

Оглавление

- [1 Сущность проблемы информационной безопасности систем. Характеристики и параметры ИС и ИВС.](#)
- [2 Характеристика угроз безопасности современным ИС и ИВС.](#)
- [3 Политика безопасности современных ИС и ИВС](#)
- [4 Энтропия источника сообщения. Энтропия Шеннона.](#)
- [5 Энтропия источника сообщения. Энтропия Хартли.](#)
- [6 Двоичный канал передачи информации.](#)
- [7 Энтропия двоичного алфавита.](#)
- [8 Условная энтропия. Энтропийная оценка потерь при передаче информации.](#)
- [9 Базовые понятия криптографии. Основы теории больших чисел. Проблема дискретного логарифма.](#)
- [10 Основная теорема арифметики. Алгоритм Евклида нахождения НОД](#)
- [11 Основы модулярной арифметики. Вычеты.](#)
- [12 Обратные вычисления по модулю в криптографии. Расширенный алгоритм Евклида.](#)
- [13 Функция Эйлера в криптографии.](#)
- [14 Хеш-функция и ее свойства. Области использования хеш-функций.](#)
- [15 Общая характеристика алгоритмов хеширования классов MD и SHA.](#)
- [16 Алгоритмы хеширования класса MD. Области использования.](#)
- [17 Алгоритмы хеширования класса SHA. Области использования.](#)
- [18 Общая классификация криптографических методов защиты информации.](#)
- [19 Подстановочные шифры. Шифр Цезаря.](#)
- [20 Особенности реализации шифровальной машины Энигма.](#)
- [21 Шифр на основе аффинной системы подстановок Цезаря.](#)
- [22 Система шифрования Цезаря с ключевым словом.](#)
- [23 Шифр Виженера.](#)
- [24 Перестановочные шифры.](#)
- [25 Методы симметричного криптовреобразования. Стандарт DES. Общая характеристика.](#)
- [26 Методы симметричного криптовреобразования. Стандарт DES. Структура одного цикла.](#)
- [Криптостойкость алгоритма.](#)
- [27 Методы симметричного криптовреобразования. Стандарты 3DES. Реализация и криптостойкость.](#)
- [28 Шифровальная машина Энигма. Устройство, функционирование, криптостойкость.](#)

- 29 Сравнительная характеристика алгоритмов Lucifer, IDEA, ГОСТ 28147-89, Blowfish.
- 30 Криптографические системы с открытым (публичным) ключом. Задача об укладке ранца.
- 31 Управление криптографическими ключами. Алгоритм рукопожатия.
- 32 Распределение ключей на основе симметричных систем.
- 33 Алгоритм передачи ключа по Диффи-Хеллману.
- 34 Алгоритм шифрования RSA. Реализация и криптостойкость.
- 35 Алгоритм шифрования Эль-Гамаля. Реализация и криптостойкость.
- 36 Потоковое шифрование. Типы. Гаммирование в потоковом шифровании.
- 37 Генерация ключевой информации для потокового шифрования. Генераторы ПСП на основе регистров сдвига.
- 38 Особенность шифра Вернама.
- 39 Стеганографические методы защиты информации. Классификация и области использования. Метод наименее значащих бит.
- 40 Текстовая стеганография. Многоключевая модель стеганографической системы.
- 41 Понятие эллиптической кривой. Принципы построения криптосистемы на эллиптических кривых
- 42 Представление и описание эллиптической кривой на основе алгебраической геометрии
- 43 Арифметические операции в эллиптической криптографии
- 44 Система согласования криптографических ключей на основе эллиптической кривой
- 45 ЭЦП. Назначение и свойства.
- 46 ЭЦП. Основные методы генерации. Атаки на ЭЦП
- 47 ЭЦП на основе симметричной криптографии
- 48 ЭЦП на основе алгоритма RSA
- 49 ЭЦП на основе симметричной криптосистемы и посредника
- 50 ЭЦП DSS.
- 51 ЭЦП на основе алгоритма Эль-Гамаля
- 52 ЭЦП на основе эллиптической кривой.
- 53 Алгоритм К. Шнорра. Стандарт ЭЦП в РБ.
- 54 Протокол Kerberos.
- 55 Деструктивные программы. Классификация и методы нейтрализации.
- 56 Оценка безопасности парольной защиты.

1 Сущность проблемы информационной безопасности систем. Характеристики и параметры ИС и ИВС.

**Сущность понятия
«информационная безопасность»**

- **информационная безопасность** — это состояние защищённости информационной среды
- **безопасность информации (при применении информационных технологий)** (англ. *IT security*) включает:
 - a) состояние защищенности информации (данных) от несанкционированного доступа к ней и от влияния дестабилизирующих факторов;
 - b) информационную безопасность автоматизированной информационной системы, в которой она реализована.

2

Информационная (информационно-вычислительная) система (ИС, ИВС) – организационно упорядоченная совокупность документов, технических средств и информационных технологий, реализующая информационные (информационно-вычислительные) процессы.

Безопасность ИВС – свойство системы, выражющееся в способности системы противодействовать попыткам несанкционированного доступа или нанесения ущерба владельцам и пользователям системы при различных умышленных и неумышленных воздействиях на нее.

информационная безопасность – все аспекты, связанные с определением, достижением и поддержанием конфиденциальности, целостности, доступности информации или средств ее обработки:

- 1) конфиденциальность (confidentiality) – состояние информации, при котором доступ к ней осуществляют только субъекты, имеющие на нее право;
- 2) целостность (integrity) – избежание несанкционированной модификации информации;
- 3) доступность (availability) – избежание временного или постоянного сокрытия информации от пользователей, получивших права доступа.

Основные характеристики элементов ИС

Определение 1. Алфавит – конечная совокупность символов (знаков), с помощью которых можно представить

любое сообщение в ИС: **A{a_i}**

a_i – i- й символ алфавита

Определение 2. Мощность алфавита – количество символов, составляющих алфавит: **N(A)**

Вероятность того, что произвольный символ ξ произвольного документа (текст, база данных, текст программы) будет буквой « a_i »: **P($\xi = a_i$) = p(a_i)**

$$\sum_{i=1}^N P(a_i) = 1 \quad (1)$$

Фактор, действующий на ИВС, – это явление, действие или процесс, результатом которых может быть утечка, искажение, уничтожение данных, блокировка доступа к ним, повреждение или уничтожение системы защиты.

Все многообразие дестабилизирующих факторов можно разделить на два класса: внутренние и внешние.

Внутренние дестабилизирующие факторы влияют:

- 1) на программные средства (ПС):
 - а) некорректный исходный алгоритм;

Внешние дестабилизирующие факторы влияют:

- 1) на программные средства:
 - а) неквалифицированные пользователи;
 - б) несанкционированный доступ к ПС с целью модификации кода;

2 Характеристика угроз безопасности современным ИС и ИВС.

Информационная (информационно-вычислительная) система (ИС, ИВС) – организационно упорядоченная совокупность документов, технических средств и информационных технологий, реализующая информационные процессы.

Одним из важнейших дестабилизирующих работу ИВС факторов считают электромагнитные и ионизирующие излучения. Ионизирующие излучения также могут иметь естественную (солнечная радиация) и искусственную (изотопы урана и тория излучают даже пластмассы) природу.

Основные факторы (угрозы):

- 1) действия злоумышленника;
- 2) наблюдение за источниками информации;
- 3) подслушивание конфиденциальных разговоров и акустических сигналов работающих механизмов;
- 4) перехват электрических, магнитных и электромагнитных полей, электрических сигналов и радиоактивных излучений;
- 5) несанкционированное распространение материальных носителей за пределами организации;
- 6) разглашение информации компетентными людьми;

- 7) потеря носителей информации;
- 8) несанкционированное распространение информации через поля и электрические сигналы, случайно возникшие в аппаратуре;
- 9) воздействие стихийных сил (наводнения, пожары и т. п.);
- 10) сбои и отказы в аппаратуре сбора, обработки и передачи информации;
- 11) отказы системы электроснабжения;
- 12) воздействие мощных электромагнитных и электрических помех (промышленных и природных).

Несанкционированный доступ с помощью деструктивных программных средств осуществляется, как правило, через компьютерные сети.

Классификацию вредоносного ПО можно представить следующим образом: 1) вирусы (viruses); 2) черви (worms); 3) кейлоггеры (keyloggers); 4) трояны (trojans); 5) боты (bots); 6) снiffeры (sniffers); 7) руткиты (rootkits).

Выделяют несколько основных угроз безопасности, возникающих при использовании бесплатных точек доступа Wi-Fi: 1) сети, организованные хакерами, могут выдавать себя за вполне легальные бесплатные точки доступа; 2) атаки с помощью вредоносного ПО компьютера, подключенного к этой точке доступа; 3) снiffинг; 4) хищение персональной информации методом «человек посередине» (man in the middle); 5) Фишинг.

3 Политика безопасности современных ИС и ИВС

Информационная (информационно-вычислительная) система (ИС, ИВС) – организационно упорядоченная совокупность документов, технических средств и информационных технологий, реализующая информационные (информационно-вычислительные) процессы.

Для построения политики информационной безопасности рассматривают следующие направления защиты ИС:

- защита объектов ИС;
- защита процессов, процедур и программ обработки информации;
- защита каналов связи;
- подавление побочных электромагнитных излучений;
- управление системой защиты.

Политика информационной безопасности систем, как и во всех подобных случаях, должна строиться на основе системного подхода, предусматривающего всесторонний анализ причин и угроз безопасности, оценки их последствий, необходимости, экономической или иной целесообразности и адекватности принимаемых противодействий.

Методы и средства защиты можно разделить на три класса:

- 1)законодательная и нормативно-правовая база;**

1. Акты национального законодательства: а) международные договоры Республики Беларусь; б) Конституция Республики Беларусь; в) законы Республики Беларусь, например Закон Республики Беларусь от 10 ноября 2008 г. № 455-З «Об информации, информатизации и защите информации»; г) указы Президента Республики Беларусь; д) постановления Правительства Республики Беларусь; е) нормативные правовые акты министерств и ведомств; ж) нормативные правовые акты субъектов, органов местного самоуправления и т. д.

2. Международные стандарты, например: а) BS 7799-1:2005 – Британский стандарт BS 7799 Part 1 – Code of Practice for Information Security Management; б) BS 7799-2:2005 – Британский стандарт BS 7799 Part 2 – Information Security Management – Specification for Information Security Management Systems; в) ISO/IEC 17799:2005; г) ISO/IEC 27001:2005; д) ISO/IEC 27002; е) ISO/IEC 27005; ж) ISO/IEC 27040:2015.

2) организационно-технические и режимные меры и методы (политика информационной безопасности);

- организацию охраны, режима, работу с кадрами, с документами;
- использование технических средств безопасности (например, простейших дверных замков, магнитных или иных карт и др.), информационно-аналитическую деятельность по выявлению внутренних и внешних угроз.

3) аппаратные, программно-аппаратные и программные способы и средства обеспечения ИБ.

1) средства защиты от несанкционированного доступа:

а) средства авторизации;

б) аудит;

2) системы мониторинга:

а) системы мониторинга сетей;

б) анализаторы протоколов;

3) антивирусные средства:

а) антивирусные программы;

б) программные и иные антиспамовые средства;

в) межсетевые экраны;

4) криптографические средства:

а) шифрование данных

; б) электронная цифровая подпись;

- 5) системы бесперебойного питания;
- 6) системы аутентификации: а) пароль; б) ключ доступа (физический или электронный);
в) биометрия (анализаторы отпечатков пальцев, анализаторы сетчатки глаза, анализаторы голоса, анализаторы геометрии ладони и др.).

4 Энтропия источника сообщения. Энтропия Шеннона.

Алфавит - это набор знаков или символов, используемых для генерации и передачи сообщения

Во всех конечных алфавитах частота или вероятность появления символа в производном документе различна.

Понятие энтропия ввел Клод Шеннон, информация оценивается количественно, показывает какое кол-во информации(бит) приходится в среднем на один символ алфавита(сообщения).

Энтропия К. Шеннона рассчитывают по следующей формуле: Сумма вероятностей умноженная на логарифм по основанию 2

$$H = -\sum P(i) \cdot \log_2 P(i)$$

Чтобы вычислить энтропию по Шеннону нужно знать мощность и вероятность появления символа.

Энтропия по Шеннону предполагает наличие разных вероятностей появления символов.

Условная энтропия – это кол-во потерянной информации, приходящейся на 1 символ сообщения

Эффективная энтропия – 1 - условная энтропия (показывает кол-во реально переданных данных)

Источник передает элементарные сигналы к различных типов. Проследим за достаточно длинным отрезком сообщения. Пусть в нем имеется N1 сигналов первого типа, N2 сигналов второго типа, ..., Nk сигналов k-го типа, причем N1 + N2 + ... + Nk = N – общее число сигналов в наблюдаемом отрезке, f1, f2, ..., fk – частоты соответствующих сигналов. При возрастании длины отрезка сообщения каждая из частот стремится к фиксированному пределу, т.е.

$$\lim f_i = p_i, (i = 1, 2, \dots, k),$$

где p_i можно считать вероятностью сигнала. Предположим, получен сигнал i-го типа с вероятностью p_i , содержащий $-\log p_i$ единиц информации. В рассматриваемом отрезке i-й сигнал встретится примерно Np_i раз (будем считать, что N достаточно велико), и общая информация, доставленная сигналами этого типа, будет равна произведению $Np_i \log p_i$. То же относится к сигналам любого другого типа, поэтому полное количество информации, доставленное отрезком из N сигналов, будет примерно равно

Чтобы определить среднее количество информации, приходящееся на один сигнал, т.е. удельную информативность источника, нужно это число разделить на N. При неограниченном росте приблизительное равенство перейдет в точное. В результате будет получено асимптотическое соотношение – формула Шеннона

формула, предложенная Хартли, представляет собой частный случай более общей формулы Шеннона. Если в формуле Шеннона принять, что $p_1 = p_2 = \dots = p_i = \dots = p_N = 1/N$, то

Знак минус в формуле Шеннона не означает, что количество информации в сообщении – отрицательная величина. Объясняется это тем, что вероятность p , согласно определению, меньше единицы, но больше нуля. Так как логарифм числа, меньшего единицы, т.е. $\log p_i$ – величина отрицательная, то произведение вероятности на логарифм числа будет положительным.

Кроме этой формулы, Шенноном была предложена абстрактная схема связи, состоящая из пяти элементов (источника информации, передатчика, линии связи, приемника и адресата), и сформулированы теоремы о пропускной способности, помехоустойчивости, кодировании и т.д.

Pigeon Pavel Andreevich
Rudenia Pavel Andreevich

$$H(A) = 4,7 \quad \text{Последний раз было 4,7 из-за}\newline \text{ошибки в формуле, а также что}\newline \text{все в ASCII были единицами}\newline \text{и } p=0,1 - одна позиция}$$

$$H(A) = 4,35$$

$$\text{т. } I(X_k) = 4,7 \cdot 22 = 103,4$$

$$\text{и } I(X_k) = 4,35 \cdot 20 = 87$$

~~$$P=0,5 \cdot 0,08 \quad P=0,5$$~~

$$H(X/Y) = -0,5 \cdot \log_2 0,5 = 0,5 \log_2 0,5$$
$$-0,5 \cdot (-1) - 0,5 \cdot (-1) =$$
$$= 0,5 + 0,5 = 1$$

$$\text{Error: } I(X_k) = 1 \cdot 22 = 22$$

$$I(X_k) = 1 \cdot 20 = 20$$

$$\text{ASCII: } 22 \cdot 8 = 176$$

$$20 \cdot 8 = 160$$

LEVERX GROUP

LeverX emer||

5 Энтропия источника сообщения. Энтропия Хартли.

Частным случаем энтропии Шеннона является энтропия Хартли. Дополнительным условием при этом является то, что все вероятности одинаковы и постоянны для всех символов алфавита. С учетом этого формулу (2.1) можно преобразовать к виду:

$$H_{Ch}(A) = \log_2 N. \quad (2.2)$$

Сообщение X_k , которое состоит из k символов, должно характеризоваться определенным количеством информации, $I(X_k)$:

$$I(X_k) = H(A) * k. \quad (2.3)$$

Здесь $H(A)$ – энтропия алфавита с соответствующим распределением вероятностей $p(a_i)$.

6 Двоичный канал передачи информации.

Обработка информации в вычислительных системах невозможна без передачи сообщений между отдельными элементами (оперативной памятью и процессором, процессором и внешними устройствами). Общая схема передачи информации показана на рис.7.1.

В канале сигнал подвергается различным воздействиям, которые мешают процессу передачи. Воздействия могут быть непреднамеренными (вызванными естественными причинами) или специально организованными (созданными) с какой-то целью некоторым противником. Непреднамеренными воздействиями на процесс передачи (помехами) могут являться уличный шум, электрические разряды (в т. ч. молнии), магнитные возмущения (магнитные бури), туманы, взвеси (для оптических линий связи) и т.п.

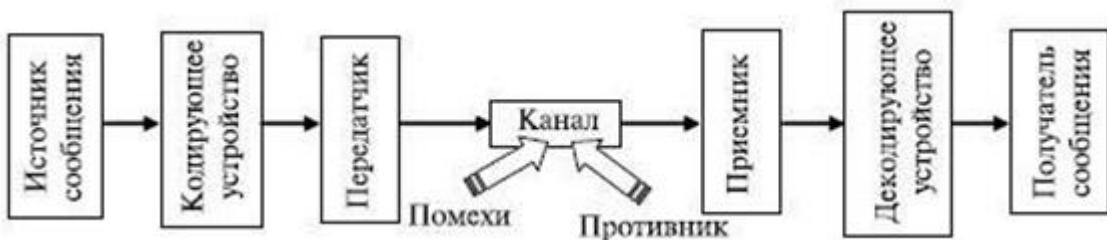


Рис. 7.1. Общая схема передачи информации

Для изучения механизма воздействия помех на процесс передачи данных и способов защиты от них необходима некоторая модель. Процесс возникновения ошибок описывает модель под названием двоичный симметричный канал (ДСК), схема которой показана на рис.7.2.

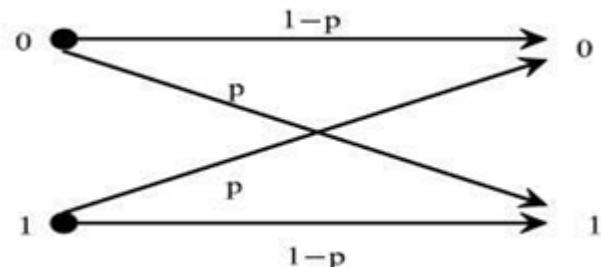


Рис. 7.2. Схема двоичного симметричного канала

При передаче сообщения по ДСК в каждом бите сообщения с вероятностью может произойти ошибка, независимо от наличия ошибок в других битах. Ошибка заключается в замене знака 0 на 1 или 1 на 0.

Некоторые типы ошибок:

замена знака 0 на 1 или 1 на 0 ;

- вставка знака ;
- пропуск знака .

Чаще других встречается замена знака. Этот тип ошибок исследован наиболее полно.

Способы повышения надежности передачи сообщений

Если при кодировании сообщений используются оптимальные коды, то при появлении всего лишь одной ошибки все сообщение или его значительная часть может быть искажена. Рассмотрим пример. Пусть *кодирование* элементарных сообщений источника осуществляется с использованием кодовой таблицы

Тогда закодированное сообщение имеет вид 011011100110. Если в первом знаке произойдет ошибка, то будет принято сообщение 111011100110, которое декодируется в слово . Полное искажение сообщения из-за одной ошибки происходит вследствие того, что одно кодовое слово переходит в другое кодовое слово в результате замены одного или нескольких знаков. Пример показывает, что оптимальное кодирование плохо защищает сообщения от воздействия ошибок.

На практике необходим компромисс между экономностью кода и защитой от ошибок.

Сначала удаляется "бесполезная" избыточность (в основном статистическая), а затем добавляется "полезная" избыточность, которая помогает обнаруживать и исправлять ошибки.

Сообщения	Кодовое слово
a1	00
a2	01
a3	10
a4	110
a5	111

Рассмотрим некоторые методы повышения надежности передачи данных. Широко известными методами борьбы с помехами являются следующие [34]:

- передача в контексте;
- дублирование сообщений;
- передача с переспросом.

Рассмотрим подробней каждый из этих способов.

- Передача в контексте. С этим хорошо известным и общепринятым способом сталкивался каждый, кто, пытаясь передать по телефону с плохой слышимостью чью-либо фамилию, называл вместо букв, ее составляющих, какие-нибудь имена, первые буквы которых составляют данную фамилию. В данном случае правильному восстановлению искаженного сообщения помогает знание его смыслового содержания.
- Дублирование сообщений. Этот способ тоже широко применяется в житейской практике, когда для того, чтобы быть правильно понятым, нужное сообщение повторяют несколько раз.

- Передача с переспросом. В случае, когда получатель имеет связь с источником сообщений, для надежной расшифровки сообщений пользуются переспросом, т. е. просят повторить все переданное сообщение или часть его.

Общим во всех этих способах повышения надежности является введение избыточности, то есть увеличение тем или иным способом объема передаваемого сообщения для возможности его правильной расшифровки при наличии искажений.

Следует отметить, что введение избыточности уменьшает скорость передачи информации, так как только часть передаваемого сообщения представляет интерес для получателя, а избыточная его доля введена для предохранения от шума и не несет в себе полезной информации.

Естественно выбирать такие формы введения избыточности, которые позволяют при минимальном увеличении объема сообщения обеспечивать максимальную помехоустойчивость.

7 Энтропия двоичного алфавита.

$A\{0, 1\}$ - алфавит, $N=2$ - мощность алфавита

Пусть вероятность встречи 0 будет:

$P(\xi = 0) = p(0)$, а вероятность 1 будет $P(\xi = 1) = p(1)$

Энтропии Шеннона:

$$H_S(A) = - \sum_{i=1}^N P(a_i) * \log_2 P(a_i),$$

где $i = \overline{1, N}$, a_i – элемент алфавита, $P(a_i)$ – вероятность $P(\xi = a_i)$.

Энтропия двоичного алфавита (просто подставили в формулу вероятности):

$$H(A2) = -p(0)*\log_2(p(0)) - p(1)*\log_2(p(1))$$

Сумма вероятностей должна быть равна 1

$$p(0) + p(1) = 1$$

Значит можем заменить $p(0) = 1 - p(1)$

Подставляем в уравнение вместо $p(0)$:

$$H(A_2) = -(1-p(1)) \cdot \log_2(1-p(1)) - p(1) \cdot \log_2(p(1))$$

Если я правильно помню, то для решения уравнения нужно решить дифференциальное уравнение:

$$\frac{d H(A_2)}{d p(1)} = 0$$

В итоге (с помощью сложнейших математических вычислений калькулятором) получаем (вместо $p(1)$ будет x):

$$\frac{d}{dx}(-(1-x) \log_2(1-x) - x \log_2(x)) = \frac{\log(1-x) - \log(x)}{\log(2)}$$

И эта вся ботва = 0, когда $\log_2(1-x) = \log_2(x)$

Они равны, когда $1-x=x$

$$1=2*x$$

$$X=0.5$$

Значит вероятность $p(1) = 0.5$, а $p(0) = 1 - 0.5 = 0.5$

Подставляем в первую формулу и вычисляем энтропию:

$$H(A_2) = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1 \text{ (бит)}$$

По формуле Хартли:

$$H_c(A) = \log_2 N$$

$$H(A_2) = \log_2 2 = 1.$$

В общем и целом энтропия двоичного алфавита равна 1

8 Условная энтропия. Энтропийная оценка потерь при передаче информации.

