович
Матвеев Геннадий Васильевич**

КРИПТОЛОГИЯ

Учебник

Редактор *Е. В. Павлова*
Художник обложки *Т. Ю. Таран*
Художественный редактор *Т. Ю. Таран*
Технический редактор *Т. К. Раманович*
Компьютерная верстка *О. А. Куцепаловой, О. Г. Кадуриной*
Корректор *С. А. Бондаренко*

Подписано в печать 30.12.2013. Формат 70×100/16. Бумага офсетная.
Печать офсетная. Усл. печ. л. 41,28. Уч.-изд. л. 38,9. Тираж 250 экз. Заказ 53.

Белорусский государственный университет.
ЛИ № 02330/0494425 от 08.04.2009.
Пр. Независимости, 4, 220030, Минск.

Республиканскоe унитарное предприятие
«Издательский центр Белорусского государственного университета».
ЛП № 02330/0494178 от 03.04.2009.
Ул. Красноармейская, 6, 220030, Минск.