Определение. Условной энтропией Источника дискретного сообщения X называем величину

$$H(X|Y) = P(Y=0)H(X|Y=0) + P(Y=1)H(X|Y=1) = - p \log_2 p - q \log_2 q$$

H(X|Y) означает потерю информации на каждый символ переданного сообщения

Пример 5. Пусть известно, что $P(X=0) = P(X=1) = 0.5$ и $p=0.01$.

Из (14) определим

$$H(X|Y) = - p \log p - q \log q = -0.01 * \log 0.01 - 0.99 * \log 0.99 = 0.081 \text{ бит}$$

Шеннон показал, что эффективная информация на выходе канала относительно входной в расчете на 1 символ (Эфф энтропия алфавита) составляет:

$$H_e = H(X) - H(X|Y)$$

Для случая из примера 5 $H_e = 0.919 \text{ бит}$

9 Базовые понятия криптографии. Основы теории больших чисел. Проблема дискретного логарифма.

Основная теорема арифметики. Всякое натуральное число N , кроме 1, можно представить как произведение простых множителей:

$$N = p_1 \cdot p_2 \cdot p_3 \cdot \dots \cdot p_n, n > 1.$$

Целое число 37 – простое. Целое число $1\ 554\ 985\ 071 = 3 \cdot 3 \cdot 4463 \cdot 38\ 713$ – произведение четырех простых чисел, два из которых совпадают. Или $39\ 616\ 304 = 2 \cdot 13 \cdot 7 \cdot 2 \cdot 23 \cdot 13 \cdot 2 \cdot 13 \cdot 2 \cdot 7 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 7 \cdot 7 \cdot 13 \cdot 13 \cdot 13 \cdot 23$.

Если известны три некоторых числа (a, x, n) , то достаточно легко можно вычислить число y :

$$y = a^x \bmod n. \quad (4.10)$$

Обратная задача: найти x , если известны a, y, n . Эта задача решается гораздо труднее. Ее называют задачей (проблемой) дискретного логарифмирования, по аналогии с вещественными числами, для которых $x = \log_a y$.

Решения существуют не для всех дискретных логарифмов (напомним, что речь идет только о целочисленных решениях).

Рассматриваемые вычисления относятся к числу так называемых односторонних функций.

Односторонняя функция – одно из центральных понятий в асимметричной криптографии.

Наглядным примером односторонней функции может служить разбиение чашки: разбить чашку на мелкие кусочки достаточно просто, однако очень не просто собрать чашку из кусочеков.

10 Основная теорема арифметики. Алгоритм Евклида нахождения НОД

Основная теорема арифметики. Всякое натуральное число N , кроме 1, можно представить как произведение простых множителей:

$$N = p_1 \cdot p_2 \cdot p_3 \cdot \dots \cdot p_n, n > 1.$$

Целое число 37 – простое. Целое число $1\ 554\ 985\ 071 = 3 \cdot 3 \times 4463 \cdot 38\ 713$ – произведение четырех простых чисел, два из которых совпадают. Или $39\ 616\ 304 = 2 \cdot 13 \cdot 7 \cdot 2 \cdot 23 \cdot 13 \cdot 2 \cdot 13 \cdot 2 \cdot 7 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 7 \cdot 7 \cdot 13 \cdot 13 \cdot 13 \cdot 23$.

Общий делитель нескольких целых чисел – это число, на которое делятся все данные числа без остатка. Наибольший из делителей называется *наибольшим общим делителем* (НОД).

Один из способов вычисления НОД двух чисел базируется на *алгоритме Евклида*.

Пусть даны два числа – a и b ; $a > 0$, $b > 0$, считаем, что $a > b$. Находим ряд равенств:

$$\left. \begin{array}{ll} a = b \cdot q_1 + r_1, & 0 < r_1 < b; \\ b = r_1 \cdot q_2 + r_2, & 0 < r_2 < r_1; \\ r_1 = r_2 \cdot q_3 + r_3, & 0 < r_3 < r_2; \\ r_2 = r_3 \cdot q_4 + r_4, & 0 < r_4 < r_3; \\ r_{n-3} = r_{n-2} \cdot q_{n-1} + r_{n-1}, & 0 < r_{n-1} < r_{n-2}; \\ r_{n-2} = r_{n-1} \cdot q_n + r_n, & 0 < r_n < r_{n-1}; \\ r_{n-1} = r_n \cdot q_{n+1}, & r_{n+1} = 0, \end{array} \right\} \quad (4.1)$$

заканчивающийся, когда получаем некоторое $r_{n+1} = 0$. Тогда r_n – наибольший общий делитель чисел a и b .

Последнее неизбежно, так как ряд b, r_1, r_2, \dots , как ряд убывающих целых, не может содержать более чем b положительных. Имеем: $b > r_1 > r_2 > \dots > r_n > 0$, следовательно, процесс оборвется максимум через b шагов.

Пример 4.1. Пусть $a = 525$, $b = 231$. Найти НОД.

Применим алгоритм Евклида:

$$\begin{aligned} 525 &= 231 \cdot 2 + 63; \\ 231 &= 63 \cdot 3 + 42; \\ 63 &= 42 \cdot 1 + 21; \\ 42 &= 21 \cdot 2. \end{aligned}$$

Получаем последний положительный остаток $r_3 = 21$.

Таким образом, НОД (525, 231) = 21.

Пример 4.2. Пусть $a = 1234$, $b = 54$. Найти НОД.

$$\begin{aligned} 1234 &= 54 \cdot 22 + 46; \\ 54 &= 46 \cdot 1 + 8; \\ 46 &= 8 \cdot 5 + 6; \\ 8 &= 6 \cdot 1 + 2; \\ 6 &= 2 \cdot 3. \end{aligned}$$

Последний ненулевой остаток равен 2, поэтому НОД (1234, 54) = 2.

Схема алгоритма приведена на рисунке.

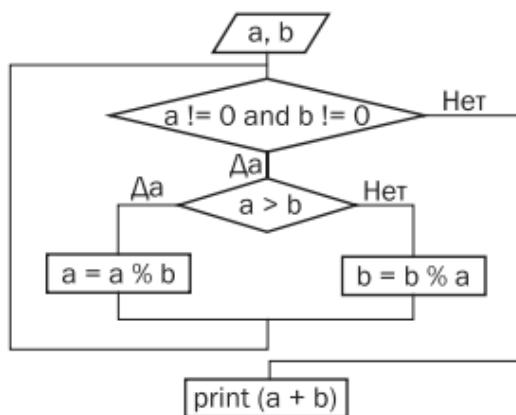


Рис. 3.5. Блок-схема алгоритма Евклида
для вычисления НОД

Взаимно простые числа – это два или несколько целых чисел, наибольший общий делитель которых равен единице. Таким образом, если НОД чисел p и q равен 1, то эти числа называются взаимно простыми. Например, числа 13 и 28 являются взаимно простыми, хотя число 28 не относится к числу простых. Числа 15 и 27 взаимно простыми не являются. Простое число взаимно просто со всеми другими числами, кроме чисел, кратных данному простому числу.

11 Основы модулярной арифметики. Вычеты.

1.

Понятие «модулярная арифметика» ввел немецкий ученый

К. Ф. Гаусс. В этой арифметике мы интересуемся остатком от деления числа a на число n .

Если таким остатком является число b ,

то можно записать:

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

Такая формальная запись читается как « a сравнимо с b по модулю n ». Множество всех чисел, сравнимых с сравнимых с b по модулю n , называется **классом вычетов по модулю n** . Любое число класса называется **вычетом** по модулю n .

При целочисленном (в том числе и нулевом) результате k деления числа a на число n справедливо $a = b + k \cdot n$.

Модулярная арифметика также коммутативна, ассоциативна и дистрибутивна, как и обычная арифметика.

Приведение каждого промежуточного результата по модулю n

дает такой же результат, как приведение всего результата вычисления по модулю n :

$$(a + b) \pmod{n} = ((a \pmod{n}) + (b \pmod{n})) \pmod{n};$$

$$(a - b) \pmod{n} = ((a \pmod{n}) - (b \pmod{n})) \pmod{n};$$

$$(a \cdot b) \pmod{n} = ((a \pmod{n}) \cdot (b \pmod{n})) \pmod{n};$$

$$(a \cdot (b + c)) \pmod{n} = (((a \cdot b) \pmod{n}) + ((a \cdot c) \pmod{n})) \pmod{n}.$$

Модулярная арифметика, как видим, ограничивает диапазон промежуточных и конечного результатов вычислений, т. е. эти

вычисления проще организовать и выполнить на компьютере.

Вычисление степени некоторого числа по модулю другого числа представляет собой последовательность операций умножения и

деления. Однако существуют методы ускорения таких вычислений.

Пример. Нужно вычислить модуль n некоторого числа a в 8 степени:

$$a^8 \pmod{n}.$$

Понятно, что семь операций умножения числа a могут дать огромное число. Порядок чисел, которыми оперирует вычислитель, можно значительно уменьшить, если воспользоваться промежуточными вычислениями по модулю: $((a^2 \pmod{n})^2 \pmod{n})^2 \pmod{n}$.

Пример. Предположим, что показатель степени не является степенью 2. Пусть это будет, например, 25, т. е. необходимо вычислить $a^{25} \pmod{n}$.

После понятных рассуждений последовательность операций можем представить в виде следующей *аддитивной цепочки*:

$$\begin{aligned} a^{25} \pmod{n} &= (a \cdot a^{24}) \pmod{n} = (((((a^2 \cdot a)^2)^2)^2 a) \pmod{n} \\ &= ((((((a^2 \pmod{n}) a) \pmod{n})^2 \pmod{n})^2 \pmod{n})^2 \pmod{n}) a) \pmod{n}. \end{aligned}$$

12 Обратные вычисления по модулю в криптографии. Расширенный алгоритм Евклида.

Вспомним, что обратное значение числа 4 есть $\frac{1}{4}$. Это означает, что их произведение должно равняться 1. В модулярной арифметике обратное значение является понятием более сложным.

Вспомним, что в анализируемой арифметике запись $a \cdot x \equiv 1 \pmod{n}$ эквивалентна $a \cdot x = n \cdot k + 1$,
поиску таких значений x и k , которые удовлетворяли бы тождеству: (4.4)

Последнюю формулу можно представить в таком виде: $x^{-1} = a \pmod{n}$. (4.5)

Число x^{-1} является обратным значением по модулю n числа a .

Следует запомнить:

- 1) уравнения (4.4) и (4.5) имеют единственное решение, если числа x и n являются взаимно простыми;
- 2) если n – простое число, то любое число от 1 до $n - 1$ является взаимно простым с n и имеет только одно обратное значение по модулю n .

Обратное значение по модулю можно вычислить, воспользовавшись расширенным алгоритмом Евклида.

Расширенный алгоритм Евклида находит наибольший общий делитель d чисел a и b и его линейное представление, т. е. целые числа x и y , для которых $ax + by = d$, и не требует «возврата», как в рассмотренном примере. Пусть d – НОД для a и b , т. е. $d = (a, b)$, где $a > b$. Тогда существуют такие целые числа x и y , что $d = ax + by$. Иными словами, **НОД двух чисел можно представить в виде линейной комбинации этих чисел с целыми коэффициентами**.

Алгоритм 3. Схема расширенного алгоритма Евклида.

1. Определить $a_0 = 1$, $a_1 = 0$, $b_0 = 0$, $b_1 = 1$, $\alpha = a$, $\beta = b$.
2. Пусть число q – частное от деления числа a на число b , а число r – остаток от деления этих чисел (т. е. $a = qb + r$).
3. Если остаток от деления r равен нулю, то выполняем шаг 6.
4. Определяем:
$$\begin{aligned} a &= b; \\ b &= r; \\ t &= a_0; \quad //t = x_{i-1} \\ a_0 &= a_1; \\ a_1 &= t - a_1 q; \quad //a_1 = x_i - \text{для правой части} = x_{i+1} \end{aligned}$$
 для правой
$$\begin{aligned} t &= b_0; \quad //t = y_{i-1} \\ b_0 &= b_1; \\ b_1 &= t - b_1 q; \end{aligned}$$
5. Возвращаемся на шаг 2.
6. Определяем $x = x_0$, $y = y_0$, $d = \alpha x + \beta y$.

13 Функция Эйлера в криптографии.

1.

Количество натуральных чисел, меньших некоторого числа n и взаимно простых с ним, можно подсчитать на основе известной функции Эйлера (по имени швейцарского математика Леонарда Эйлера (1707–1783)), иногда называемой «фи-функцией», $\phi(n)$. Например, для числа 24 ($n = 24$) существует 8 взаимно простых с ним чисел (1, 5, 7, 11, 13, 17, 19, 23), поэтому $\phi(24) = 8$.

Если n – простое число, то

$$\phi(n) = n - 1. \quad (4.2)$$

Другие примеры:

$$\begin{array}{lll} \phi(1) = 1; & \phi(5) = 4; & \phi(9) = 6; \\ \phi(2) = 1; & \phi(6) = 2; & \phi(10) = 4; \\ \phi(3) = 2; & \phi(7) = 6; & \phi(11) = 10; \\ \phi(4) = 2; & \phi(8) = 4; & \phi(12) = 4. \end{array}$$

Любое положительное целое число p может быть выражено с помощью положительных целых чисел, не превосходящих и взаимно простых с каждым делителем числа p . Например, $6 = 2 \cdot 3$ имеет четыре делителя: 1, 2, 3 и 6.

$$\phi(1) + \phi(2) + \phi(3) + \phi(6) = 1 + 1 + 2 + 2 = 6.$$

Если $n = p \cdot q$, то

$$\phi(n) = (p - 1) \cdot (q - 1). \quad (4.3)$$

Если числа p и q – взаимно простые, то

$$\phi(p \cdot q) = \phi(p) \cdot \phi(q).$$

Например, пусть $p = 8$ и $q = 15$. Тогда $\phi(8) = 4$, поскольку только 1, 3, 5 и 7 – положительные целые числа, которые меньше 8 и взаимно простые с 8. Также $\phi(15) = 8$, поскольку только 1, 2, 4, 7, 8, 11, 13 и 14 – положительные целые числа, которые меньше 15 и взаимно простые с 15. Следовательно,

$$\phi(120) = \phi(8) \cdot \phi(15) = 32,$$

что можно проверить непосредственно.

С другой стороны, если p и q – очень большие простые числа и известен результат их перемножения (число n), то обратная задача – найти p и q по известному n (*задача факторизации*) – даже для современных вычислительных средств представляется практически неразрешимой. Эта особенность используется, в частности, в некоторых алгоритмах асимметричной криптографии.

14 Хеш-функция и ее свойства. Области использования хеш-функций.

Хеширование (англ. *hashing*) – это преобразование входного массива данных определенного типа и произвольной длины в выходную битовую строку фиксированной длины. Такие преобразования также называются хеш-функциями

или функциями свертки, а их результаты называют *хешем* или *дайджестом* сообщения (англ. message digest).

Хеш-функция – математическая или иная функция, которая принимает на входе строку символов переменной (произвольной) длины и преобразует ее в выходную строку фиксированной (обычно – меньшей) длины, называемой значением хеш-функции.

Хеш-функцию можно использовать для преобразования произвольного входного текста в подходящий формат. Стоит также заметить, что использование хеш-функции не обязательно при вычислении ЭЦП, а сама функция не является частью алгоритма ЭП, поэтому хеш-функция может использоваться любая или не использоваться вообще.

Принято считать, что хорошей, с точки зрения практического применения, является такая хеш-функция, которая удовлетворяет следующим основным условиям:

- функция должна быть простой с вычислительной точки зрения;
- функция должна минимизировать число *коллизий*, т. е. ситуаций, когда разным сообщениям соответствует одно значение хеш-функции.

При этом первое свойство хорошей хеш-функции зависит, в основном, от параметров компьютера, а второе – от значений данных и алгоритма хеширования. В дополнение к приведенным свойствам добавим, пожалуй, важнейшее: свойство *однонаправленности*.

Однонаправленная (или *односторонняя*) хеш-функция предполагает простоту ее вычисления (вычисления $h(x)$ по известному аргументу x) и сложность обратного вычисления (вычисления x по известному $h(x)$).

Однонаправленность – важнейшее свойство многих криптографических алгоритмов.

Обратимся сейчас к коллизиям. Эти, понятные с логической и с семантической точек зрения, термин и явление являются следствием того, что множество возможных сообщений всегда будет превышать множество возможных хеш-функций, поскольку длина последних ограничена (чаще всего эта длина составляет от 128 до 512 битов). Это означает, что коллизии неизбежны, по крайней мере, с теоретической точки зрения. Принято коллизии разделять на два типа.

Коллизией 1-го рода считаем ситуацию, при которой для данного сообщения M и для иного произвольного сообщения M' ($M \neq M'$) имеем $h(M) = h(M')$, вычисленные с использование одной и той же хеш-функции (или алгоритма хеширования). *Коллизией 2-го рода* считаем ситуацию, при которой для двух произвольных сообщений M и M' ($M \neq M'$) имеем $h(M) = h(M')$, вычисленные с использование одной и той же хеш-функции (или алгоритма хеширования).

Коллизии первого рода: подобрать к уже известному сообщению и хеш-значению от него другое сообщение с тем же хеш-значением

$$m_1, H(m_1) \rightarrow m_2, H(m_2) = H(m_1)$$

Коллизии второго рода: подобрать к известному хеш-значению сообщение, которое даст такое же значение хеша

$$H(m) \rightarrow g, H(g) = H(m)$$

Рассматриваемая схема ЭЦП состоит из трех основных шагов:

- генерация ключа;
- формирование подписи; для заданного электронного документа M (с помощью закрытого ключа) вычисляется подпись;
- проверка (верификация) подписи; для данных документа и подписи с помощью открытого ключа определяется действительность подписи.

15 Общая характеристика алгоритмов хеширования классов MD и SHA.

1.

Схема MD5

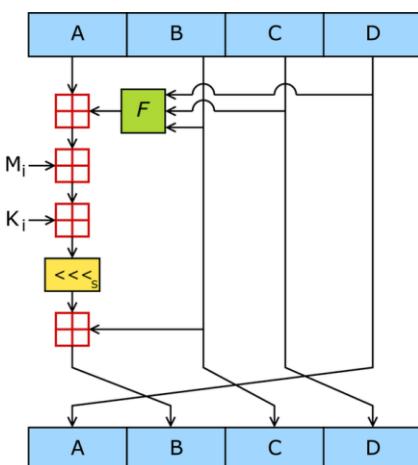
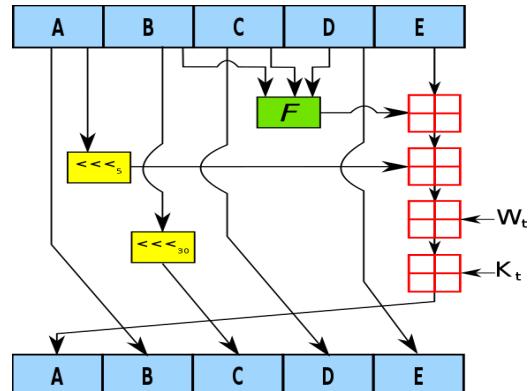


Схема SHA1



Сходства SHA1 и MD5:

1. Четыре этапа.
2. Каждое действие прибавляется к ранее полученному результату.
3. Размер блока обработки равный 512 бит.
4. Оба алгоритма выполняют сложение по модулю 2^{32} : рассчитаны на 32-битную архитектуру.

Различия SHA1 и MD5 (основные):

1. В MD5 длина дайджеста составляет **128 б**, в SHA1 — **160 б**.
2. В MD5 четыре различных элементарных логических функции, в SHA1 — три.
3. SHA1 содержит больше раундов (80 вместо 64) и выполняется на 160-битном буфере по сравнению со 128-битным буфером MD5. SHA-1 приблизительно на 25 % медленнее, чем MD5
4. В SHA1 добавлена пятая переменная.

5. SHA1 использует циклический код исправления ошибок.

16 Алгоритмы хеширования класса MD. Области использования.

Алгоритм условно можно разделить на 5 стадий:

- расширение входного сообщения; (Сообщение расширяется таким образом, чтобы его длина (в битах) была конгруэнтна 448 по модулю 512, т. е. сообщение расширяется до размера так, что ему недостает всего 64 бита, чтобы иметь длину, кратную 512)
- разбивка расширенного сообщения на блоки; (На этой стадии каждый 512-битный блок разделяется на 16 32-разрядных слов)
- инициализация начальных констант (константы A, B, C, D или MD-буфер);
- обработка сообщения поблочно (основная процедура алгоритма хеширования);
- вывод результата.(конкатенация четырех 32-разрядных слов, начиная с младшего байта регистра A и заканчивая старшим байтом регистра D)

Применяется для:

Поиск дублирующихся файлов на компьютере или в интернете (сравнивая MD5 файлов)

Пример. Графическая программа dupliFinder под Windows и [Linux](#)

Проверка целостности скачанных файлов — некоторые программы идут вместе со значением хеша. Пример. Диски для инсталляции.

Хеширование паролей. Пример. ("md5") = 1bc29b36f623ba82aaf6724fd3b16718 ("") = d41d8cd98f00b204e9800998ecf8427e (нулевая строка)

М = ИЯТУНОВИЧ
ТАТЬЯНА ВАЛЕРЬЕВНА

Чижукових Т.Н.
Ч. 8-2, ФИТ
3 курс

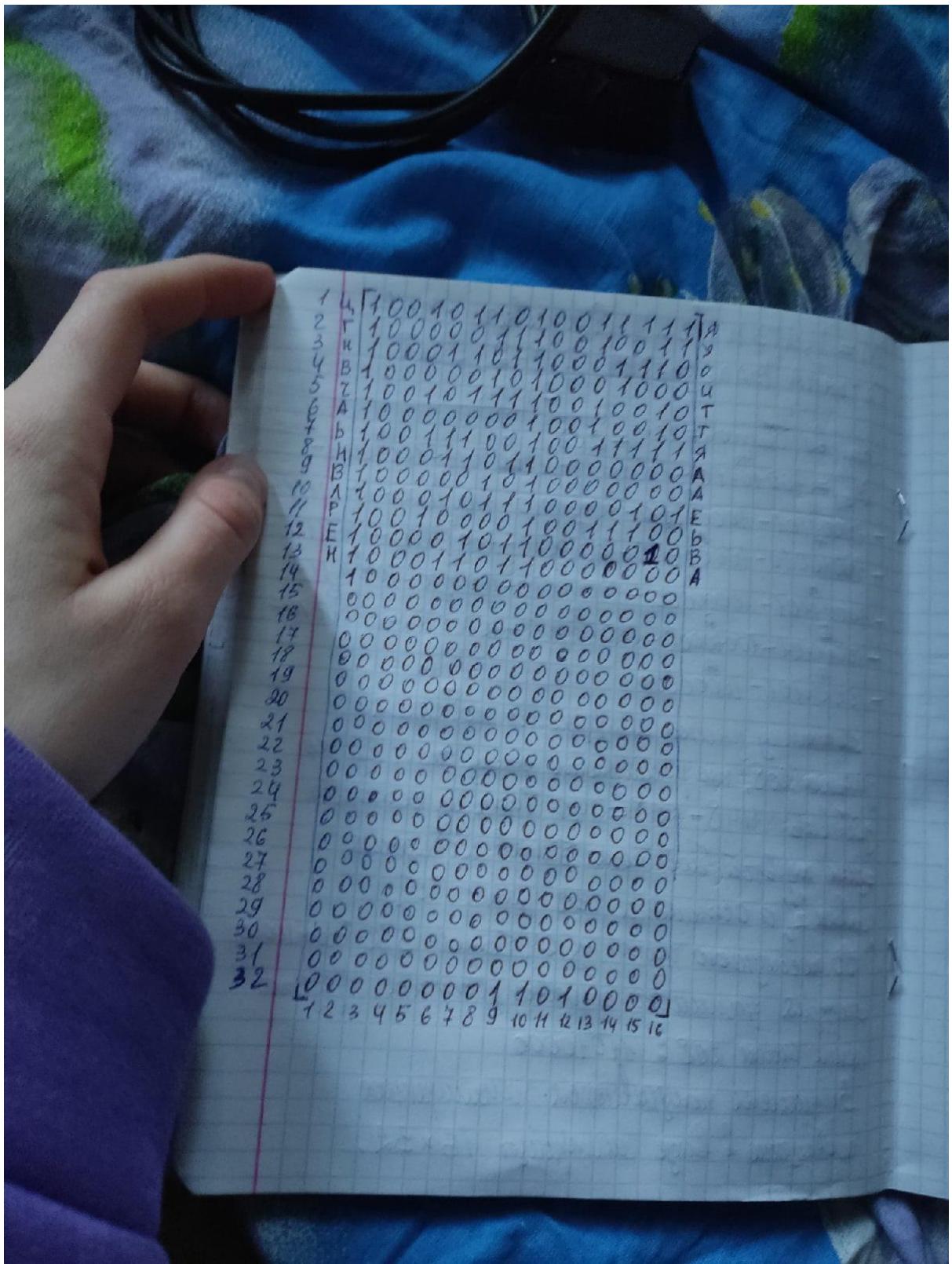
$$\text{Кол-во битов} = 26 \cdot 8 = 208$$

$$\begin{aligned} M_{\text{одн}} &= 10010110 - 10011111 - 10000011 - 10010011 - \\ &- 10001101 - 10001110 - 10000010 - 10001000 - 10010111 - \\ &- 10010010 - 10000000 - 10010010 - 10011100 - 10011111 - \\ &- 10001101 - 10000000 - 10000010 - 10000000 - 10001011 - \\ &- 10000101 - 10010000 - 10011100 - 10000101 - 10000010 - \\ &- 10001101 - 10000000 \end{aligned}$$

Поскольку данное изображение сообщения
имеет 208, то расширить данную единицу памяти
достаточно $448 - 208 = 240$ бит, из которых
один первого бит несет битов сообщения,
оценки "1", а оставшиеся 239 рабочих "0".

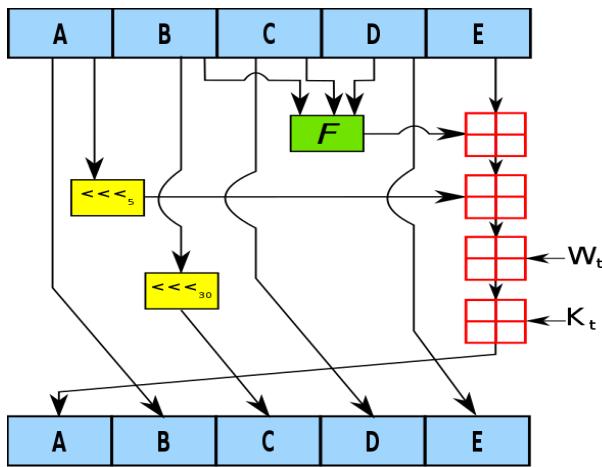
В последнюю часть из 64 битов 512-битного
изображения записано информационное представле-
ние числа 208: 11010000

Записанное получившееся расширенное
сообщение в базе шестнадцатирица 16×32 .



17 Алгоритмы хеширования класса SHA. Области использования.

1.



Обратимся , к алгоритму SHA-1. Он выдает 160-битное число. Для этого используются и инициируются изначально пять 32-разрядных констант:

A: 67 45 23 01, **B:** fef c dab 89, **C:** 98 ba dc fe, **D:** 10 32 54 76, **E:** c3 d2 e1 f0.

Вначале эти константы копируются в соответствующие переменные: A – в a, B – в b и т. д.

Главный цикл состоит из четырех раундов, каждый из которых включает по 20 операций. Каждая такая операция предусматривает вычисление нелинейной функции над тремя переменными из набора a, b, c, d, e. После этого производятся операции сдвига и сложения, аналогичные рассмотренным выше, с использованием константы t. В рассматриваемом алгоритме применяется следующий набор функций:

$F(x, y, z) = (x \text{ AND } y) \text{ OR } (\text{NOT } x \text{ AND } z)$ для t от 0 до 19,

$G(x, y, z) = x \text{ XOR } y \text{ XOR } z$ для t от 20 до 39,

$H(x, y, z) = (x \text{ AND } y) \text{ OR } (x \text{ AND } z) \text{ OR } (y \text{ AND } z)$ для t от 40 до 59,
[SEP]

$I(x, y, z) = x \text{ XOR } y \text{ XOR } z$ для t от 60 до 79.

Применяется для:

1. ЭЦП
2. Системы управления версиями (*Version Control System, VCS* — ПО для облегчения работы с изменяющейся информацией. СУВ позволяет хранить несколько версий одного и того же документа (коды программ), возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение.
3. Для построения кодов аутентификации (процедура проверки подлинности: путем сравнения введенного пароля с паролем в БД пользователей);

18 Общая классификация криптографических методов защиты информации.

1) По генерации ключа:

симметричные рисунок1(с открытым ключом - криптосистемы, использующие одинаковое значение ключа для зашифрования и расшифрования ($K_1 = K_2 = K$)) – это алгоритм DES, 3DES, Lucifer, IDEA, Blowfish, ГОСТ 28147-89;

асимметричные рисунок2 (с открытым и закрытым ключом, поскольку значение ключа должно быть известно только отправителю и получателю сообщений) – это RSA, Эль-Гамаль, Диффи-Хеллман, ранцевый алгоритм.

Ключ – секретный параметр, управляющий ходом преобразования. Ключ определяет конкретный вариант преобразования. Ключ используется в обеих операциях: как зашифрования, так и расшифрования. Таким образом, теперь функции зашифрования и расшифрования принимают следующий вид:

функция зашифрования E :

$$E_K(M) = C, \quad (3.5)$$

функция расшифрования D :

$$D_K(C) = M \quad (3.6)$$

функция зашифрования E :

$$E_{K_1}(M) = C, \quad (3.7)$$

функция расшифрования D :

$$D_{K_2}(C) = M \quad (3.8)$$

или

$$D_{K_2}(E_{K_1}(M)) = M. \quad (3.9)$$

Ключи K_1 и K_2 являются разными, но взаимозависимыми (один из них тайной не является). Поэтому асимметричные криптосистемы называют также криптосистемами с открытым или публичным ключом.

2) По способу обработки информации:

блочные (обрабатывают группы (блоки) битов открытого текста) – это Lucifer, Blowfish, IDEA, ГОСТ 28147-89;

потоковые (обрабатывается поток символов по одному - обрабатывают открытый текст побитово) – это шифр Вернама.

3) По алгоритму шифрования:

подстановочные (на место реальных букв подставляются другие буквы из алфавита) – **полиалфавитные**(на основе 2 и более алфавитов) и **моноалфавитные** (один алфавит) – это шифр Цезаря обычный, с ключевым словом, аффинная подстановка, шифр Виженера;

перестановочные (когда буквы из сообщения переставляются местами друг с другом и не заменяются другими буквами) – это вертикальный перестановочный шифр, по уровням считывания.

19 Подстановочные шифры. Шифр Цезаря.

Сущность подстановочного шифрования состоит в том, что, как правило, исходный текст (M) или зашифрованный текст (C) используют один и тот же алфавит, а тайной является алгоритм подстановки. Такой шифр называется простым или моноалфавитным.

Примером такого шифра является известный шифр Цезаря, в котором каждый символ открытого текста заменяется символом, находящимся тремя символами правее: $k = 3$ (по модулю 26 или по принципу кольца): «A» меняется на «D», «B» – на «E», «W» – на «Z», «X» – на «A» и т. д. (в некоторых случаях во внимание принимается 27-й символ – пробел). Для расшифрования необходимо выполнить обратную замену. Как видим, такая криптосистема строится на основе некоторой таблицы подстановок.

Если сопоставить каждому символу алфавита его порядковый номер (индекс), начиная с 0, то зашифрование и расшифрование можно выразить соотношениями:

$$y = (x + k) \bmod N, \quad (3.3)$$

$$x = (y - k) \bmod N, \quad (3.4)$$

где x и y – соответственно порядковые номера (индексы) символов открытого и зашифрованного текстов; k – ключ; N – мощность алфавита (количество символов).

Пример 3.1. Имеем открытый текст $M = \text{«cba»}$. На основе шифра Цезаря $C = \text{«fed»}$. Здесь $k = 3$, $N = 26$. Первый символ открытого текста (c) имеет индекс 2 (помним, что начальный символ алфавита (a) имеет нулевой индекс). Значит, первый символ шифротекста (c) будет иметь индекс $2 + k = 5$. А такой индекс в алфавите принадлежит символу f, и т. д.

Известное послание Цезаря VENI VIDI VICI (в переводе на русский означает «Пришел, увидел, победил»), направленное его другу Аминтию после победы над понтийским царем Фарнаком, выглядело бы в зашифрованном виде так:

YHQL YLGL YLFL.

Простой подстановочный шифр используется, например, в системе UNIX: простая программа шифрования (ROT13) использует смещение на 13 позиций, т. е. символ «A» заменяется на «N», и т. д.

Анализируемые шифры взламываются без труда, поскольку не скрывают частоту (вероятность) использования различных символов в открытом (а соответственно – и в зашифрованном) тексте.

Как видим, в шифре Цезаря использовались только аддитивные свойства множества целых чисел. Однако концепция, заложенная в систему шифрования Цезаря, оказалась весьма плодотворной, о чем свидетельствуют ее многочисленные модификации.

20 Особенности реализации шифровальной машины Энigma.

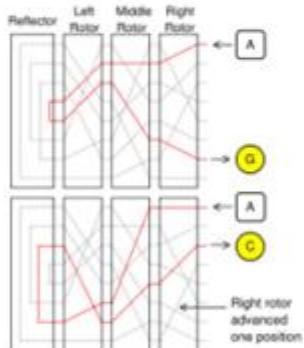
1.

Энгма – шифровальная машина, которую в XX веке использовали спецслужбы многих стран мира. Наибольшую известность получила германская версия машины и её применение во время Второй Мировой Войны.

Работа по шифрованию в энгме выполняется роторами и рефлектором.

Ротор – это диск, имеющий с двух сторон контакты (каждый контакт соответствует символу алфавита), контакты с разных сторон попарно соединены в случайном порядке. Таким образом, 1 диск осуществляет простойmonoалфавитный шифр подстановки.

Роторы расположены так, что выходные контакты одного ротора задевают входные контакты соседнего, и так до последнего диска, называемого рефлектором. Рефлектор имеет контакты только с одной стороны, и эти контакты также попарно соединены в случайном порядке.



Собранная таким образом машина выполняет над шифруемым текстом несколько monoалфавитных шифров подстановки подряд, но такое действие эквивалентно всего одному шифру подстановки. Чтобы шифр не был таким простым, после зашифровывания каждой буквы роторы по некоторому правилу поворачиваются. После результата поворота контакты повернувшегося ротора начинают касаться уже других выводов у соседнего, т.е. буква будет шифроваться уже по другому пути (см. рис). Тем самым, после шифрования каждой буквы алфавит подстановки изменяется.

Шифрующее действие Энгмы показано для двух последовательно нажатых клавиш — ток течет через роторы, «отражается» от рефлектора, затем снова через роторы. Замечание: Серыми линиями показаны другие возможные электрические цепи внутри каждого ротора. Буква А шифруется по-разному при последовательных нажатиях одной клавиши, сначала в G, затем в С. Сигнал пошёл по другому маршруту за счёт поворота ротора.

Обычно роторы поворачиваются по следующим правилам:

- последний диск поворачивается после каждой буквы,
- предпоследний диск поворачивается, когда последний диск совершил полный оборот,
- предпредпоследний диск поворачивается, когда предпоследний диск совершил полный оборот, и т.д.,
- Кроме того, для дополнительного усложнения шифра, диски поворачиваются также в некоторых особых ситуациях (таким образом, смена алфавитов происходит не так ‘монотонно’).

21 Шифр на основе аффинной системы подстановок Цезаря.

Аффинный шифр — это частный случай более общего моноалфавитного шифра подстановки. Поскольку аффинный шифр легко дешифровать, он обладает слабыми криптографическими свойствами.

Применяя одновременно операции сложения и умножения по модулю n над элементами множества (индексами букв алфавита), можно получить систему подстановок, которую называют аффинной системой подстановок Цезаря. Определим преобразование в такой системе:

$$E(x) = (ax + b) \bmod n \text{ - зашифровка,}$$

$$D(x) = a^{-1}(x - b) \bmod n \text{ – расшифровка, } a^{-1} = a^*a^{-1} \bmod n$$

где a и b – целые числа. При этом взаимно однозначные соответствия между открытым текстом и шифртекстом будут иметь место только при выполнении следующих условий: $0 \leq a < n$, $\gcd(a, n) = 1$, наибольший общий делитель (НОД) чисел a , b равен 1, т. е. эти числа являются взаимно простыми.

Пример: $a = 3$, $b = 4$, $n = 26$

Первый шаг шифрования — запись чисел, соответствующих каждой букве сообщения.

сообщение	A	T	T	A	C	K	A	T	D	A	W	N
x	0	19	19	0	2	10	0	19	3	0	22	13

Теперь, для каждого значения x найдем значение $3x+4$. Возьмем остаток от деления $(3x+4) \bmod 26$.

сообщение	A	T	T	A	C	K	A	T	D	A	W	N
x	0	19	19	0	2	10	0	19	3	0	22	13
$3x + 4$	4	61	61	4	10	34	4	61	13	4	70	43
$(3x + 4) \pmod{26}$	4	9	9	4	10	8	4	9	13	4	18	17

Последний шаг процесса шифрования заключается в подстановке вместо каждого числа соответствующей ему буквы.

сообщение	A	T	T	A	C	K	A	T	D	A	W	N
x	0	19	19	0	2	10	0	19	3	0	22	13
$3x + 4$	4	61	61	4	10	34	4	61	13	4	70	43
$(3x + 4) \pmod{26}$	4	9	9	4	10	8	4	9	13	4	18	17
шифротекст	E	J	J	E	K	I	E	J	N	E	S	R

Расшифровка: $a^{-1} = 9$ (потому что $1 = 3*9 \pmod{26}$), $b = 4$, $m = 26$

Для начала запишем численные значения для каждой буквы шифротекста. Теперь для каждого Y необходимо рассчитать $9(y + 26 - 4)$ и взять остаток от деления этого числа на 26.

шифротекст	E	J	J	E	K	I	E	J	N	E	S	R
y	4	9	9	4	10	8	4	9	13	4	18	17
$9(y + 26 - 4)$	234	279	279	234	288	270	234	279	315	234	360	351
$9(y + 26 - 4) \pmod{26}$	0	19	19	0	2	10	0	19	3	0	22	13

Поставить в соответствие числам буквы.

шифротекст	E	J	J	E	K	I	E	J	N	E	S	R
y	4	9	9	4	10	8	4	9	13	4	18	17
$9(y + 26 - 4)$	234	279	279	234	288	270	234	279	315	234	360	351
$9(y + 26 - 4) \pmod{26}$	0	19	19	0	2	10	0	19	3	0	22	13
сообщение	A	T	T	A	C	K	A	T	D	A	W	N

22 Система шифрования Цезаря с ключевым словом.

В качестве ключевого слова необходимо выбирать слово или короткую фразу (не более длины алфавита). Все буквы ключевого слова должны быть различными.

Для создания таблицы замены ключевое слово записываем под буквами алфавита, начиная с буквы, числовой код которой совпадает с выбранным числовым ключом K. Оставшиеся буквы алфавита замены записываем в алфавитном порядке (избегая повтора букв) после ключевого слова. При достижении конца таблицы циклически переходим на ее начало и дописываем последние буквы алфавита не встречавшиеся ранее.

код	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
исх. текст	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
шифртекст	Э	Ю	Я	Ш	И	Ф	Р	О	В	К	А	Б	Г	Д	Е	Ж
код	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
исх. текст	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Ь	Э	Ю	Я
шифртекст	З	Й	Л	М	Н	П	С	Т	У	Х	Ц	Ч	Щ	Ь	Ы	Ь

23 Шифр Виженера.

В шифре Виженера, мы имеем дело с последовательностью сдвигов, циклически повторяющейся. Основная идея заключается в следующем. Создается таблица (таблица Виженера) размером $N \times N$ (N – число знаков в используемом алфавите). Эти знаки могут включать не только буквы, но и, например, пробел или иные знаки. В первой строке таблицы записывается весь используемый алфавит. Каждая последующая строка получается из предыдущей циклическим сдвигом последней на 1 символ влево. Таким образом, при мощности алфавита (английского языка), равной 26, необходимо выполнить последовательно 25 сдвигов для формирования всей таблицы.

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

Рис. 3.3. Таблица (квадрат) Виженера

Шифрование происходит на основе таблицы. Выбор символа подстановки при зашифровании каждого символа сообщения (выбираются из крайнего левого столбца таблицы) происходит определением пересечения этого символа и соответствующего ему (с тем же индексом) символа ключевой последовательности (выбирается из верхней строки таблицы). Например, если первым символом сообщения будет символ В, а первым символом ключа будет символ Т, то первым символом шифртекста будет символ И (находится на пересечении 2-й строки (В) и 20-го столбца (Т) таблицы): здесь $k = 18$; при зашифровании символа Е выбирается 5 строка, и если вторым символом ключа будет символ I (9-й столбец матрицы), то вторым символом шифртекста будет символ М: здесь уже $k = 4$. Расшифрование производится так же по этой таблице, путём сопоставления символа ключа соответствующему символу зашифрованного текста.

Пример.(основываясь на таблице 3.3)

Сообщение	B	E	L	S	T	U
Ключ	T	I	R	T	I	R
Шифртекст	U	M	C	L	B	L

24 Перестановочные шифры.

Перестановочные шифры используют перестановку символов исходного сообщения в соответствии с установленным правилом.

Открытый текст остается неизменным, но символы в нем «перетасовываются» (подвергаются *пермутации*). Так, в *простом вертикальном перестановочном шифре* открытый текст пишется по горизонтали на разграфленном листе бумаги фиксированной длины, а шифртекст считывается по вертикали. Рассмотрим это на примере.

Пример 3.5. $M = \text{«ВАСЯ ЛЮБИТ МАШУ»}$. Запишем этот текст как показано на рис. 3.4.

В	А	С	Я	-
Л	Ю	Б	И	Т
-	М	А	Ш	У

Рис. 3.4. Использование простого вертикального перестановочного шифра

Считывание по столбцам снизу вверх приводит к такому шифртексту: $C = \text{«_ЛВМЮААБСШИЯУТ_»}$.

Принцип записи исходного сообщения и порядок считывания символов может быть различным. Обратимся к следующему примеру.

Пример 3.6. Открытый текст возьмем из предыдущего примера, а запишем его так, как показано на рис. 3.5.

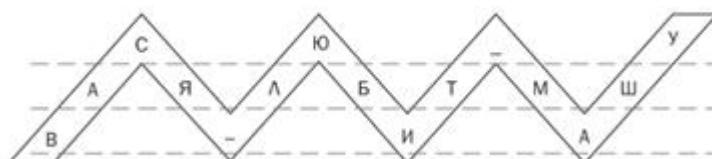


Рис. 3.5. Использование перестановочного шифра

Осуществляя считывания по уровням, начиная с верхнего, получим следующий шифртекст: $C = \text{«СЮ_УАЯЛБТМШВ_ИА»}$.

Существуют еще более сложные перестановочные шифры, но компьютеры достаточно быстро справляются с ними. При этом использование данных шифров требует большого объема памяти.

Если защита, обеспечиваемая алгоритмом, основана на сохранении в тайне самого алгоритма, то это ограниченный алгоритм. Ограниченные алгоритмы представляют некоторый исторический интерес, но не соответствуют современным стандартам.

Ограниченные алгоритмы не допускают эффективного контроля или стандартизации. Каждая группа пользователей должна использовать собственный уникальный алгоритм. Такие группы не могут использовать открытые аппаратные или программные продукты – злоумышленник может приобрести такой же продукт и раскрыть алгоритм. Этим группам приходится разрабатывать и реализовывать собственные алгоритмы.

Несмотря на указанные фундаментальные недостатки, ограниченные алгоритмы необычайно популярны в приложениях с низким уровнем защиты. Пользователи либо не осознают проблем, связанных с безопасностью своих систем, либо слабо заботятся о решении проблемы.

25 Методы симметричного криптопреобразования. Стандарт DES. Общая характеристика.

Симметричной крипtosистемой называется крипосистема, в которой для шифрования, и для дешифрования используется один и тот же ключ. Функции зашифрования и расшифрования принимают следующий вид:

функция зашифрования

$$E: E_K(M) = C,$$

функция расшифрования

$$D: D_K(C) = M$$

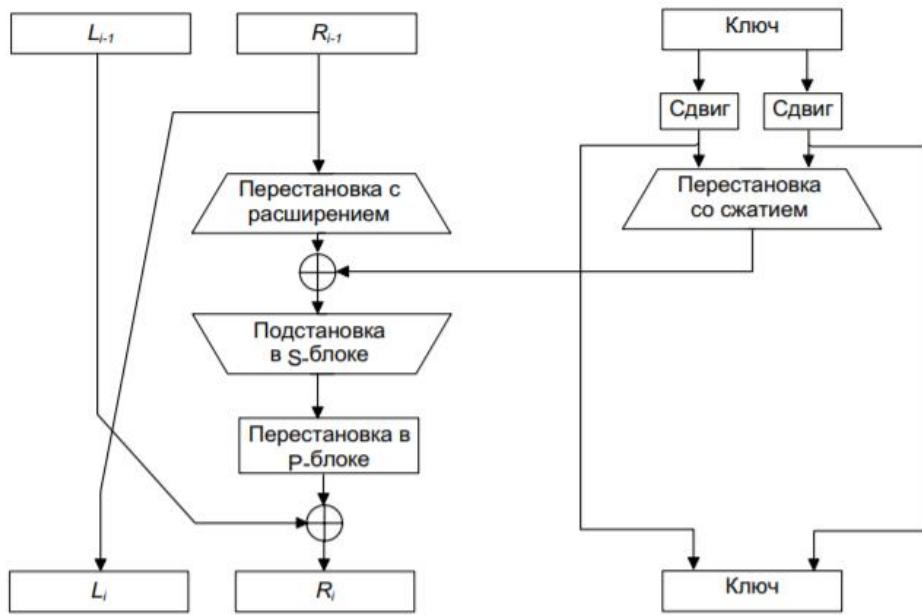
Симметричные алгоритмы подразделяются на два подкласса. Одни алгоритмы обрабатывают открытый текст побитово (иногда побайтово). Такие алгоритмы называют **потоковыми**. Другие алгоритмы обрабатывают группы (блоки) битов открытого текста. Эти алгоритмы называют **блочными**. Примеры симметричных алгоритмов: DES, ГОСТ 28147-89, Lucifer, Blowfish.

DES является блочным алгоритмом и оперирует с блоками данных размером 64 бита. При этом используется ключ длиной 56 битов;

DES представляет собой блочный шифр, он шифрует данные 64-битовыми блоками. С одного конца алгоритма вводится 64-битовый блок открытого текста, а с другого конца выходит 64-битовый блок шифротекста. DES является симметричным алгоритмом: для шифрования и дешифрования используются одинаковые алгоритм и ключ (за исключением небольших различий в использовании ключа).

Длина ключа равна 56 битам. (Ключ обычно представляется 64-битовым числом, но каждый восьмой бит используется для проверки четности и игнорируется. Биты четности являются наименьшими значащими битами байтов ключа.) Ключ, который может быть любым 56-битовым числом, можно изменить в любой момент времени. Ряд чисел считаются слабыми ключами, но их можно легко избежать. Безопасность полностью определяется ключом.

На простейшем уровне алгоритм не представляет ничего большего, чем комбинация двух основных методов шифрования: смещения и диффузии. Фундаментальным строительным блоком DES является применение к тексту единичной комбинации этих методов (подстановка, а за ней - перестановка), зависящей от ключа. Такой блок называется этапом. DES состоит из 16 этапов, одинаковая комбинация методов применяется к открытому тексту 16 раз (см. 11-й).



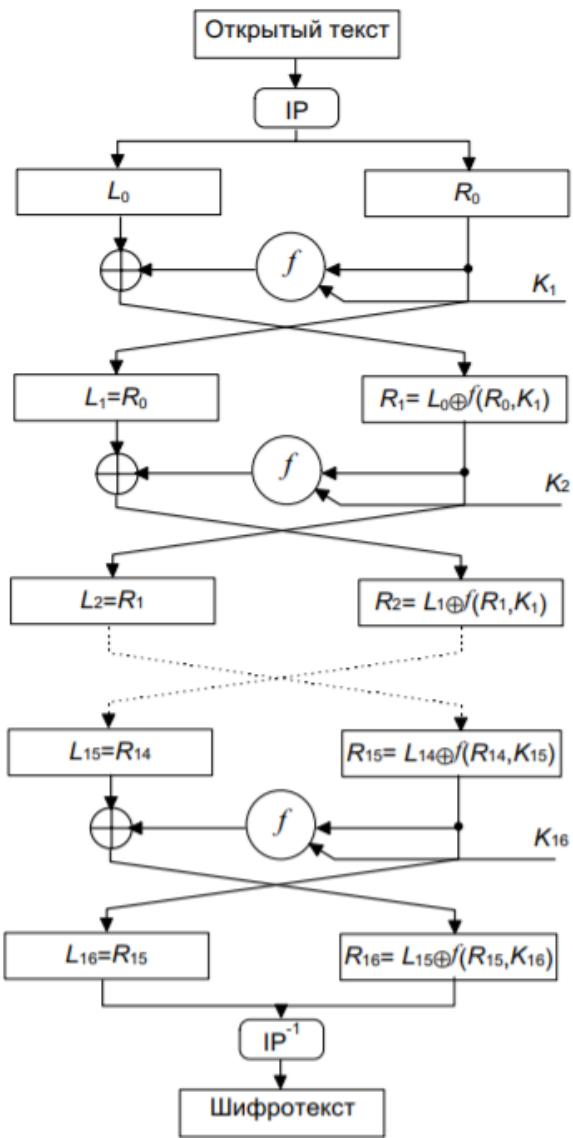


Рис. 12-1. DES.

и применить окончательную перестановку IP^{-1} как определено в следующей таблице:

IP^{-1}
40 8 48 16 56 24 64 32
39 7 47 15 55 23 63 31
38 6 46 14 54 22 62 30
37 5 45 13 53 21 61 29
36 4 44 12 52 20 60 28
35 3 43 11 51 19 59 27
34 2 42 10 50 18 58 26
33 1 41 9 49 17 57 25

То есть выходной сигнал алгоритма имеет бит 40 блока предварительного вывода в качестве его первого бита, бит 8 в качестве его второго бита и т. д., пока бит 25 блока предварительного вывода не станет последним битом вывода.

Пример: если мы обработаем все 16 блоков, используя метод, определенный ранее, мы получим, в 16-м раунде,

$$\begin{aligned}L_{16} &= 0100\ 0011\ 0100\ 0011\ 0010\ 0011\ 0100 \\R_{16} &= 0000\ 1010\ 0100\ 1100\ 1101\ 1001\ 1010\end{aligned}$$

Мы меняем порядок этих двух блоков и применяем окончательную перестановку к

$$R_{16}\ L_{16} = 00001010\ 01001100\ 11011001\ 10010101\ 01000011\ 01000010\ 00110010\ 00110100$$

$$IP^{-1} = 10000101\ 11101000\ 00010011\ 01010100\ 00001111\ 00001010\ 10110100\ 00000101$$

который в шестнадцатеричном формате

85E813540F0AB405.

Это зашифрованная форма $M = 0123456789ABCDEF$, а именно: $C = 85E813540F0AB405$.

Расшифровка - это просто обратная сторона шифрования, выполняющая те же шаги, что и выше, но изменяющая порядок, в котором применяются подключи.

26 Методы симметричного криптопреобразования. Стандарт DES. Структура одного цикла. Криптостойкость алгоритма.

Симметричной крипtosистемой называется крипосистема, в которой для шифрования, и для дешифрования используется один и тот же ключ. Функции зашифрования и расшифрования принимают следующий вид:

функция зашифрования

$$E: E_k(M) = C,$$

функция расшифрования

$$D: D_k(C) = M$$

Симметричные алгоритмы подразделяются на два подкласса. Одни алгоритмы обрабатывают открытый текст побитово (иногда побайтово). Такие алгоритмы называют **потоковыми**. Другие алгоритмы обрабатывают группы (блоки) битов открытого текста. Эти алгоритмы называют **блочными**. Примеры симметричных алгоритмов: DES, ГОСТ 28147-89, Lucifer, Blowfish.

DES является блочным алгоритмом и оперирует с блоками данных размером 64 бита. При этом используется ключ длиной 56 битов;

Входной блок данных, состоящий из 64 битов, преобразуется в выходной блок идентичной длины. В алгоритме широко используются рассеивания (подстановки) и перестановки битов текста. Комбинация двух указанных методов преобразования образует фундаментальный строительный блок DES, называемый **раундом** или **циклом**. Один блок данных подвергается преобразованию (и при зашифровании, и при расшифровании) в течение 16

раундов.

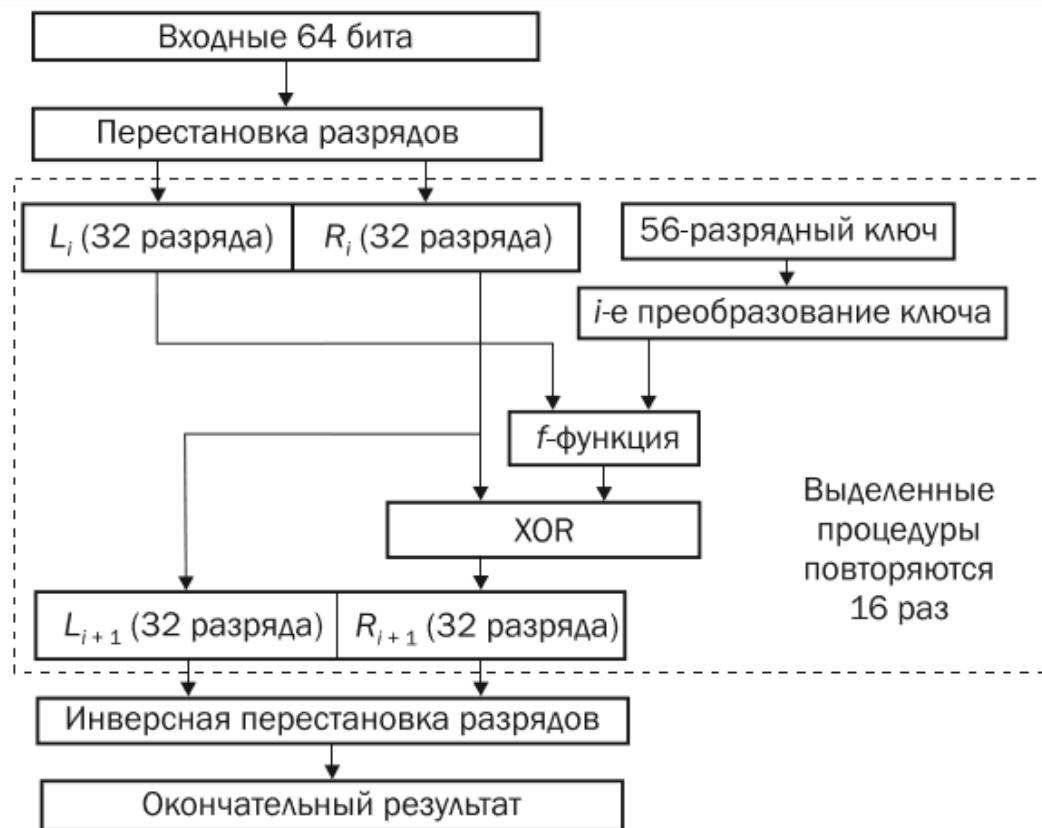


Рис. 5.1. Общая схема алгоритма DES

Перестановка разрядов означает, например, что первый бит входного блока сообщения будет размещен на 40-й позиции, а 58-й – на 1-й и т. д.

Поскольку программная реализация рассмотренной перестановки достаточно трудна, и с учетом того что сама процедура не влияет на общую криптостойкость алгоритма, во многих программных приложениях эта перестановка, как и конечная, из алгоритма исключается.

После первоначальной перестановки и разделения 64-битного блока данных на правую (R) и левую (L) половины длиной по 32 бита выполняются 16 раундов одинаковых действий.

Вводится функция f (на рис. 5.1 – f -функция), которая оперирует с 32-разрядными словами исходного текста и использует в качестве параметра 48-разрядный ключ (изначально ключ состоит из 56 бит, в каждом из фиксированного множества преобразований биты ключа сдвигаются и затем из 56 битов ключа выбираются 48).



Рис. 5.2. Схема реализации функции f

32 входных разряда расширяются до 48 (расширяющая перестановка, при этом некоторые разряды повторяются), после чего происходит сложение их по модулю два (XOR) с ключом. Расширяющая перестановка для 32 входных разряда:

Результирующий 48-разрядный код преобразуется далее в 32-разрядный с помощью S-матриц. На выходе S-матриц осуществляется перестановка символов согласно рекомендуемой схеме перестановок. Всего используется 8 S-матриц (или S-блоков). Каждый такой блок имеет 6 входов и 4 выхода, т. е. на его выходе формируется 4-битное слово, значение которого определяется входным 6-битным словом.

Расшифрование в DES. Осуществляется в обратной последовательности. Компоненты алгоритма подобраны так, чтобы для зашифрования и расшифрования применялся одинаковый алгоритм. Понятно, что ключи используются в обратном порядке в сравнении с зашифрованием.

Криптостойкость DES. Этот важнейший параметр алгоритма определяется не только ключом, но и преобразованием блока открытого текста или шифртекста с помощью S-матриц. Что касается ключа и его преобразований, то рассматриваемому алгоритму свойственна так называемая проблема слабых ключей. Вспомним, что 56-битный ключ делится пополам. Если все биты каждой половины равны 0 или 1, то во всех раундах будет использоваться одинаковый ключ. Кроме того, существуют пары различных ключей, которые при зашифровании превращают открытые тексты в одинаковый шифртекст.

27 Методы симметричного криптопреобразования. Стандарты 3DES. Реализация и криптостойкость.

В симметричных криптоалгоритмах для зашифровывания и расшифровывания сообщения используется один и тот же блок информации (ключ). Хотя алгоритм воздействия на передаваемые данные может быть известен посторонним лицам, но он зависит от секретного ключа, которым должны обладать только отправитель и получатель. Симметричные криптоалгоритмы выполняют преобразование небольшого блока данных (1 бит либо 32—128 бит) в зависимости от секретного ключа таким образом, что прочесть исходное сообщение можно только зная этот секретный ключ.

Он предусматривает выполнение формально трех DES (тройной DES или 3DES). Один из вариантов такого подхода предполагает следующие операции: отправитель сначала шифрует сообщение первым ключом, зачем расшифровывает результат вторым ключом и, наконец, опять шифрует первым ключом: $C = EK1(DK2(EK1(M)))$, получатель соответственно расшифровывает сообщение первым ключом, зашифровывает вторым, расшифровывает первым: $M = DK1(EK2(DK1(C)))$.

В литературе такой режим часто называют зашифрование-расшифрование-зашифрование (Encrypt-Decrypt-Encrypt, EDE).

Еще большей криптостойкостью обладает 3DES при использовании трех различных ключей: $C = EK3(DK2(EK1(M)))$, $M = DK1(EK2(DK3(C)))$.

Криптостойкость такого преобразования значительно возрастает.

1. 3DES с различными ключами имеет длину ключа равную 168 бит, но из-за атак «встреча посередине» (известны ***M*** и ***C***, нужно найти ***K***) эффективная криптостойкость составляет только 112 бит;
2. в варианте DES-EDE, в котором ***K₁*=K₃**, эффективный ключ имеет длину 80 бит;
3. Для успешной атаки на 3DES потребуется около 2^{32} бит известного открытого текста 2^{113} шагов, 2^{90} циклов DES-шифрования и 2^{88} [бит](#) памяти.

28 Шифровальная машина Энигма. Устройство, функционирование, криптостойкость.

Машина «Энигма» – это электромеханическое устройство. Как и другие роторные машины, «Энигма» состоит из комбинации механических и электрических подсистем. Механическая часть включает в себя клавиатуру, набор вращающихся дисков – роторов, которые расположены вдоль вала и прилегают к нему, и ступенчатого механизма, движущего один или несколько роторов при каждом нажатии на клавишу

Электрическая часть, в свою очередь, состоит из электрической схемы, соединяющей между собой клавиатуру, коммутационную панель, лампочки и роторы (для соединения роторов использовались скользящие контакты).

На рис. 4.1 показана фотография одной из моделей «Энигмы» с указанием месторасположения основных модулей машины. Как видно на этом рисунке, «Энигма» состоит из 5 основных блоков:

- панели механических клавиш 1 (дают сигнал поворота роторных дисков);
- трех (или более) роторных дисков 2, каждый имеет контакты по сторонам, по 26 на каждую, которые коммутируют в случайном порядке; по окружности нанесены буквы латинского алфавита либо числа;
- рефлектора 3 (имеет контакты с крайним слева ротором);
- коммутационной панели 4 (служит для того, чтобы дополнительно менять местами электрические соединения (контакты) двух букв);
- панели в виде электрических лампочек 5; индикационная панель с лампочками служит индикатором выходной буквы в процессе шифрования.



Рис. 4.1. Одна из моделей (трехроторная) «Энигмы» [19]:
1 – панель механических клавиш; 2 – роторные диски; 3 – рефлектор;
4 – коммутационная панель; 5 – индикационная панель

Конкретный механизм мог быть разным, но общий принцип был таков: при каждом нажатии на клавишу самый правый ротор сдвигался на одну позицию, а при определенных условиях сдвигались и другие роторы. Движение роторов приводило к различным криптографическим преобразованиям при каждом следующем нажатии на клавишу на клавиатуре, т. е. зашифрование/расшифрование сообщений основано на выполнении ряда замен (подстановок) одного символа другим

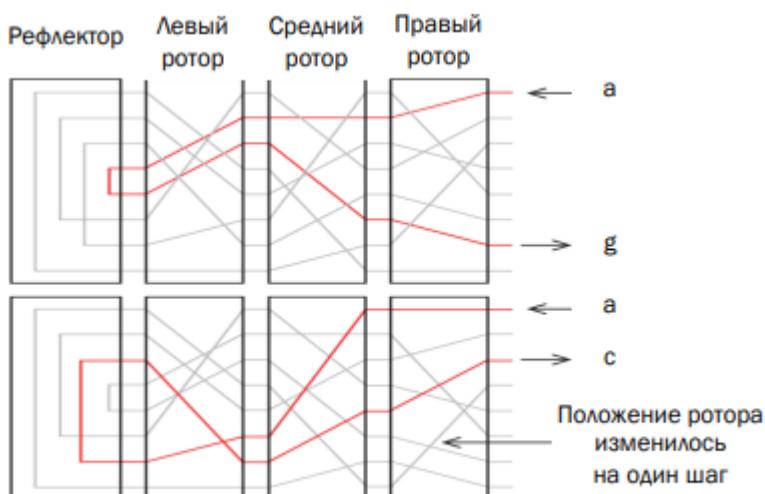


Рис. 4.3. Пояснение к принципу шифрования путем формирования электрической цепи [20]

Как мы отмечали выше, «Энигма» строится на основе подстановочных шифров, подобных шифру Цезаря, в котором, как известно, ключ сообщения, который должен знать получатель, – это просто смещение между двумя алфавитами. Принято считать, что в основе шифра «Энигмы» лежит динамический шифр Цезаря.

Криптостойкость:

Как мы неоднократно подчеркивали, преобразование «Энигмы» для каждой буквы может быть определено математически как результат подстановок. Рассмотрим трехроторную модель «Энигмы». Положим, что символом В обозначаются операции с использованием коммутационной панели, соответственно, символы Re – отражателя, а L, M и R обозначают действия левых, средних и правых роторов соответственно. Тогда процесс зашифрования символа m с использованием некоторой ключевой информации K формально можно записать в следующем виде:

$$E_K = f(m, B, Re, L, M, R).$$

Чтобы оценить криптостойкость шифра, нужно учитывать все возможные настройки машины. Для этого необходимо рассмотреть следующие свойства «Энигмы»:

- выбор и порядок роторов;
- разводку (коммутацию) роторов;
- настройку колец на каждом из роторов;
- начальное положение роторов в начале сообщения;
- отражатель;
- настройки коммутационной панели.

Чтобы выбрать 3 ротора из возможных 5, существует 60 комбинаций ($5 \cdot 4 \cdot 3$). Каждый ротор (его внутренняя проводка) может быть установлен в любом из 26 положений. Следовательно, с 3 роторами имеется 17 576 различных положений ротора ($26 \cdot 26 \cdot 26$). Кольцо на каждом роторе содержит маркировку ротора (что здесь неважно) и выемку, которая влияет на шаг перемещения расположенного левее ротора. Каждое кольцо может быть установлено в любом из 26 положений. Поскольку слева от третьего (наиболее левого) ротора нет ротора, на расчет влияют только кольца самого правого и среднего ротора. Это дает 676 комбинаций колец ($26 \cdot 26$).

Коммутационная панель обеспечивает самый большой набор возможных настроек. Для первого кабеля одна сторона может иметь любое из 26 положений, а другая сторона – любое из 25 оставшихся положений (одна буква коммутируется с другой). Однако поскольку комбинация и ее обратная сторона идентичны (AB такая же, как BA), мы должны игнорировать все двойные числа во всех возможных комбинациях для одного кабеля, предоставляем $(26 \cdot 25) / (1! \cdot 21)$, или 325 уникальных способов коммутаций одним кабелем. Для двух кабелей – $(26 \cdot 25)$ комбинаций для первого кабеля и, поскольку два разъема уже используются, то получается $(24 \cdot 23)$ комбинаций для второго кабеля. Следуя этой простой логике, получается $(26 \cdot 25 \cdot 24 \cdot 23) / (2! \cdot 22) = 44\,850$ уникальных способов коммутаций с использованием двух кабелей. Для трех кабелей – $(26 \cdot 25 \cdot 24 \cdot 23 \cdot 22 \cdot 21) / (3! \times 23) = 3\,453\,450$ комбинаций и т. д.

Таким образом, с использованием 10 кабелей на коммутационной панели получаются 150 738 274 937 250 различных комбинаций. Формула, где n равно количеству кабелей, равна $26! / (26 - 2n)! \cdot n! \cdot 2n$. Численно это дает: $60 \cdot 17\,576 \cdot 676 \cdot 150\,738 \cdot 274 \cdot 937 \cdot 250 = 107\,458\,687\,327\,250\,619\,360\,000$, или $1,07 \cdot 10^{23}$. Таким образом, практически рассматриваемая версия «Энигмы» (3 ротора с выбором из 5

роторов, отражатель В и 10 штекерных кабелей для коммутационной панели) может быть настроена на $1,07 \cdot 1023$ различных состояний, чт

29 Сравнительная характеристика алгоритмов Lucifer, IDEA, ГОСТ 28147-89, Blowfish.

Для сравнения с IDEA выбраны [DES](#), [Blowfish](#) и [ГОСТ 28147-89](#). Выбор [DES](#) обусловлен тем, что IDEA проектировался как его замена. [Blowfish](#) выбран потому, что он быстр, и был придуман известным криптологом Брюсом Шнайером. Для сравнения также выбран [ГОСТ 28147-89](#), блочный шифр, разработанный в [СССР](#). Как видно из таблицы, размер ключа у IDEA больше, чем у DES, но меньше, чем у ГОСТ 28147-89 и Blowfish. Скорость шифрования IDEA на [Intel486SX](#)/33МГц больше в 2 раза, чем у DES, выше чем у ГОСТ 28147-89, но почти в 2 раза меньше, чем у Blowfish.

Таблица параметров

Алгоритм	Размер ключа, бит	Длина блока, бит	Число раундов	Скорость шифрования на Intel486SX /33МГц (Кбайт/с)	Основные операции
DES	56	64	16	35	Подстановка, перестановка, побитовое исключающее ИЛИ
IDEA	128	64	8	70	Умножение по модулю $\{2^{16}+1\}$, сложение по модулю $\{2^{16}\}$, побитовое исключающее ИЛИ
Blowfish	32-448	64	16	135	Сложение по модулю $\{2^{32}\}$, подстановка, побитовое исключающее ИЛИ

<u>ГОСТ</u> <u>28147-89</u>	256	64	32	53	Сложение по модулю 2^{32} , подстановка, побитовое исключающее ИЛИ, циклический сдвиг
--------------------------------	-----	----	----	----	---

30 Криптографические системы с открытым (публичным) ключом. Задача об укладке ранца.

Две проблемы, связанные с практическим использованием симметричных криптосистем (хранение и обмен ключевой информацией), стали важными побудительными мотивами для разработки принципиально нового класса методов шифрования: криптографии с открытым ключом

В основу положена идея использовать ключи парами: один – для зашифрования (открытый или публичный ключ), другой – для расшифрования (тайный ключ)

- Алгоритм Диффи – Хеллмана для распределения ключей
- Ранцевый алгоритм
- RSA
- Алгоритм Эль-Гамала

Проблема укладки ранца формулируется просто. Дано множество предметов различного веса. Спрашивается, можно ли положить некоторые из этих предметов в ранец так, чтобы его вес стал равен определенному значению? Более формально задача формулируется так: дан набор значений M_1, M_2, \dots, M_n и суммарное значение S . Требуется вычислить значения b_i такие, что: $S = b_1 \cdot M_1 + b_2 \cdot M_2 + \dots + b_n \cdot M_n$. Здесь b_i может быть либо нулем, либо единицей. Значение $b_i = 1$ означает, что предмет M_i кладут в рюкзак, а $b_i = 0$ – не кладут

Предметы из «кучи» выбираются с помощью блока открытого текста, длина которого (в битах) равна количеству предметов в куче. При этом биты открытого текста соответствуют значениям b_i , а шифртекст является полученным суммарным весом

Пример: открытый текст 1 1 1 1 1 0; вещи в рюкзаке 3 4 6 7 10 11; шифртекст $3*1 + 4*1 + 6*1 + 7*1 + 10*1 + 11*0 = 13011$

Сложность шифра заключается в том, что существуют две проблемы рюкзака: «лёгкая» и «трудная». «Лёгкая» проблема может быть решена за линейное время, «трудная» нет. Открытый ключ – «трудная» проблема, так как её легко применять для шифрования и невозможно для дешифровки сообщения. Закрытый ключ – «лёгкая» проблема, так как позволяет легко дешифровать сообщение. При отсутствии закрытого ключа нужно решать «трудную» проблему рюкзака. Меркл и Хеллман, используя модульную арифметику, разработали способ трансформации «лёгкого» рюкзака в «трудный»

Lekun B. A. 3 курс 8 группы

Задача 1. Код

$$\{26, 27, 54, 108, 216, 432, 864, 1728\}$$
$$m = 3456 \quad n = 7$$

Безумная отыскание кодов

$$26 \cdot 7 \pmod{3456} = 182$$
$$27 \cdot 7 \pmod{3456} = 189$$
$$54 \cdot 7 \pmod{3456} = 378$$
$$108 \cdot 7 \pmod{3456} = 756$$
$$216 \cdot 7 \pmod{3456} = 1512$$
$$432 \cdot 7 \pmod{3456} = 3024$$
$$864 \cdot 7 \pmod{3456} = 2592$$
$$1728 \cdot 7 \pmod{3456} = 1728$$

$$\{182, 189, 378, 756, 1512, 3024, 2592, 1728\}$$

$$C = 209 = 11010001$$

$$B = 194 = 11000010$$

$$A = 192 = 11000000$$

$$C \Rightarrow 182 + 189 + 756 + 1728 = 2855$$

$$B \Rightarrow 182 + 189 + 2592 = 2963$$

$$A \Rightarrow 182 + 189 = 371$$

Кодирование: 2855, 2963, 371

$$7 \cdot h^{-1} \equiv 1 \pmod{3456}$$

$$h^{-1} = 1925$$

$$2855 \cdot 1925 \pmod{3456} = 1889 = 11010001$$

$$2963 \cdot 1925 \pmod{3456} = 317 = 11000010$$

$$371 \cdot 1925 \pmod{3456} = 53 = 11000000$$

31 Управление криптографическими ключами. Алгоритм рукопожатия.

Управление ключами – информационный процесс, включающий в себя три элемента:

- генерацию ключей;
- накопление ключей;
- распределение ключей.

Генерация ключей. В реальных системах используются специальные аппаратные и программные методы генерации случайных ключей. Как правило используют датчики случайных чисел. Однако степень случайности их генерации должна быть достаточно высокой. Идеальными генераторами являются устройства на основе “натуральных” случайных процессов (радиоактивный распад, белый шум)

Накопление ключей. Под накоплением ключей понимается организация их хранения, учета и удаления

Распределение ключей – обмен ключами между сторонами

Протокол рукопожатия

SSL клиент и сервер договариваются об установлении связи с помощью процедуры рукопожатия. Во время рукопожатия клиент и сервер договариваются о различных параметрах, которые будут использованы, чтобы обеспечить безопасность соединения:

1. Клиент посыпает серверу номер версии SSL клиента, поддерживающие алгоритмы шифрования и сжатия, специфичные данные для сеанса и другую информацию, которая нужна серверу, чтобы общаться с клиентом, используя SSL.
2. Сервер посыпает клиенту номер версии SSL сервера, алгоритм сжатия и шифрования (выбранные из посланных ранее клиентом), специфичные данные для сеанса и другую информацию, которая нужна серверу, чтобы общаться с клиентом по протоколу SSL. Сервер также посыпает свой сертификат, который требует проверки подлинности клиента. После идентификации сервер запрашивает сертификат клиента.
3. Клиент использует информацию, переданную сервером для проверки подлинности. Если сервер не может быть проверен, пользователь получает предупреждение о проблеме и о том, что шифрование и аутентификация соединения не может быть установлена. Если сервер успешно прошел проверку, то клиент переходит к следующему шагу.
4. Используя все данные, полученные до сих пор от процедуры рукопожатие, клиент (в сотрудничестве с сервером) создаёт предварительный секрет сессии, в зависимости от используемого шифра от сервера, шифрует его с помощью открытого ключа сервера (полученного из сертификата сервера, отправленного на 2-м шаге), а затем отправляет его на сервер.
5. Если сервер запросил аутентификацию клиента (необязательный шаг рукопожатия), клиент также подписывает ещё один кусок данных,

который является уникальным для этого рукопожатия и известным как для клиента, так и сервера. В этом случае, клиент отправляет все подписанные данные и собственный сертификат клиента на сервер вместе с предварительно зашифрованным секретом.

6. Сервер пытается аутентифицировать клиента. Если клиент не может пройти проверку подлинности, сеанс заканчивается. Если клиент может быть успешно аутентифицирован, сервер использует свой закрытый ключ для расшифровки предварительного секрета, а затем выполняет ряд шагов (которые клиент также выполняет), чтобы создать главный секрет.
7. И клиент, и сервер используют секрет для генерации ключей сеансов, которые являются симметричными ключами, использующиеся для шифрования и расшифрования информации, которой обмениваются во время SSL сессии. Происходит проверка целостности (то есть, для обнаружения любых изменений в данных между временем когда он был послан, и временем его получения на SSL-соединении).
8. Клиент посыпает сообщение серверу, информируя его, что будущие сообщения от клиента будут зашифрованы с помощью ключа сеанса. Затем он отправляет отдельное, зашифрованное сообщение о том, что часть рукопожатия закончена.
9. И в заключение, сервер посыпает сообщение клиенту, информируя его, что будущие сообщения от сервера будут зашифрованы с помощью ключа сеанса. Затем он отправляет отдельное, зашифрованное сообщение о том, что часть рукопожатия закончена.

На этом рукопожатие завершается, и начинается защищенное соединение, которое зашифровывается и расшифровывается с помощью ключевых данных. Если любое из перечисленных выше действий не удаётся, то рукопожатие SSL не удалось, и соединение не создается.

Протокол рукопожатия (основа протокола SSL)

А и В желают определить общий секретный ключ (K). Они знают открытые ключи друг друга.

1. А посыпает В сообщение $C = E_B(I_A, R_A)$; E_B – процедура зашифрования с открытым ключом B , I_A – идентификатор А и R_A – случайное число.
2. В расшифровывает С и получает I_A и R_A .

В посыпает $C' = E_A(I_B, R_A)$ в адрес А;

После расшифрования С' А может проверить, что В получил R_A , поскольку только В может расшифровать С.

3. А посыпает В сообщение $C'' = E_B(K_B)$,

В расшифрует С" и сможет проверить, что А получил

I_B , поскольку только А может расшифровать С'.

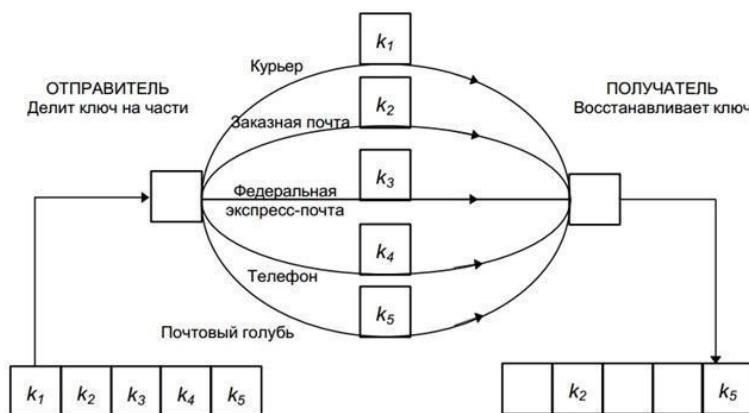
Тем самым А и В аутентифицировали друг друга.

Теперь **A** посыпает **B**: $C''' = E_B(K)$, **B** расшифровывает сообщение и получает **K**.

Алгоритм обеспечивает как секретность, так и аутентичность при обмене ключом K. Может использоваться **T_A** – временной штамп (метка) стороны **A**. Стороны А и В завершили процедуру взаимной аутентификации (пожали друг другу руки).

32 Распределение ключей на основе симметричных систем.

Одним из решений проблемы распределения ключа является процесс его разделения на части и передачи по независимым каналам. Одна часть может быть передана по телефону, другая – почтой, и т.д. Для перехвата ключа шифрования злоумышленнику потребуется перехватить сразу все каналы передачи, выделить в них на фоне других данных передаваемую часть ключа и только затем правильно скомпоновать ключ. Такую задачу достаточно непросто решить на практике.



В процессе передачи некоторые части ключа могут быть искажены или утрачены, поэтому необходимо также предусмотреть меры по повышению надежности передачи частей ключа и их целостности.

Ключи шифрования ключей, общие для пары пользователей, очень удобно использовать в небольших сетях. Однако при таком способе распределения ключей в сети из N человек потребуется $(N(N-1))/2$ обменов, что на практике резко ограничивает возможность использования такой схемы.

В случаях, когда количество участников в сети может быть достаточно велико, гораздо более эффективным будет использование **центрального сервера ключей**.

Классический алгоритм **Диффи-Хеллмана** позволяет согласовать двум участникам секретный ключ по открытому каналу, который может прослушивать злоумышленник

33 Алгоритм передачи ключа по Диффи-Хеллману.

Алгоритм Диффи – Хеллмана для распределения ключей: абоненты А и В могут воспользоваться этим алгоритмом для обмена ключевой информацией по открытым каналам. Предварительно стороны выбирают большие простые числа n и g .

Протокол обмена:

1. А выбирает случайное большое число x , вычисляет
 $X = g^x \text{ mod } n$ и результат вычисления отсылает В.
2. В выбирает случайное большое число y , вычисляет $Y = g^y \text{ mod } n$
и результат вычисления отсылает А.
3. А вычисляет $k_1 = Y^x \text{ mod } n$.
4. В вычисляет $k_2 = X^y \text{ mod } n$.

Таким образом, $k_1 = k_2 = g^{(xy)} \text{ mod } n = k$.

Это число сторонами может использоваться как совместный ключ (секретный). Для того чтобы третья сторона смогла вычислить значение k , она должна вычислить значение дискретного логарифма.

Протокол Диффи – Хеллмана является уязвимым для атаки, называемой «человек в середине»: злоумышленник С может перехватить открытое значение, посыпанное от А к В, и послать вместо него свое открытое значение. Затем он может перехватить открытое значение, посыпанное от В к А, и также передать вместо него свое открытое значение. Тем самым С получит общие секретные ключи с А и В и сможет читать и/или модифицировать сообщения, передаваемые от одной стороны к другой.

34 Алгоритм шифрования RSA. Реализация и криптостойкость.

Описание алгоритма RSA. Предварительный этап – генерация ключа, состоящего из трех чисел.

1. Случайным образом выбираются два больших простых числа p и q . Рассчитывается произведение $n = p \cdot q$. Это первое число, входящее в ключ.
2. Вычисляется функция Эйлера $\phi(n) = (p - 1)(q - 1)$.
3. Случайным образом выбирается простое число e – вторая часть ключа, которое удовлетворяет условиям: $e < \phi(n)$; e и $\phi(n)$ должны быть взаимно простыми числами.
4. Вычисляется число d – третья часть ключа, которое является обратным числу e : $e \cdot d \equiv 1 \pmod{\phi(n)}$.

Пара чисел (e, n) делается открытым ключом и помещается в общедоступный справочник (база данных), а о числах p , q можно забыть (но это также является тайной информацией). Пара (d, n) – секретный ключ (понятно, что секретным является лишь значение первого числа из этой пары); эти числа также являются взаимно простыми. Первая обычно используется для зашифрования, другая – для расшифрования. Не вдаваясь в детали, отметим важную особенность генерации и использования ключа: числа e и d можно поменять местами. Только в этом случае первое из них будет тайным (входит в ключ для расшифрования), второе – открытым.

Следующий этап – использование ключа.

Зашифрование. Если шифруется сообщение M , состоящее из r блоков: $m_1, m_2, \dots, m_i, \dots, m_r$, то шифртекст C будет состоять из такого же числа (r) блоков, представляемых числами:

$$c_i = (m_i)^e \bmod n. \quad (5.3)$$

Расшифрование. Для расшифрования каждого зашифрованного блока производится вычисление вида

$$m_i = (c_i)^d \bmod n. \quad (5.4)$$

Рассмотрим пример.

Генерация ключа. Принимаем $p = 11$, $q = 5$.

Вычисляем $n = 11 \cdot 5 = 55$.

Определяем функцию Эйлера: $\phi(55) = (11 - 1)(5 - 1) = 40$.

Выбираем ключ шифрования $e = 7$, который удовлетворяет условиям $7 < 40$; НОД $(7, 40) = 1$.

Определяем d – ключ расшифровывания – из уравнения $7 \cdot d \equiv 1 \pmod{40}$.

Рассмотрим способ нахождения d .

Для решения уравнения $7 \cdot d \equiv 1 \pmod{40}$ используем алгоритм Евклида:

$$40 = 7 \cdot 5 + 5;$$

$$7 = 5 \cdot 1 + 2;$$

$$5 = 2 \cdot 2 + 1;$$

$$2 = 1 \cdot 2 + 0.$$

Обратная подстановка дает:

$$1 = 5 - 2 \cdot 2 = 5 - (7 - 5 \cdot 1)2 = 5 \cdot 3 + 7(-2) = (40 - 7 \cdot 5)3 + 7(-2) = 40 \cdot 3 + 7(-17).$$

Поскольку $-17 \equiv 23 \pmod{40}$, то $d = 23$.

Зашифровываем сообщение $M = 15$ (сообщение состоит из одного блока), используя выражение (5.3):

$$C = 15^7 \bmod 55 = 5.$$

Для расшифрования C воспользуемся формулой (5.4):

$$M = 5^23 \bmod 55 = 15.$$

Криптостойкость алгоритма. Зависит от трудоемкости решения проблемы разложения на множители больших чисел. «Лобовой метод» вскрытия системы RSA заключается в нахождении числа d , обратного e по модулю $\phi(n)$. Это легко сделать, если известны числа p и q . Отметим, что математически не доказано, что для восстановления сообщения по шифртексту и по значению открытого ключа нужно разложить n на множители. Тем не менее именно этот подход является наиболее очевидным. Известно, что факторизованы числа длиной более 512 битов. Значит, нужно выбирать n больше этого значения. Кроме того, числа p и q не должны быть слишком близкими друг к другу.

Существует одна важная особенность. Если группа пользователей применяет одно и то же значение модулю, но каждый имеет разный ключ, и одно и то же сообщение шифровалось разными пользователями, то вероятность взлома шифра значительно возрастает.

Существует и другое направление взлома шифра. Криптоаналитик располагает шифртекстом $C = E_e(M)$. Цель – восстановить M (расшифровать C). Для этого он шифрует известным открытым ключом произвольное сообщение M' . Получает C' . Если $C = C'$, то получено M ($M = M'$), в ином случае шифруется другое сообщение M'' и т. д.

В заключение отметим, что аппаратно реализованный алгоритм RSA работает более чем в 1000 медленнее, чем алгоритм DES. При программной же реализации обоих алгоритмов быстродействие первого из них хуже примерно в 100 раз

35 Алгоритм шифрования Эль-Гамаля. Реализация и криптостойкость.

Данный алгоритм является альтернативой алгоритму RSA и, при равном значении ключа, обеспечивает ту же криптостойкость. Стойкость алгоритма Эль-Гамаля основана на трудности вычисления дискретных логарифмов.

Предварительный этап – генерация ключа, состоящего из четырех чисел:

1. Выбирается простое число p и два случайных числа, меньших чем p : числа x и g .
2. Далее вычисляется
 $y = g^x \text{ mod } p$.

Открытый ключ: y, g и p ; тайный ключ: x .

Следующий этап – использование ключа.

В отличие от алгоритма RSA, рассматриваемый алгоритм предусматривает использование дополнительного параметра, который обозначим k . Использование этого параметра снижает вероятность взлома шифра. Это число является секретным и должно быть взаимно простым с $p - 1$.

Зашифрование. Если шифруется сообщение M , состоящее из r блоков: $m_1, m_2, \dots, m_i, \dots, m_r$, то шифртекст C будет состоять из та- кого же числа (r) блоков, представляемых парой чисел:

$$a_i = g^k \text{ mod } p, (5.5)$$

$$b_i = (y_k \cdot m_i) \bmod p. \quad (5.6)$$

Расшифрование. Для расшифрования производится вычисление вида:

$$m_i = b_i / (a_i)_x \bmod p. \quad (5.7)$$

Так как $(a_i)_x$ в силу формулы (5.5) можно заменить на $g_{kx} \bmod p$ и учитывая то, что $y = g_x \bmod p$, можем получить подтверждение справедливости выражения (5.7):

$$b_i / (a_i)_x \equiv y_k \cdot m_i / (a_i)_x \bmod p \equiv g_{kx} \cdot m_i / g_{kx} \bmod p = m_i \bmod p. \text{ Справедливо также}$$

$$m_i = b_i \cdot (a_i)_{p-1-x} \bmod p. \quad (5.8)$$

36 Потоковое шифрование. Типы. Гаммирование в потоковом шифровании.

Особенности:

1. Операции зашифрования и расшифр. вып-ся поразрядно.

2. Каждый символ шифротекста получается в рез-те поразрядной операции слож. по модулю два символа откр.текста и символа ключа

3. Поточный шифратор и деш-р требует задания начального значения ключа

4. Пот.шифры исп-ся в специальных приложениях и редко обсуждаются

Важнейшее достоинство ПШ перед блочными - высокая скорость шифрования - обеспечивается шифрование практически в реальном масштабе времени

Классический ПШ – Шифр Вернама (One-time pad — схема одноразовых блокнотов):

Зашифрование - открытый текст объединяется операцией «XOR» с ключом.

Ключ (гамма) должен обладать тремя критически важными свойствами:

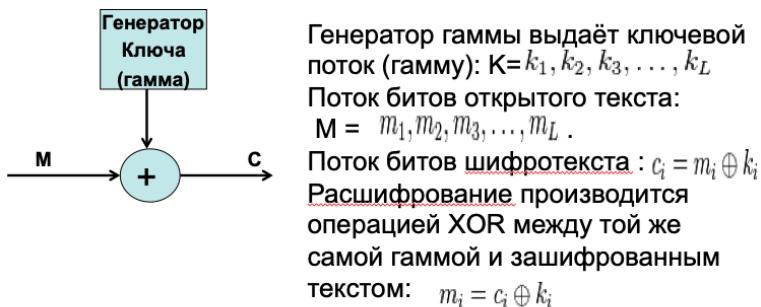
• быть истинно случайным (последовательность, полученная с использованием любого алгоритма, является не истинно случайной, а псевдослучайной);

• совпадать по размеру с заданным открытым текстом;

• применяться только один раз.

Шифр Вернама является самой безопасной криптосистемой из всех возможных.

Идея гаммирования для ПШ



Если последовательность битов гаммы не имеет периода и выбирается случайно, то **взломать шифр невозможно**.

Типы поточных шифров:

1. Синхронные –

- поток гаммы генерируется независимо от открытого текста и шифротекста;
- для успешного расшиф-я необходимо синхрон-ть ключ с шифротекстом;

Свойства:

1. Искажение одного символа в шифротексте искажает только один символ в расшифр-м тексте (+),
2. Защита от любых вставок и удалений шифротекста, так как они приведут к потере синхронизации и будут обнаружены (+)
3. Нарушение синхр-ии (добавление или удаление символа) приводит к искажению всех символов после потери синхр-ии (-)

2. Самосинхронизирующиеся (асинхронные) –

- значение ключа зависит либо от исх текста, либо от шифротекста;
- поток ключей создается функцией ключа и фиксированного числа знаков шифротекста (N): внутреннее состояние генератора является функцией предыдущих N битов шифротекста - генератор потока ключей (при расшифровании), приняв N битов, автоматически синхронизируется с шифрующим генератором

Свойства:

- Так как каждый знак открытого текста влияет на следующий шифротекст, статистические свойства открытого текста распространяются на весь шифротекст (+),
- ошибочно удаленный или добавленный символ (бит) вызывает только ограниченное кол-во ошибочных символов в дешифрованном тексте, после чего правильный текст восстанавливается (+)
- каждому неправильному биту шифротекста соответствуют N ошибок в открытом тексте (-)

37 Генерация ключевой информации для потокового шифрования.

Генераторы ПСП на основе регистров сдвига.

Основная идея современных потоковых шифров – реализовать концепцию одноразового блокнота, используя секретный ключ меньшей длины, из которого для гаммы генерируется псевдослучайная числовая последовательность, похожая на случайную.

Безопасность системы полностью зависит от свойств генератора потока ключей. Если этот генератор выдает бесконечную строку нулей, шифртекст будет совпадать с открытым текстом и преобразование будет бессмысленным. Если генератор потока ключей выдает повторяющийся, например, 16-битный шаблон, криптостойкость системы будет пренебрежимо мала. В случае бесконечного потока случайных битов криптостойкость потокового шифра будет эквивалентна криптостойкости одноразового блокнота.

Регистры сдвига (РС) используются в качестве генераторов ПСП в силу простоты реализации на основе цифровой логики. РС с линейной обратной связью (РСЛОС) состоит из двух частей: собственно РС и функции обратной связи.

Функция обратной связи реализуется с помощью сумматоров сложения по модулю 2

Последний разряд РС в каждом такте формирует очередной символ псевдослучайной последовательности (ПСП).

Выходная последовательность зависит, главным образом, от начального состояния каждой ячейки регистра.

Структура генератора описывается уравнением $C(x) = 1 + c_1 \cdot x + c_2 \cdot x^2 + \dots + c_L \cdot x^L$.

Вспомним, что важным свойством многочлена $C(x)$ является приводимость. Многочлен называется приводимым, если он может быть представлен как произведение двух многочленов меньших степеней с коэффициентами из данного поля. Если нет, то многочлен называется неприводимым. Если многочлен является неприводимым, то период ПСП будет максимально возможным: $2L - 1$.

$$C(x) = 1 + c_1 \cdot x + c_3 \cdot x^3: (310)$$

Положим, что начальное состояние ячеек регистра будет 001.

После первого такта работы генератора будут выполнены следующие операции:

- 1) содержимое ячейки 2 сформирует первый символ выходной последовательности (ПСП) генератора: этим символом будет 1;
- 2) в ячейку 0 будет записана сумма по модулю 2 начальных значений 0-й и 2-й ячеек: $(0 + 1) \bmod 2 = 1$;
- 3) в 1-ю ячейку будет записано состояние 0-й ячейки: 0;
- 4) во 2-ю ячейку будет записано состояние 1-й ячейки: 0.

Таким образом, после первого такта (цикла) работы генератора состояние его ячеек будет таким: **100** (жирным выделен очередной символ ПСП).

Легко вычислить состояние ячеек после второго цикла: **110**. После седьмого цикла состояние ячеек повторит начальное: 001. Это означает, что период ПСП равен 7.

38 Особенность шифра Вернама.

Шифр Вернама — система симметричного шифрования, изобретённая в 1917 году.

Шифр Вернама является примером системы с абсолютной криптографической стойкостью При этом он считается одной из простейших крипtosистем

Для получения шифротекста открытый текст объединяется операцией «исключающее ИЛИ» с секретным ключом. Так, например, при применении ключа (1 1 1 0 1) на букву «А» (1 1 0 0 0) получаем зашифрованное сообщение (0 0 1 0 1)

Зная, что для принимаемого сообщения имеем ключ (1 1 1 0 1), легко получить исходное сообщение той же операцией: (00101)XOR (11101)=(11000)

Для абсолютной криптографической стойкости ключ должен обладать тремя критически важными свойствами:

1. Иметь случайное равномерное распределение: $P(k)=1/2^{\{N\}}$, где k — ключ, а N — количество бинарных символов в ключе;
2. Совпадать по размеру с заданным открытым текстом;
3. Применяться только один раз

Без знания ключа такое сообщение не поддаётся анализу. Даже если бы можно было перепробовать все ключи, в качестве результата мы получили бы все возможные сообщения данной длины плюс колоссальное количество бессмысленных дешифровок, выглядящих как беспорядочное нагромождение букв.

39 Стеганографические методы защиты информации. Классификация и области использования. Метод наименее значащих бит.

Стеганография – это способ передачи или хранения информации с учетом сохранения в тайне самого факта передачи/хранения.

Стеганосистема состоит из 4 компонентов: 1. Сообщение, 2. Контейнер (это любая информация, данные, оболочка, помогающая скрыть сообщение), 3. Стеганографический ключ (это секретная информация, известная только законному пользователю, которая определяет конкретный вид алгоритма сокрытия), 4. Канал передачи.

Методы текстовой стеганографии: 1) Синтаксические, 2) Лингвистические, 3) На основе избыточности.

Синтаксические – те, которые не затрагивают синтаксис. Примеры: Line-shift coding - изменение интервалов между строк, Word-shift coding - различная длина пробелов между словами, Feature coding -(внесение специфических изменений в шрифты (начертания отдельных букв), Метод изменения интервала табуляции, Метод увеличения длины строки (с помощью пробелов), Использование регистра букв, Метод невидимых символов.

Лингвистические - те, которые затрагивают синтаксис. Примеры: метод переменной длины слова, метод первой буквы, метод синонимов, мимикия, спам.

Пример мимикрии:

Пример 7.2. Необходимо передать секретное сообщение 10101, используя следующее бинарное дерево:

Старт → существительное
существительное → Илья || Иван
глагол → поехал куда || пошел куда
куда → на работу, чтобы зачем || домой, чтобы зачем
зачем → забрать что || взять что
что → деньги || одежду.

Внедрение секретного кода на основе бинарного дерева

Узлы принятия решения	Скрытый бит	Ответы, полученные в результате внедрения кода 10101
Старт	-	старт → существительное
Существительное	1	существительное → Иван
Иван глагол	0	глагол → поехал
Иван поехал куда	1	куда → домой, чтобы
Иван поехал домой, чтобы зачем	0	зачем → забрать
Иван поехал домой, чтобы забрать что	1	что → одежду

Окончательно получилось следующее предложение: *Иван поехал домой, чтобы забрать одежду.*

На основе избыточности - Метод LSB (Least Significant Bit – наименее значащий бит) основывается на ограниченных способностях зрения или слуха человека, вследствие чего людям тяжело различать незначительные вариации цвета или звука.

Суть метода: заменяем младшие биты в байтах, отвечающих за кодирование цвета. Сначала разбиваем байт секретного сообщения на 4 2-х битовые части **11|00|10|11**. Исходные байты в изображении: 11101100, 01001110, 01111100, 0101100111. Вставляем секретное сообщение: 11101111, 01001100, 01111110, 0101100111.

Методы компьютерной стеганографии:



Использование стеганографических систем является наиболее эффективным при решении проблемы защиты информации с ограниченным доступом. Это означает возможность скрытой передачи информации. Кроме указанного направления, стеганография является одним из перспективных средств для аутентификации и

маркировки авторской продукции с целью защиты авторских прав на цифровые объекты от пиратского копирования.

40 Текстовая стеганография. Многоключевая модель стеганографической системы.

Стеганографическая система (стегосистема) – совокупность средств и методов, которые используются для формирования скрытого канала передачи информации.

Стегосистема образует стегоканал, по которому передается (или в котором хранится) заполненный контейнер. Этот канал считается подверженным воздействиям со стороны нарушителей.

При построении стегосистемы должны учитываться следующие положения:

- противник имеет полное представление о стеганографической системе и деталях ее реализации; единственной информацией, которая остается неизвестной потенциальному противнику, является ключ, с помощью которого только его держатель может установить факт присутствия и содержание скрытого сообщения;
- если противник каким-то образом узнает о факте существования скрытого сообщения, это не должно позволить ему извлечь подобные сообщения в других данных до тех пор, пока ключ хранится в тайне;
- потенциальный противник должен быть лишен каких-либо технических и иных преимуществ в распознавании или раскрытии содержания тайных сообщений.

Любая стегосистема должна отвечать следующим требованиям:

- свойства контейнера должны быть модифицированы, чтобы изменение невозможно было выявить при визуальном контроле. Это требование определяет качество сокрытия внедряемого сообщения: для обеспечения беспрепятственного прохождения стегосообщения по каналу связи оно никоим образом не должно привлечь внимание атакующего;
- стегосообщение должно быть устойчиво к искажениям, в том числе и злонамеренным. В процессе передачи изображение (звук или другой контейнер) может претерпевать различные трансформации: уменьшаться или увеличиваться,

преобразовываться в другой формат и т. д. Кроме того, оно может быть сжato, в том числе и с использованием алгоритмов сжатия с потерей данных;

- для сохранения целостности встраиваемого сообщения необходимо использование кода с исправлением ошибки;
- для повышения надежности встраиваемое сообщение должно быть продублировано.

Многоключевая модель информационной системы предполагает интегрированное использование различных методов стеганографии, криптографии для повышения криптостойкости системы.

Стеганографическая система (стегосистема) — объединение методов и средств, используемых для создания скрытого канала для передачи информации.

- Сообщение — общее название передаваемой скрытой информации, будь то лист с надписями молоком, голова раба или цифровой файл.
- Контейнер — любая информация, используемая для сокрытия тайного сообщения.
 - Пустой контейнер — контейнер, не содержащий секретного послания.
 - Заполненный контейнер (стегоконтейнер) — контейнер, содержащий секретное послание.

В компьютерной стеганографии в качестве контейнеров могут быть использованы различные оцифрованные данные: растровые графические изображения, цифровой звук, цифровое видео, всевозможные носители цифровой информации, а также текстовые и другие электронные документы.

- Стеганографический канал (стегоканал) — канал передачи стегоконтейнера.
 - Ключ (стегоключ) — секретный ключ, нужный для сокрытия стегоконтейнера. Ключом (стегоключом) называют секретную информацию, известную только законному пользователю, которая определяет конкретный вид алгоритма сокрытия.
- Ключи в стегосистемах бывают двух типов: закрытые (секретные) и открытые. Если стегосистема использует закрытый ключ, то он должен быть создан или до начала обмена сообщениями, или передан по защищенному каналу.
- Стегосистема, использующая открытый ключ, должна быть устроена таким образом, чтобы было невозможно получить из него закрытый ключ. В этом случае открытый ключ можно передавать по незащищенному каналу.

Текстовая стеганография

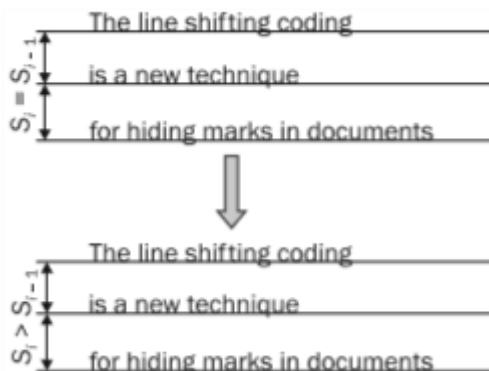
Виды:

- Синтаксические — не затрагивают семантику текстового сообщения.

- Лингвистические (классика жанра) — трансформируют текстовый файл, сохраняя смысловое содержание текста.

Синтаксические методы

Line-shift coding (изменение расстояния между строками электронного текста). Он также называется методом изменения межстрочных интервалов. Его сущность заключается в том, что используется текст с различными межстрочными расстояниями. Выделяется максимальное и минимальное расстояния между строками, позволяющее кодировать соответственно символы «1» и «0» осаждаемого сообщения (рис. 7.3). Разница в межстрочных расстояниях авторами изменялась на 1/300 дюйма (это расстояние было привязано к существовавшей в то время разрешающей способности монитора). Очевидным недостатком метода является его низкая эффективность: размер в битах осаждаемой информации не может превысить количество строк в контейнере.



Word-shift coding (изменение расстояния между словами в одной строке электронного текста). Суть метода состоит в том, что осаждение информации основано на модификации расстояния между словами текста-контейнера. Аналогично предыдущему методу, выделяется максимальное и минимальное расстояния между словами, обозначающие соответственно символ «1» и «0», и остальные расстояния, или некоторые из них, увеличивают или уменьшают до размеров уже выделенных. Частный случай этого метода – метод изменения количества пробелов (рис. 7.4). Данный рисунок показывает пример внедрения в текст-контейнер бинарной последовательности 0101100100111010. Как видно, переход с одинарного пробела на двойной кодирует «1», переход же с двойного пробела на одинарный кодирует «0».

.. 01→0	This distressed the monks and terrified them. They were not used to hearing these awful beings called names, and they did not know what might be the consequence. There was a dead silence now; superstitious bodings were in every mind. The magician began to pull his wits together, and when he presently smiled an easy, nonchalant smile, it spread a mighty relief around; for it indicated that his mood was not destructive.
.. 10→1	

0101100100111010

Данный метод имеет недостатки. Во-первых, он мало эффективен, так как необходим контейнер большого объема (объем скрытых данных в данном случае приблизительно равен одному биту на 160 байт текста). Во-вторых, возможность сокрытия зависит от структуры текста (некоторые тексты, например белые стихи, не имеют четких признаков конца). В-третьих, текстовые редакторы часто автоматически добавляют символы пробела после точки.

Feature coding (внесение специфических изменений в шрифты (начертания отдельных букв)). Этот метод заключается в изменении написания отдельных букв используемого стандартного шрифта (рис. 7.5).

:S AND 1 Incremental Mod
:S AND 1 Incremental Mod
:S AND 1 Incremental Mod

Визуально заметны различные образы, соответствующие буквам с верхними (например, I, t, d) или нижними (например, a, g) выносными элементами. Так, букву «A» можно модифицировать, незначительно укорачивая длинную нижнюю часть буквы. При этом можно закодировать стегосообщение так, что модифицированная буква будет означать «1», а немодифицированная – «0».

Модифицировать можно несколько букв. Таким образом, объем встраиваемого сообщения будет увеличиваться. Результат внедрения секретного сообщения «1» в текст-контейнер «A», при использовании метода feature coding и текстового процессора MS Office Word 2007, показан на рис. 7.6.



Пример применения метода **feature coding**: а – пустой контейнер; б – заполненный контейнер (со стегосообщением «1»)

Известны иные методы текстовой стеганографии. Кратко охарактеризуем их.

Метод изменения интервала табуляции. Аналогичен вышеописанному методу изменения количества пробелов, только в этом случае меняется не количество пробелов, а соответственно расстояние между строками и интервал табуляции.

Null chipper (дословно – несуществующий, нулевой лепет). Предполагает размещение тайной информации на установленных позициях слов или в определенных словах текста-контейнера, который, как правило, лишен логического смысла (как видно, действительно лепет). На рис. 7.7 показан пример реализации метода – скрытой информацией являются первые символы слов.

«President's embargo ruling should have immediate notice. grave situation affecting international law. statement foreshadows ruin of many neutrals. yellow journals unifying national excitement immensely».

Pershingsailsfromnyjune! → Pershing Sails From NY June I.

Рис. 7.7. Пример реализации метода null chipper

Метод увеличения длины строки. Предусматривает искусственное увеличение длины каждой строки за счет пробелов: например, одному пробелу соответствует логический 0, двум – 1 (рис. 7.8).

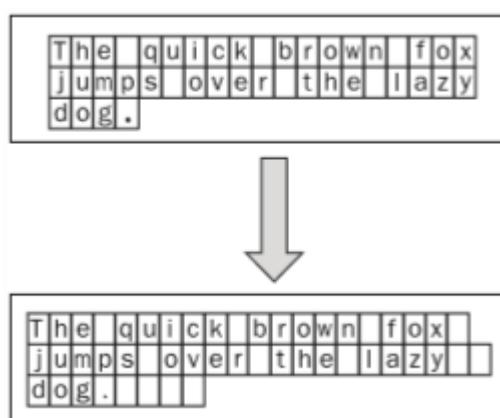


Рис. 7.8. Иллюстрация метода увеличения длины строки
Преимущество такого метода кодирования состоит в том, что оно может быть выполнено с любым текстом; изменения в формате резко не бросаются в глаза читателю, обеспечивается передача большего числа скрытых данных по сравнению с предыдущим методом (примерно 1 бит на 80 байт содержимого контейнера). Недостаток метода состоит в том, что некоторые компьютерные программы (например, Sendmail) могут неосторожно удалять дополнительные пробелы. Помимо этого, скрытые таким образом данные не всегда могут быть восстановлены с печатной копии документа.

Использование регистра букв. Для обозначения бита секретного сообщения, представленного единицей, используется символ нижнего регистра, а нулем – верхнего (или наоборот). Например, секретный текст, состоящий из одной буквы «А», необходимо внедрить в текст-контейнер «steganography». Для этого используем двоичное представление кода символа «А» – «01000001». Далее предположим, что для обозначения бита секретного сообщения, представленного единицей, используется символ верхнего регистра, а нулем – нижнего. Результат внедрения секретного сообщения «А» в текст-контейнер «steganography» показан на рис. 7.9.

«A» – «01000001» Контейнер – «steganography» «A» + «steganography» → «01000001» + «steganography» → «sTeganoGraphy»

Рис. 7.9. Иллюстрация метода на основе регистра клавиатуры
Существует модификация данного метода, основанная на применении различных алфавитов, в которых используются символы, имеющие одинаковое начертание, но различную кодировку (например, а, е, р, т, с).

Метод невидимых символов. Пробел кодируется символом с кодом 32, но в тексте его можно заменить также символом, имеющим код 255 (или 0), который является «невидимым» и отображается как пробел. Методы могут применяться независимо и совместно, сохраняют исходный смысл текста, а обеспечиваемые ими показатели плотности кодирования при совмещении складываются. Рассмотренные методы работают успешно до тех

пор, пока тексты представлены в коде ASCII. Существуют также стеганографические методы, которые интерпретируют текст как двоичное изображение. Необходимо отметить, что *данные методы не чувствительны к изменению масштаба документа, что обеспечивает им хорошую устойчивость к большинству искажений, которые могут иметь место при активных атаках.*

Описанные выше методы – синтаксические – легко применяются к любому тексту, независимо от его содержания, назначения и языка. *Синтаксические системы стеганографии легко реализуются в программном коде*, так как они полностью автоматические и не требуют вмешательства оператора. Однако синтаксические методы неустойчивы к форматированию текста (вспомним робастность систем на основе цифрового водяного знака (ЦВЗ)), и поэтому информация может быть потеряна при простом применении иного стиля форматирования текста-контейнера, скрывающего в себе стегосообщение. К тому же с помощью синтаксических методов можно передать незначительное количество информации. В литературе отсутствуют результаты экспериментального исследования эффективности проанализированных методов с проверкой на больших объемах данных.

Лингвистические методы текстовой стеганографии

Стеганографические методы, основанные на лексической структуре текста, обладают большими возможностями.

Лингвистическая стеганография занимается скрытым кодированием произвольной информации, представленной в двоичном виде, в текстах.

Необходимо отметить, что осмысленность и внешняя «безобидность» текста должна сохраниться. К лексическим методам встраивания скрытой информации в текстовые файлы-контейнеры относят метод переменной длины слова, метод первой буквы, метод синонимов и другие.

Одним из наиболее обсуждаемых методов является метод, основанный на системе синонимов языка, используемого для написания электронного текста. Проведенные исследования для случая английского языка показали, что среднее количество синонимов в одном подмножестве синонимов равняется 2,56.

Минимальное количество синонимов в одном множестве синонимов равняется 2, а максимальное 13.

Отличительной особенностью методов лексической стеганографии является то, что пользователь, как правило, должен сам составлять (или видоизменять) тот объект (текст-контейнер), в котором будет передаваться или храниться тайная информация.

Так, при использовании **метода переменной длины** пользователю, который хочет послать секретное сообщение, необходимо сгенерировать (набрать) текст, в котором слова должны иметь соответствующую длину. Длина этих слов зависит от секретного сообщения и способа кодирования. Обычно одно слово текста-контейнера определенной длины кодирует два бита информации из стегосообщения. Например, слова текста длиной в 4 и 8 символов могут означать комбинацию бит «00», длиной в 5 и 9 – «01», 6 и 10 – «10», 7 и 11 букв – «11». Слова короче 4 и длиннее 11 букв можно вставлять где угодно для лексической и грамматической связки слов в предложении. Программное приложение, которое декодирует принятое сообщение, будет просто игнорировать их.

При использовании **метода первой буквы** можно передавать ещё больше скрытой информации в одном слове: обычно это три или четыре бита. Программа-помощник в этом методе накладывает ограничение уже не на длину слова, а на первую (можно на вторую) букву. Обычно одну и ту же комбинацию могут кодировать несколько букв, например, комбинацию «101» означают слова, начинающиеся с «А», «Г» или «Т». Это дает большую свободу выбора оператору, придумывающему стегосообщение, и текст не будет нелепым и не содержащим смысла.

Другим, не менее распространенным лексическим методом передачи скрытой информации является **мимикрия**. Мимикрия генерирует осмысленный текст, используя синтаксис, описанный в Context Free Grammar (CFG), и встраивает информацию, выбирая из CFG определенные фразы и слова. Грамматика CFG – это один из способов описания языка, который состоит из статических слов и фраз языка, а также узлов. Узлы в простейшем случае представляют собой места в генерируемом тексте, где может быть принято решение, какое слово или фразу дальше необходимо вставить в текст. Мимикрия создает бинарное дерево, которое основано на

возможностях CFG, и составляет текст, выбирая те листья дерева, которые кодируют нужный бит.

Недостатками этого метода являются невозможность передавать большие объемы информации, а также низкая производительность метода. Кроме того, необходимо отметить невысокую скрытность секретного сообщения, которое в сильной мере влияет на структуру передаваемого текста.

Еще одним методом, близким к мимикрии, является **Spammimic** (**уподобление спаму**). Здесь в качестве контейнера используется обычный спам (или любой нейтральный текст, см. рис. 7.10), внутри которого размещаются установленным обеими сторонами способом значащие символы (стегосообщение).

Dear Friend, Especially for you - this red-hot intelligence. We will comply with all removal requests. This mail is being sent in compliance with Senate bill 2116, Title 9; Section 303! THIS IS NOT A GET RICH SCHEME. Why work for somebody else when you can become rich inside 57 weeks. Have you ever noticed most everyone has a cellphone & people love convenience. Well, now is your chance to capitalize on this. WE will help YOU SELL MORE and sell more! You are guaranteed to succeed because we take al the risk! But don't believe us. Ms. Simpson of Washington tried us and says "My only problem now is where to park al my cars". This offer is 100% legal. You will blame yourself forever if you don't order now! Sign up a friend and you'll get a discount of 50%. Thank-you for your serious consideration of our offer. Dear Decision maker: Thank-you for your interest in our briefing.
If you are not interested in our publications and wish to ...

Рис. 7.10. Текст-контейнер с осажденным в нем сообщением «Здравствуйте!»

Существует и множество других методов преобразования текста. В любом случае, при разработке эффективных лексических стеганографических методов необходимо искать золотую середину: контейнер должен быть «плотно» насыщен стегоинформацией и при этом совершенно не должен выделяться из обычной общей массы файлов такого же формата и наполнения.

В табл. 7.2 представлены результаты выполненного автором данного пособия сравнительного анализа эффективности некоторых из проанализированных методов.

**Сравнительные результаты анализа эффективности
синтаксических методов текстовой стеганографии**

Метод	Количество стегознаков	Плотность заполнения, %	Часть стегосообщения, содержащаяся в каждом символе контейнера, бит
Line-shift coding	811 998	1,2	0,013
Word-shift coding	8 349 980	13,1	0,132
Feature coding	49 145 754	77,6	0,776
Метод изменения регистра буквы	49 145 754	77,6	0,776
Метод изменения цвета символов	63 328 767	100,0	15,000
Метод изменения масштаба символов	63 328 767	100,0	3,000
Изменения цветовых координат ²²	63 328 767	100,0	3,000

Под «плотностью заполнения» (третий столбец таблицы) будем понимать отношение стегознаков в пустом контейнере к общему числу символов в пустом контейнере. Например, для метода изменения регистра буквы и метода feature coding стегознаками являются буквы любого алфавита, т. е. пробелы, цифры, специальные знаки и символы не учитываются. А в методах изменения цвета символов, изменения масштаба символов стегознаками являются все символы документа, в том числе специальные знаки и символы.

41 Понятие эллиптической кривой. Принципы построения крипtosистемы на эллиптических кривых

Эллиптические кривые – математический объект, который может быть определен над любым полем. В криптографии обычно используются конечные поля. Для точек на эллиптической кривой вводится операция сложения, которая играет ту же роль, что и операция умножения в крипtosистемах RSA и Эль-Гамала. Еще одним преимуществом крипtosистем на эллиптических кривых является высокая скорость обработки информации. Например, уровень стойкости, который достигается, скажем, в RSA при использовании 1024-битных ключей, в системах на эллиптических кривых реализуется при том же параметре длиной 160 битов.

Их безопасность, как правило, основана на трудности решения задачи дискретного логарифмирования в группе точек эллиптической кривой над конечным полем. Этим и обусловлена их высокая криптостойкость по сравнению с другими алгоритмами.

Крипtosистемы на эллиптических кривых, как, впрочем, и другие крипtosистемы с открытым ключом, нецелесообразно применять для шифрования больших объемов данных. Но зато их можно эффективно использовать для систем цифровой подписи и ключевого обмена.

Принципы:

1. Для использования эллиптической криптографии все участники должны согласовать все параметры, определяющие эллиптическую кривую, т.е. набор параметров криптографического протокола. Эллиптическая кривая определяется константами a и b
2. Размер поля должен как минимум в два раза превосходить размер ключа.
Например, для 128-битного ключа рекомендуется использовать эллиптическую кривую над полем, где p имеет длину 256 бит.

Шаг 1. Выбираем случайно простое число p . Его длина в битах ($t = \log p + 1$; понятно, что t должно быть целочисленным с округлением в большую сторону) должна быть такой, чтобы сделать невозможным применение общих методов нахождения логарифмов на кривой. Величина t в настоящее время принимается не менее 256 битов.

Шаг 2. Выбираем случайные числа a и b такие, что

$$a, b \pmod{p} \neq 0 \text{ и } (4 \cdot a^3 + 27 \cdot b^2) \pmod{p} \neq 0.$$

Следует обратить внимание на то, что при вычислении композиции точек параметр b нигде не фигурирует. Поэтому для повышения эффективности счета рекомендуют случайно выбирать только b , а a принимать равным небольшому целому числу (например, -3).

Шаг 3. Определяем число точек N на кривой, равное $\#E_p(a, b)$. Важно, чтобы число N имело большой простой делитель q , а лучше всего чтобы само было простым числом: $N = q$. Если поиск кривой с $N = q$ занимает слишком много времени, то можно допустить $N = h \cdot q$, где h – небольшое число. Еще раз следует подчеркнуть, что стойкость крипtosистемы на эллиптической кривой определяется не модулем p , а числом элементов q в подмножестве точек кривой. Но если множитель h – небольшое число, то q является величиной того же порядка, что и p . Если N не соответствует предъявляемым требованиям, то необходимо вернуться к шагу 2.

Шаг 4. Проверяем, выполняются ли неравенства $(p^k - 1) \pmod{q} \neq 0$ для всех k , $0 < k < 32$. Если нет, то возвращаемся к шагу 2. Эта проверка предотвращает возможность некоторых видов атак и исключает из рассмотрения так называемые суперсингулярные кривые и кривые с $N = p - 1$.

Шаг 5. Проверяем, выполняется ли неравенство $q \neq p$. Если нет, то возвращаемся к шагу 2. Для кривых с $q = p$, которые называются аномальными, существуют эффективные методы вычисления логарифмов, т. е. взлома шифра.

Шаг 6. Необходимая для криптографических приложений кривая уже получена. Имеем параметры p, a, b , количество точек N и размер подмножества точек q . Чтобы получить случайную точку на кривой, берем случайное число $x < p$, вычисляем $e \equiv (x^3 + a \cdot x + b) \pmod{p}$ и пытаемся извлечь квадратный корень $y = \sqrt{e} \pmod{p}$. Если корень существует, то получаем точку (x, y) , в противном случае пробуем другое число x .

$$\begin{aligned}
 k = 100 & \quad 100P = 32P + 3L0 + 3L + 2P + 2L \\
 k = 256 & \quad 256P = 100P + 100P + 32P + 20P + 2P + 2D \\
 k = 751 & \quad 751P = 256P + 256P + 100P + 100P + 32P + 5P + 2P \\
 k = 1024 & \quad 1024P = 751P + 256P + 100P + 5P + P
 \end{aligned}$$

11. $E_5(6; -8)$

$$y^2 \equiv x^3 + 6x - 8 \pmod{5}$$

$$4 \cdot 6^3 + 27 \cdot (-8)^2 \not\equiv 0 \pmod{5} \quad \text{---}$$

$$1 \not\equiv 0 \pmod{5}$$

$$0^2 \pmod{5} = 0$$

$$1^2 \pmod{5} = 1$$

$$2^2 \pmod{5} = 4$$

$$3^2 \pmod{5} = 4$$

$$4^2 \pmod{5} = 1$$

$$5^2 \pmod{5} = 0$$

$$\begin{array}{l} 0^2 \pmod{5} = 0 \\ 1^2 \pmod{5} = 1 \\ 2^2 \pmod{5} = 4 \\ 3^2 \pmod{5} = 4 \\ 4^2 \pmod{5} = 1 \end{array}$$

$$\left\{ \begin{array}{l} 0, 1, 4 \end{array} \right\}$$

$$y^2 \equiv 0^2 + 6 \cdot 0 - 8 \pmod{5} \quad \text{---} \quad 0 \pmod{5}$$

не проходит через $(0, 1), (0, 4)$

$$y^2 \equiv 1^2 + 6 \cdot 1 - 8 \pmod{5} \quad \text{---} \quad 3 \pmod{5}$$

не проходит через $(1, 3)$

$$y^2 \equiv 4^2 + 6 \cdot 4 - 8 \pmod{5} \quad \text{---} \quad 1 \pmod{5}$$

$$(4, 1), (4, 4)$$

11. $E_{11}(1, 2)$

$$y^2 \equiv 1^3 + 1 \cdot 1 + 2 \pmod{11}$$

$$\begin{array}{l} 10^2 \pmod{11} = \\ 11^2 \pmod{11} = \end{array}$$

$$\left\{ \begin{array}{l} 1, 0 \end{array} \right.$$

$$x = 0$$

$$y^2 = 0^3 +$$

не про

$$x = 1$$

$$y^2 = 1^3 +$$

$$x = 3$$

42 Представление и описание эллиптической кривой на основе алгебраической геометрии

В общем случае кубические уравнения для эллиптических кривых имеют вид

$$y^2 + a \cdot x \cdot y + b \cdot y = x^3 + c \cdot x^2 + d \cdot x + e, \quad (5.13)$$

где a, b, c, d, e являются действительными числами, удовлетворяющими некоторым простым условиям. Такие уравнения называются еще уравнениями третьего порядка

С другой стороны, эллиптическая кривая – это набор точек (x, y) , удовлетворяющих уравнению (5.13) для переменных (x, y) и констант (a, b, c, d, e) , принадлежащих множеству F , где F – поле.

Вспомним некоторые важные характеристики поля. Конечным полем называется алгебраическая система, которая состоит из конечного множества F и двух бинарных операций (сложения и умножения).

Порядком поля называется количество элементов в поле (во множестве F).

Определение эллиптической кривой включает также некоторый элемент, называемый несобственным элементом (или бесконечным элементом, или нулевым элементом).

$y^2 = (x^3 + a \cdot x + b) \bmod p$ (5.14), Эллиптическая кривая или множество $E_p(a, b)$ состоит из всех точек (x, y) , $x > 0$, $y < p$, удовлетворяющих уравнению (5.14), и точки в бесконечности O_{SEP} .

Определение 5.1. Порядок эллиптической кривой – это число, которое показывает количество точек кривой над конечным полем.

Определение 5.2. Аддитивной абелевой группой называется множество A с операцией сложения, обладающей следующими свойствами:

- [1] $a + b = b + a, \forall a, b \in A$ (коммутативность);
- 2) $(a + b) + c = a + (b + c), \forall a, b \in A$ (ассоциативность);
- 3) во множестве A существует такой элемент O (нуль), что $a + O = a, \forall a \in A$;
- 4) для любого элемента $a \in A$ существует такой элемент $-a \in A$ (противоположный элемент), что $a + (-a) = O$.

Определение 5.3. Наименьшее значение числа q , для которого выполняется равенство $q \cdot P = O$, называется порядком точки P .

Определение 5.4. Порядок группы точек эллиптической кривой равен числу различных точек ЭК, включая точку O .

Определение 5.5. Генерирующая (базисная) точка эллиптической кривой – это такая точка G на эллиптической кривой, для которой минимальное значение n , такое что $n \cdot G = O$, является очень большим простым числом.

43 Арифметические операции в эллиптической криптографии

Эллиптические кривые – математический объект, который может быть определен над любым полем. В криптографии обычно используются конечные поля

Арифметические операции в эллиптической криптографии производятся над точками кривой. Основной операцией является **сложение**. Сложение двух точек легко представить графически

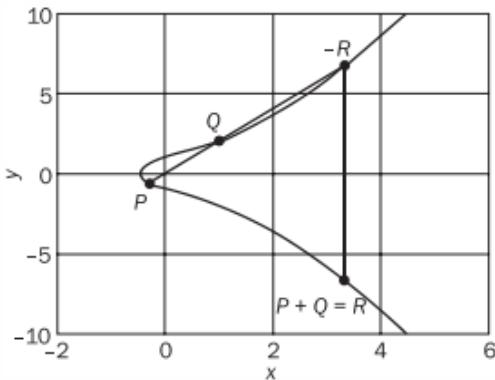


Рис. 5.4. Пояснение к операции сложения двух точек эллиптической кривой $y^2 = x^3 + 2 \cdot x + 1$

Как видно из рис., для сложения точек P и Q необходимо провести между ними прямую линию, которая обязательно пересечет кривую в какой-либо третьей точке ($-R$; иногда эту точку обозначают R). Отразим точку $-R$ относительно горизонтальной оси координат и получим исходную точку: $P + Q$ – результат сложения: $P + Q = R$, т. е. точки R и $-R$ симметричны относительно горизонтальной оси.

По определению, эллиптическая кривая обладает следующим свойством: *если три ее точки лежат на одной прямой, то их сумма равна O .*

Это свойство позволяет описать основные правила сложения, а также умножения точек эллиптической кривой:

- пусть P и Q – две различные точки эллиптической кривой, и P не равно Q .

Проведем через P и Q прямую. Она пересечет эллиптическую кривую только в одной точке, называемой $-R$. Точка $-R$ отображается относительно оси X в точку R , равную сумме точек P и Q ; **закон сложения точек эллиптической кривой: $P + Q = R$** ;

• прямая, проходящая через точки R и $-R$, является вертикальной прямой, которая не пересекает эллиптическую кривую ни в какой третьей точке; если $R = (x, -y)$, то $R + (x, y) = O$. Точка (x, y) является отрицательным значением точки R и обозначается $-R$; таким образом, по определению $R + (-R) = O$;

• если O – нулевой элемент, то справедливо равенство $O = -O$, а для любой точки P эллиптической кривой имеем $P + O = P$;

• чтобы сложить точку P с ней самой, нужно провести касательную к кривой в точке P ; **закон удвоения точки P : $P + P = 2 \cdot P$** ;

• умножение точки P на целое положительное число k определяется как сумма k точек P : $k \cdot P = P + P + P + \dots + P$!

• скалярное умножение осуществляется посредством нескольких комбинаций сложения и удвоения точек эллиптической кривой. Например, точка $25 \cdot P$ может быть представлена как $25 \cdot P = (2 \cdot (2 \cdot (2 \cdot (2 \cdot P)))) + (2 \cdot (2 \cdot (2 \cdot P))) + P$.

Идея, надежность и криптостойкость эллиптической криптографии напрямую связаны с операцией умножения точки на целое число: задача вычисления дискретного логарифма на эллиптической кривой, заключающаяся в отыскании целого числа x по

известным точкам P и $Q = x \cdot P$, является трудноразрешимой. Если $P = (x_1, y_1)$ и $Q = (x_2, y_2)$, то $P + Q = (x_3, y_3)$ определяется в соответствии с правилам:

$$x_3 = (\lambda^2 - x_1 - x_2) \bmod p;$$

$$y_3 = (\lambda \cdot (x_1 - x_3) - y_1) \bmod p,$$

$$\text{где } \lambda = (y_2 - y_1)/(x_2 - x_1), \quad \text{если } P \neq Q,$$

$$\lambda = (3x_1^2 + a)/2y_1, \quad \text{если } P = Q.$$

Из этого следует, что число λ – угловой коэффициент секущей, проведенной через точки $P = (x_1, y_1)$ и $Q = (x_2, y_2)$. При $P = Q$ секущая превращается в касательную, чем и объясняется наличие двух формул для вычисления λ .

44 Система согласования криптографических ключей на основе эллиптической кривой

Обмен ключами с использованием эллиптических кривых может быть выполнен следующим образом. Сначала выбирается простое число p и параметры a и b для эллиптической кривой. Это задает эллиптическую группу точек $E_p(a, b)$. Затем в $E_p(a, b)$ выбирается генерирующая точка $G = (x, y)$. При этом важно, чтобы наименьшее значение q , при котором выполняется условие $q \cdot G = O$, оказалось очень большим простым числом. Параметры $E_p(a, b)$ и G криптосистемы являются параметрами, известными всем участникам. Обмен ключами между участниками информационного процесса (А и В) можно провести по следующему алгоритму.

1. Сторона А выбирает целое число k_A , меньшее q . Это число будет тайным ключом стороны А. Затем А генерирует открытый ключ: $Y_A = k_A \cdot G$, и отсылает его стороне В. Открытый ключ представляет собой некоторую точку из $E_p(a, b)$.
2. Параллельно сторона В выбирает для себя тайный ключ k_B и вычисляет открытый ключ: $Y_B = k_B \cdot G$, и отсылает его стороне А.
3. Сторона А, получив открытый ключ стороны В, генерирует секретный ключ $K_A = k_A \cdot Y_B = K$, а сторона В соответственно генерирует секретный ключ $K_B = k_B \cdot Y_A = K$.

Несложно установить, что обе стороны получили один и тот же результат, поскольку $k_A \cdot Y_B = k_A \cdot (k_B \cdot G) = k_B \cdot (k_A \cdot G) = k_B \cdot Y_A = K$.

Пример. Пусть $p = 211$; $G = (2, 2)$; $E_p(0, -4)$, что соответствует кривой $y^2 = x^3 - 4$. Стороны совместно выбирают $q = 241$. Можно подсчитать, что $241 \cdot G = O$. Тайным ключом стороны А является $k_A = 121$, поэтому открытым ключом стороны А будет $Y_A = 121(2, 2) = (115, 48)$. Тайным ключом стороны В будет значение $k_B = 203$, поэтому открытым ключом стороны В будет $Y_B = 203(2, 2) = (130, 203)$. Общим секретным ключом является одно значение, соответствующее точке на эллиптической кривой: $K = 121(130, 203) = 203(115, 48) = (161, 69)$.

45 ЭЦП. Назначение и свойства.

Определение: Электронная цифровая подпись – последовательность символов, являющаяся реквизитом электронного документа, зависящая от содержания этого документа и предназначенная для подтверждения целостности и подлинности электронного документа.

Назначения:

- аутентифицировать лицо, подписавшее сообщение; ЭЦП получается в результате криптографического преобразования электронных данных документа с использованием личного ключа ЭЦП;

- контролировать целостность сообщения; ЭЦП вычислена на основании исходного состояния документа и соответствует лишь ему, поэтому при любом случайному или преднамеренному изменении документа подпись станет недействительной;
- защищать сообщение от подделок; любая подделка должна быть выявлена путем операций сравнения соответствующих атрибутов, подписанных и полученного адресатом сообщений;
- доказать авторство лица, подписавшего сообщение; создать корректную ЭЦП можно, лишь зная закрытый ключ, известный только его владельцу (лицу, подписавшему документ).

Свойства:

- ЭЦП представляет собой бинарную последовательность (в отличие от графического образа, каковым является подпись от руки);
- указанная бинарная последовательность зависит от содержания подписываемого сообщения.

Важное свойство цифровой подписи заключается в том, что ее может проверить каждый, кто имеет доступ к открытому ключу ее автора (здесь речь идет об ЭЦП на основе алгоритмов асимметричного шифрования)

И, наконец, еще одна особенность. Обычно подписываемые с помощью ЭЦП сообщения не шифруются.

46 ЭЦП. Основные методы генерации. Атаки на ЭЦП

Существует несколько схем построения ЭЦП:

- на основе алгоритмов симметричного шифрования; авторизацией документа является сам факт зашифрования его секретным ключом;
- на основе алгоритмов асимметричного шифрования; упоминавшаяся выше в данном разделе ЭЦП на основе RSA относится именно к этому классу;
- на основе алгоритмов асимметричного шифрования и хеш-функции; это наиболее распространенная схема.

Целью **атак** на цифровые подписи является, в конечном итоге, возможность их подделки или фальсификации

Основные виды атак:

- Атака с использованием открытого ключа. Имеется в виду ключ субъекта, подписывающего документ. Криptoаналитик обладает только открытым ключом.
- Атака на основе известных сообщений. В распоряжении криptoаналитика имеются некоторые ЭЦП и соответствующие им документы.
- Адаптивная атака на основе выбранных сообщений. Криptoаналитик может получить подписи электронных документов, которые он выбирает сам.

Основные результаты или цели описанных атак можно классифицировать следующим образом:

- Полный взлом ЭЦП. Получение закрытого ключа отправителя подписанного сообщения. Это означает полный взлом алгоритма.
- Универсальная подделка подписи. Криptoаналитик находит алгоритм, позволяющий подделывать подписи для любого электронного документа.
- Выборочная подделка подписи. Дает возможность подделывать подписи для документов, выбранных криptoаналитиком.
- Экзистенциальная подделка подписи. Дает возможность получения допустимой подписи для некоторого документа, не выбиpаемого криptoаналитиком.

Современные алгоритмы генерации и использования ЭЦП оставляют криptoаналитику мало шансов на получение закрытого ключа алгоритма из-за вычислительной сложности задач, на основе которых ЭЦП построена. Более вероятен поиск коллизий.

Коллизия первого рода эквивалентна экзистенциальной подделке, а коллизия второго рода – выборочной.

47 ЭЦП на основе симметричной криптографии

Понятно, что симметричная схема цифровой подписи использует один и тот же ключ для генерации ЭЦП и ее проверки.

Основные функции ЭЦП, с формальной точки зрения, реализуются примитивной процедурой шифрования/расшифрования. Ведь аутентичность, целостность, защиту от подделок и доказательство авторства обеспечивает собственно симметричный ключ, используемый двумя сторонами в процессе обмена сообщениями.

Из этого следует, что подписывать (шифровать) следует каждый бит сообщения M , т. е. по размеру ЭЦП может превосходить подписываемый документ на несколько порядков. В силу этого серьезного недостатка идея не нашла практического применения. Однако если предусмотреть наличие в системе третьего лица – арбитра или посредника (Π), пользующегося доверием обеих сторон, то можно избежать указанного недостатка.

48 ЭЦП на основе алгоритма RSA

1.

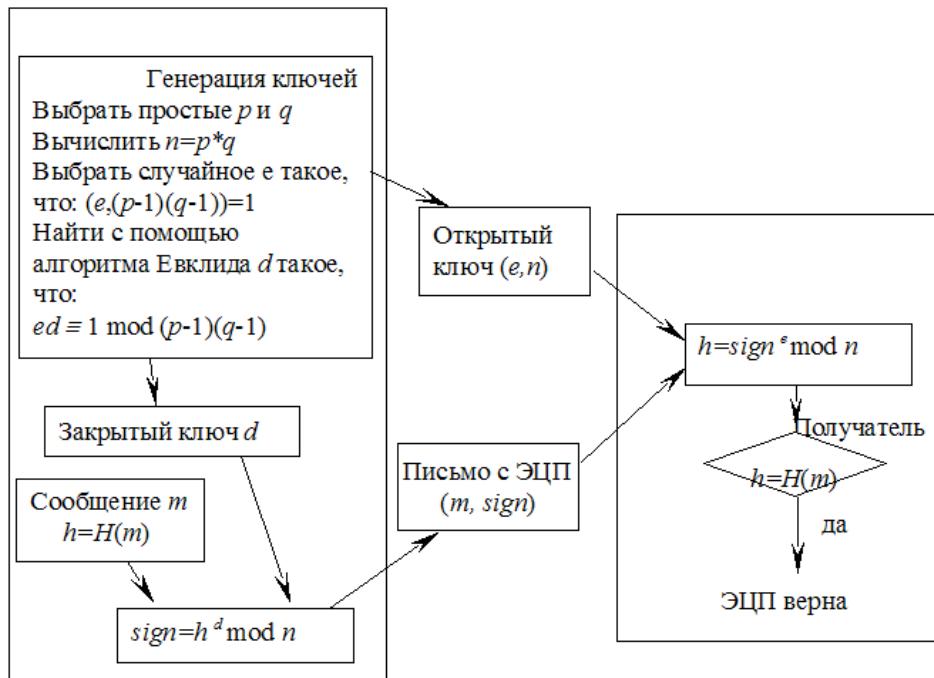


Схема ЭЦП на основе алгоритма RSA

По сути ЭЦП на основе RSA является классической схемой шифрования RSA, однако есть нюанс. Хэш сообщения шифруется тайным ключом, а проверка подписи происходит по открытому ключу(инверсия RSA).

Один из классических алгоритмов генерации ЭЦП на основе RSA отличается от рассмотренного случая лишь тем, что отправитель шифрует не само сообщение M (см. пункт 5.6.2), а его хеш $h(M)$. Понятно, что предварительно должна быть выполнена процедура генерации ключа отправителем.

Генерация ЭЦП или зашифрование

$h(M) : s = (h(M))^d \pmod{n}$, здесь пара чисел $(d \text{ и } n)$ – тайный ключ отправителя.

После этого число s , являющееся ЭЦП, присоединяется к сообщению M и пересыпается получателю

Алгоритм проверки ЭЦП на основе RSA:

- восстановление $h(M) : s^e \pmod{n} = h'(M)$; (11.2)
- вычисление хеша для полученного сообщения: $h''(M)$ (для строгости анализа применяем новое обозначение);
- сравнение хешей: $h'(M)$ и $h''(M)$; только при равенстве сравниваемых величин сообщение с подписью будет однозначно аутентифицировано

49 ЭЦП на основе симметричной криптосистемы и посредника

Основной алгоритм состоит в следующем.

1. Посредник Π вырабатывает для A и B разные (сеансовые, например) ключи: KA и KB .

2. Абонент А шифрует свое сообщение M ключом КА и отсылает его посреднику: $C = EKA(M)$.

3. П расшифровывает C ключом КА ($M = DKA(C)$), тем самым извлекает сообщение M . Присоединяет к этому сообщению подтверждение того, что автором его является абонент А (обозначим эту часть нового сообщения МД). Таким образом сформирован конкатенированный документ $M' = M || MД$. Посредник шифрует M' ключом КВ: $C1 = EKB(M')$. Зашифрованное сообщение $C1$ отсылается абоненту В.

4. Абонент В расшифровывает $C1$ ключом КВ ($M' = DKB(C1)$), тем самым извлекает сообщение M' . Из этого сообщения он получает оригинал сообщения M и подтверждение тому, что автором этого сообщения является абонент А.

Таким образом, здесь выполнены все функциональные требования, присущие ЭЦП:

- 1) подпись достоверна (П – гарант);
- 2) подпись неподдельна, так как только А и В знают ключи, использовавшиеся в процедурах (к П – абсолютное доверие);
- 3) подписанный документ нельзя изменить или подделать;
- 4) подпись нельзя отрицать. Если ключи действительно были сеансовыми, то подпись невозможно использовать повторно.

50 ЭЦП DSS.

В августе 1991 года американским институтом стандартизации ((NIST) был утвержден стандарт DSS (Digital Signature Standard – стандарт цифровой подписи) для использования в алгоритме ЭЦП DSA (Digital Signature Algorithm – алгоритм цифровой подписи). В алгоритме используются следующие параметры: p – простое число длиной от 64 до 1024 битов (число должно быть кратно 64); q – 160-битный простой множитель $p - 1$. Далее вычисляется число g : $g = v(p - 1)/q \text{ mod } p$, где v – любое число, меньшее $p - 1$, для которого выполняется условие $v(p - 1)/q \text{ mod } p > 1$. Числа p , q , v могут использоваться группой лиц.

Открытый ключ u вычисляется в соответствии с выражением $u = gx \text{ mod } p$, (11.9) где $x < q$; x – закрытый ключ. Понятно, что открытый ключ размещается в общедоступной базе данных. Вспомним, что в алгоритме подписывается не само сообщение M , а его хеш: $h(M)$. Для хеширования сообщений используется алгоритм SHA. После генерации ключей их обладатель может подписывать свои сообщения.

Генерация ЭЦП: выбирается из условия, описанного выше, число k . Подпись состоит из пары чисел, r и s : $r = (g^k \text{ mod } p) \text{ mod } q$, (11.10)

$s = (k - 1 \cdot (h(M) + x \cdot r)) \bmod q$. (11.11) Как видим, соотношения (11.10), (11.11) незначительно отличаются от (11.6) и (11.7). Для проверки подписи получатель подписанного сообщения (M, r, s) выполняет вычисления: $w = s^{-1} \bmod q$, $u_1 = (h(M) \cdot w) \bmod q$, $u_2 = (r \cdot w) \bmod q$, $a = ((gu_1 \cdot u_2) \bmod p) \bmod q$. Подпись считается достоверной, если $a = r$. На рисунке приведена общая схема алгоритма. Общая схема алгоритма ЭЦП DSA Быстродействие практической реализации рассматриваемого алгоритма можно существенно увеличить с помощью предварительных вычислений. Обратим внимание на то, что r не зависит от подписываемого сообщения M . Можно создать массив чисел k и рассчитать r для каждого k . Кроме того, можно вычислить значение $k - 1$ для каждого k . Далее брать очередную пару r и $k - 1$ для очередного сообщения M . По данным Б. Шнайера, такие предварительные вычисления по времени сопоставимы с процедурой верификации подписи. При этом длительность операций вычисления s составляет лишь несколько процентов от длительности всей процедуры генерации ЭЦП.

51 ЭЦП на основе алгоритма Эль-Гамала

Схему Эль-Гамала можно использовать как для шифрования (см. пункт 5.4.3), так и для цифровых подписей. В сравнении, например, с ЭЦП на основе RSA рассматриваемая схема обеспечивает более высокое быстродействие. Вспомним, что ключ состоит из четырех чисел. Для его генерации нужно выполнить следующие действия. 1. Выбирается простое число p и два случайных числа, меньших, чем p : числа x и g . 2. Далее вычисляется $y = g^x \bmod p$. Открытый ключ: y, g и p ; тайный ключ: x . Чтобы подписать сообщение M , обладатель используемых для ЭЦП ключей должен выбрать, как и в предыдущей схеме, случайное число k , взаимно простое с $p - 1$. Затем вычисляются числа a и b , являющиеся цифровой подписью: $a = g^k \bmod p$; (11.13) для вычисления b с помощью расширенного алгоритма Евклида (см. листинг на с. 55–56) решается уравнение $M = (x \cdot a + k \cdot b) \bmod (p - 1)$. (11.14) Для верификации подписи нужно убедиться, что выполняется равенство $y^a \cdot a^b = g^M \bmod p$. (11.15) Как видим, в классическом алгоритме ЭЦП по схеме Эль-Гамала используется не хеш сообщения M , а подписываемое сообщение целиком, которое следует представлять также числом. Пример. Положим, что $p = 11$, $g = 2$, $x = 8$, вычисляем третье значение

открытого ключа: $y = gx \bmod p = 28 \bmod 11 = 3$. Таким образом, открытый ключ: $p = 11$, $g = 2$, $y = 3$; закрытый ключ: $x = 8$. Подпись сообщения M ($M = 5$). Выбирается случайное число $k = 9$, взаимно простое с $p - 1 = 10$; вычисляется в соответствии с выражением (11.13) первый элемент ЭЦП: $a = g k \bmod p = 29 \bmod 11 = 6$. Второй элемент ЭЦП находим на основе уравнения (11.14): $M = (x \cdot a + k \cdot b) \bmod (p - 1) = 5 = (8 \cdot 6 + 9 \cdot b) \bmod 10$. Решением будет число $b = 3$. Подписанное сообщение: $M = 5$, $a = 6$, $b = 3$. Проверка подписи в соответствии с уравнением (11.15): $(36 \cdot 63) \bmod 11 = 25 \bmod 11 = 10$. В ЭЦП DSS и Эль-Гамала для каждой новой подписи необходимо использовать новое значение k . Если третья сторона добудет два сообщения, подписанные с использованием одного и того же k , ее шансы на взлом закрытого ключа отправителя значительно возрастают. Это обстоятельство мы подчеркивали при рассмотрении схем шифрования по Эль-Гамалю.

52 ЭЦП на основе эллиптической кривой.

Алгоритм:

A. Создание ключей:

- Выбираем эллиптическую кривую $E(a, b)$. Число точек на ней должно делиться на n .
- Выбираем базовую точку G принадлежащую $E(a, b)$ порядка n , $n^* G = \infty$.
- Выбираем случайное число d принадлежащее диапазону $(1, n)$.
- Вычисляем $Q = d * G$.
- В итоге: d – закрытый ключ, кортеж $\langle a, b, G, n, Q \rangle$ – открытый ключ.

B. Создание подписи:

- Выбираем случайное число k принадлежащее диапазону $(1, n)$. M – это сообщение.
- Вычисляем $k * G = (x_1, y_1)$ и $r = x_1 \pmod n$.
- Проверяем, что бы $r \neq 0$ (иначе выбираем другое k).
- Вычисляем $k^{-1} \pmod n$.
- Вычисляем $s = k^{-1} * (H(M) + d * r) \pmod n$.
- Проверяем, что бы $s \neq 0$ (иначе выбираем другое k , потому что не получится сделать проверку подписи, если будет равно 0).

C. Отправка:

- Отправляем открытый ключ – кортеж $\langle a, b, G, n, Q \rangle$ и пару (r, s) .

D. Проверка подписи:

- Проверяем, что r и s принадлежат диапазону $(1, n)$.
- Вычисляем $w = s^{-1} \pmod n$ и $H(M)$.
- Вычисляем $u_1 = H(M) * w \pmod n$ и $u_2 = r * w \pmod n$.
- Вычисляем $u_1 * G + u_2 * Q = (x_0, y_0)$ и $v = x_0 \pmod n$.

- Если $v = r$ – то подпись верна, в других случаях – нет.

53 Алгоритм К. Шнорра. Стандарт ЭЦП в РБ.

Рассматриваемая схема является основой стандарта ЭЦП в Беларуси. Алгоритм ЭЦП К. Шнорра (K. Schnorr) является вариантом алгоритма ЭЦП Эль-Гамаля.

Одной из особенностей ЭЦП Эль-Гамаля является то, что число p должно быть очень большим, чтобы сделать действительно трудной проблему дискретного логарифма. Рекомендуемая длина p должна составлять по крайней мере 1024 бита. Чтобы уменьшить размер подписи, Шнорр предложил новую схему, но с уменьшенным размером подписи.

Ключевая информация: p – простое число в диапазоне от 512 до 1024 битов; q – 160-битное простое число, делитель $(p - 1)$; любое число g ($g \neq 1$) такое, что

$$g^q \equiv 1 \pmod{p}. \quad (10.8)$$

Числа p, g, q являются открытыми и могут применяться группой пользователей.

Выбирается число $x < q$ (x является тайным ключом) и вычисляется последний элемент открытого ключа:

$$y \equiv g^{-x} \pmod{p}. \quad (10.9)$$

Секретный ключ имеет длину не менее 160 битов.

Для подписи сообщения M_0 выбирается случайное число k ($1 < k < q$) и вычисляет параметр a :

$$a \equiv g^k \pmod{p}. \quad (10.10)$$

Далее вычисляется хеш от канкатенации сообщения M_0 и числа a : $h = H(M_0 || a)$. Обратим внимание, что хэш-функция непосредственно не применяется к сообщению. Создается хеш-образ подписываемого сообщения, спереди присоединенного к числу a . Далее вычисляется значение b :

$$b \equiv (k + xh) \pmod{q}. \quad (10.11)$$

Получателю отправляются $M' = M_0 || S$; $S = \{h, b\}$.

Для проверки подписи получатель вычисляет

$$X \equiv g^b y^h \pmod{p}. \quad (10.12)$$

Затем он проверяет выполнение равенства: $h = H(M_{\text{п}}||X)$. Подпись достоверна, если равенство выполняется.

Основные вычисления для генерации подписи могут производиться предварительно. Порядок величин x и h – около 140 двоичных разрядов, порядок числа k – около 70–72 разрядов. С учетом этого сложность операций умножения можно считать ничтожно малой по сравнению с модульным умножением в схеме RSA.

Алгоритм ЭЦП Шнорра

Основа стандарта ЭЦП РБ (Claus Schnorr)

является модификацией схем Эль-Гамаля (1985) и Фиата-Шамира (1986)

p – простое число длиной примерно **512** или **1024** бит,

q – примерно **140**-битный простой множитель $(p-1)$; т.е $p - 1 \equiv 0 \pmod{q}$

выбирается любое число z ($z \neq 1$) такое, что $z^q \equiv 1 \pmod{p}$;

p, z и q являются открытыми и могут применяться группой пользователей;

Выбирается число $s < q$; вычисляется $v = z^{-s} \pmod{p}$; или $vz^s \equiv 1 \pmod{p}$

s – тайн ключ, v – открытый

Генерация ЭЦП сообщения M (A для B). A выбирает случайное число k ($k < q$) и вычисляет $x = z^k \pmod{p}$;

подпись – числа e и y: $e = h(M||x)$; $y = (k + s \cdot e) \pmod{q}$

A персылает B: M, e, y

Проверка подписи: B вычисляет $x' = z^y \cdot v^e \pmod{p}$; затем вычисляет $e' = h'(M||x')$

- **Подпись достоверна, если $e = e'$**

Пример

- Генерация ключей:

$p = 11$, $q = 5$; причем $p = 2q + 1$, т.е. $p - 1 \equiv 0 \pmod{q}$

Выбирается $z = 3$, д.б.: $z \neq 1$, и $z^q \equiv 1 \pmod{p}$: $3^5 \equiv 1 \pmod{11}$

Тайный ключ $s = 3$, тогда $v = z^s \pmod{p} = 3^3 \pmod{11}$

Или $vz^s \equiv 1 \pmod{p}$, $v = 9$

Открытый ключ: $p = 11$, $q = 5$, $z = 3$, $v = 9$

Тайный ключ: $s = 3$

- Генерация подписи: $M = 1000$

Выбирается случайное $k = 2$ ($k < q$)

вычисляется $x = z^k \pmod{p} = 3^2 \pmod{11} = 9$

конкатенация $M||x : 10009$; предположим $e = h(M||x) = 15$

вычисляется $y = (k + s^e) \pmod{q} = (2 + 3^{15}) \pmod{5} = 2$

Получателю высыпается: 1000, 15, 2

Безопасность – на трудности вычисления ДИСКР Логар

54 Протокол Kerberos.

Kerberos – это протокол аутентификации компьютерной сети, который работает на основе билетов, позволяя узлам, обменивающимся данными по небезопасной сети, безопасно подтверждать свою личность друг другу. Его разработчики нацелили его в первую очередь на модель клиент–сервер, и он обеспечивает взаимную аутентификацию— и пользователь, и сервер проверяют личность друг друга. Сообщения протокола Kerberos защищены от подслушивания и повторных атак.

Протокол Kerberos

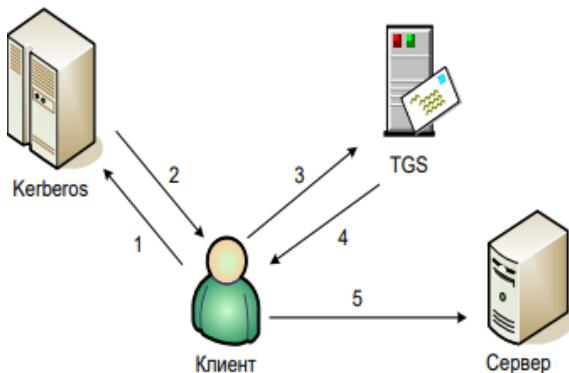


Рис.2. Общая схема взаимодействия компонент в протоколе Kerberos

- 1 — Клиент запрашивает Керберос разрешение на обращение к службе ТГС.
- 2 — После анализа предоставленных документов о возможности организации сообщения между Кл и Серв Керберос выдает Кл-ту соответствующее разрешение.
- 3 — Пользуясь разрешением службы Керберос, Кл запрашивает ТГС о выделении ему мандата на организацию канала между Клиентом и Сервером.
- 4 — Получение такого мандата.
- 5 — Клиент пересыпает соответствующее сообщение серверу

C — Клиент (Client),
S — Сервер (Server),
A — Сетевой адрес Клиента (Address) — имя Клиента,
v — Временная метка, содержащая начальное и конечное время действия мандата,
t — просто метка времени, соответствующая периоду времени, в течение которого действует сеансовый ключ,
Kx — секретный ключ объекта X,
Kx,y — сеансовый ключ для организации сеанса между X и Y,
{m}Kx — сообщение m, зашифрованное ключом Kx,
Tx,y — мандат, выданный X на использование Y,
Ax,y — аутентификатор, выданный X для Y, то есть информация, с помощью которой Y аутентифицирует X.

Операции (стрелки 1-5 на рис.2)
могут быть записаны в формализованном виде:

- 1 — Клиент-Керберос: **C, TGS**
- 2 — Kerberos-Клиенту: **{Kc,tgs}Kc; {Tc,tgs}K_{TGS}**
- 3 — Клиент-TGS: **{Ac,s}K_{c,TGS}; {Tc,tgs}K_{TGS}**
- 4 — TGS-Клиенту: **{K_{c,s}}K_{c,TGS}; {Tc,s}K_s**
- 5 — Клиент-Серверу: **{Tc,s}K_s**

Kerberos использует 2 типа удостоверений:

- Мандаты (для безопасной передачи Серверу данных о личности Клиента):

$$T_{c,s} = S, \{C, A, v, K_{c,s}\}K_s$$

Клиент не может расшифровать мандат, поскольку он не знает секретный ключ K_s , но он может предъявить его Серверу, как док-во его аутентичности, т.е. прочитать либо изменить мандат ни Клиент, ни кто-либо иной не может.

- Аутентификаторы (это дополнительная информация, предъявляемая вместе с мандатом):

$$Ac,s = \{C, t, \text{Ключ}\} K_{c,s}$$

Клиент создает аутентификатор на каждый сеанс, Ключ - является просто ключом и необязательным дополнительным элементом сеанса и все эти данные шифруются общим ключом, известным Клиенту и Серверу: $K_{c,s}$. В отличие от мандата, аутентификатор используется только один раз

55 Деструктивные программы. Классификация и методы нейтрализации.

1. **Вирусы** (viruses) — это саморазмножающиеся программы путем дописывания собственных кодов к исполняемым файлам. Вирусы могут содержать, а могут не содержать деструктивные функции.

Макровирусы - это файловые вирусы. Макровирусы заражают различные документы и электронные таблицы, такие, как, например, файлы редакторов Word и Excel. Код этих вирусов создается на макро-языках, отсюда и их название. Большинство макровирусов обладают свойствами резидентов и действуют только во время работы с инфицированным документом.

2. **Черви** (worms) — это программы, которые самостоятельно размножаются по сети и, в отличие от вирусов, не дописывают себя (как правило) к исполняемым файлам. Все черви съедают ресурсы компьютера, "нагоняют" интернет-трафик и могут привести к утечке данных с вашего компьютера.

3. **Кейлоггеры** (keyloggers) — программы, которые регистрируют на нажатия клавиш, делают снимки рабочего стола, отслеживают действия пользователя во время работы за компьютером и сохраняют эти данные в скрытый файл на диске, затем этот файл попадает к злоумышленнику.

4. **Трояны** (trojans), **троянские кони** — собирают конфиденциальную информацию с компьютера пользователя (пароли, базы данных и пр.) и тайно по сети высыпают их злоумышленнику. Существует разновидность троянов под названием **Trojan-Downloader**, которая, осуществляет несанкционированную загрузку на компьютер пользователя программного обеспечения (обычно зловредного).

5. **Боты** (bots) — распространенный в наше время вид зловредного ПО, который устанавливается на компьютерах пользователей и используется для атак на другие компьютеры (сети **botnet**).

6. **Снифферы** (sniffers) — это анализаторы сетевого трафика. Могут использоваться в составе зловредного ПО, скрытно устанавливаться на компьютере пользователя и отслеживать данные, которые отправляет или получает пользователь по сети.

7. **Руткиты** (rootkits) — сами по себе не являются зловредным ПО. Назначение — скрывать работу других зловредных программ (кейлоггеров, троянов, червей и т.д.) как от пользователя, так и от программ безопасности (антивирусов, файерволов, систем обнаружения атак и пр.).

8. “**Звонилка**” (Dialer или Porn-Dialer) —может просто изменять настройки уже существующих соединений удаленного доступа на компьютере пользователя или создавать новое соединение.

9. **Эксплоиты** (exploits)— это программы, которые через ошибку в программном обеспечении компьютера могут предоставить несанкционированный доступ машине или просто вывести ее из строя (завесить, перезагрузить).

10. **AdWare** (приставка "Ad" является сокращением от английского слова "advertisement" — реклама, а слово "Ware" переводится как "продукт") — это приложение, которое показывает рекламу, доставляемую через интернет.

11. SpyWare (англ. Spy — **шпион**, Ware — **продукт**) — программа-шпион, которая собирает и передает посторонним лицам информацию о пользователе без его согласия. В основном, SpyWare используется для маркетинговых исследований, поэтому собранная информация обычно передается на серверы рекламных фирм.

Методы нейтрализации:

- регулярное обновление операционной системы;

- использование безопасного браузера (Opera, Mozilla Firefox) и почтового клиента(The Bat!, Mozilla Thunderbird , Sylpheed) ;

- установка надежного файервола (Outpost Firewall Pro, ZoneAlarm Free Firewall, Gomodo Firewall);

Файервол (англ. firewall — огненная стена) — это браузер типа **программный фильтр** (существуют и программно-аппаратные файерволы), который отслеживает входящий и исходящий сетевой трафик компьютера и блокирует потенциально опасные соединения.

- установка антивируса и анти-шпионского ПО

- Частое обновление антивирусных баз

- Проверка файлов в архивах

- Проверка на вредоносное ПО с помощью антишпионов (AVG Anti-Spyware Free Edition), антируткитов (Rootkit Unhooker), антикейлоггеров (Advanced Anti Keylogger).

- отключение неиспользуемых служб

- отключение неиспользуемых служб:

- Проверка автозагрузки файлов

- Просмотр списка процессов

56 Оценка безопасности парольной защиты.

Одним из важнейших методов защиты для соблюдения конфиденциальности является разграничение доступа. Практически с момента создания первых многопользовательских операционных систем для ограничения доступа используются пароли.

В памяти компьютера пароли хранятся в виде хэша.

Для взлома парольной защиты используются следующие методы.

1. Узнавание пароля.

2. Угадывание пароля.

3. Словарная атака.

4. Метод прямого перебора (время взлома пароля длиной 8 символов, состоящего из цифр и букв английского алфавита составляет до 6 суток;)

5. Использование программных закладок.

6. Удаленный доступ к компьютеру.

7. Непосредственный доступ к компьютеру.

8. Перехват паролей с использованием технических средств.

Windows 95/98 сохраняли пароль в PWL-файле. Вместе с тем стоит отметить, что несмотря на то, что содержимое PWL-файла было зашифровано, извлечь из него пароли было довольно просто.

В операционных системах Windows XP/2000/2003 применяется более надежная защита парольного метода аутентификации. Но в то же время необходимо выполнять следующие рекомендации Microsoft:

- длина пароля должна составлять не менее восьми символов;
- в пароле должны встречаться большие и маленькие буквы, цифры и спецсимволы;
- время действия пароля должно составлять не более 42 дней;
- пароли не должны повторяться.

В настоящее время основным способом защиты информации от несанкционированного доступа (НСД) является внедрение так называемых средств AAA (Authentication, Authorization, Accounting — аутентификация, авторизация, управление правами пользователей). При использовании этой технологии пользователь получает доступ к компьютеру лишь после того, как успешно прошел процедуры идентификации и аутентификации.

Классификация средств идентификации и аутентификации



Внедрение **комбинированных** систем существенно увеличивает количество идентификационных признаков и тем самым повышает безопасность

- системы на базе бесконтактных смарт-карт и USB-ключей;
- системы на базе гибридных смарт-карт;
- биоэлектронные системы.

Эффективность использования пароля

- $A = \{a_i\}$ – алфавит, состоящий из фиксированного набора символов, $i \in [1, N]$, N – мощность алфавита
- s - длина пароля H ; при $H = '12AAa!!*' s = 8$
- Кол-во комбинаций пароля при фиксир $N : I_H = N^s$;
Пример1. $A = \{a,b,c,d,\dots,z\}$, $N=26$; при $s = 8$ $N^s = 26^8 = 208\ 827\ 064\ 576$
- Безопасное время использования пароля

$$t_H = \frac{1}{2} (I_H \cdot t), \quad (1)$$

$$t = E/R, \quad E = S + S_{sl};$$

Пример2. $N = 5$ симв, $S = 6$ симв, скорость передачи $R = 3$ [Кбит/с];
принимаем $S_{sl}=4$ симв, тогда $E = 6+4=10$ симв (либо 80 бит) и

$$t_H = \frac{1}{2} (I_H \cdot t) = 1/2(5^6 * 80/(3*1024)) = 203 \text{ с}$$

Пример3. $N = 26$ симв, $S = 6$ симв, скорость передачи $R = 32$ [Кбит/с];
принимаем $S_{sl}=14$ симв, тогда $E = 6+14=20$ симв (либо 160 бит) и

$$t_H = \frac{1}{2} (I_H \cdot t) = 1/2(26^6 * 160/(32*1024)) = 7.5 * 10^5 \text{ с} = 3.5 \text{ ч}$$

Безопасное время использования пароля

Принимаем P – это вероятность того, что пароль будет взломан за M мес,

P_0 – нижняя граница P ; $P_0 = n1/n2$; $n1$ – число попыток взлома пароля за M мес; $n2$ – число всех возможных паролей при определенных N и s ;

$n1 = n11/n12$; $n11$ – число символов, которые можно передать по сети за M мес, $n12$ – число символов, передаваемых в одной попытке;

$$n1 = (R \cdot M \cdot 24(\text{ч/д}) \cdot 60(\text{мин/ч}) \cdot \mathbf{60 \text{ (сек/мин)}} \cdot 30(\text{д/мес})) / E, \quad (2)$$
$$n2 = N^s,$$

$$\text{тогда } P_0 = (R \cdot M \cdot 24 \cdot 60 \cdot \mathbf{60} \cdot 30) / (E \cdot N^s). \quad (3)$$

Так как $P > P_0$, $P > (R \cdot M \cdot 24 \cdot 60 \cdot \mathbf{60} \cdot 30) / (E \cdot N^s)$ или иначе

$$N^s \geq (4.32 \cdot 10^4 \cdot R \cdot M) / (E \cdot P) - \text{ф-ла Андерсена} \quad (4)$$

$$N^s \geq (2.59 \cdot 10^6 \cdot R \cdot M) / (E \cdot P)$$

Пример. $P = 10^{-3}$, $M = 3$; $R = 10$ (сим/сек); $E = 20$ (сим); $N = 26$ (сим); $s = 6$ (сим);

$$(2.59 \cdot 10^6 \cdot R \cdot M) / (E \cdot P) = (2.59 \cdot 10^6 \cdot 10 \cdot 3) / (20 \cdot 10^{-3}) = 3.9 \cdot 10^9;$$

$$N^s = 26^6 \approx 3.089 \cdot 10^8 \leq 3.9 \cdot 10^9.$$

Это означает, что при выбранном размере алфавита и длине пароля, необходимое условие неравенства не выполняется.

При $s = 7$ (сим):

$$26^7 \approx 8.03 \cdot 10^9 \geq 3.9 \cdot 10^9.$$

Выполнение условия означает, что для выбранного алфавита пароль длиной 7 символов будет взломан за 3 месяца с вероятностью не более, чем $P = 10^{-3}$.