**Н. Г. Бутакова Н. В. Федоров**

**Криптографические методы и средства защиты информации**

*учебное пособие*

*для студентов по направлению*

*«Информационная безопасность»*

ООО «ИЦ «Интермедия»

Санкт-Петербург 2016

УДК 003.26

ББК 32.973.26-018я73

Б93

Главный редактор: *Т.С. Кулакова*

Техническая подготовка: *В.Ю. Антипова*

Дизайн обложки: *В.С. Кулаков*

Б93 Бутакова Н.Г., Федоров Н.В.

Криптографические методы и средства защиты информации: учебное пособие / Н.Г. Бутаков, Н.В. Федоров – СПб.: ИЦ «Интермедия», 2016. – 384 с.

В учебном пособии рассмотрены теоретические и практические аспекты криптографической защиты информации в процессе ее хранения, обработки в информационных системах и передачи по общедоступным каналам связи.

Содержание пособия охватывает основные положения, необходимые для изучения методов криптографической защиты. Подробно рассмотрены вопросы практического использования криптографических методов в условиях информационных технологий. Также нашли отражение последние достижения в области современной криптографии.

Для студентов высших учебных заведений, обучающихся по направлению 10.03.01 «**Информационная безопасность**», аспирантов и преподавателей, интересующихся проблемами защиты информации.

ISBN 978-5-4383-0135-6 УДК 003.26

ББК 32.973.26-018я73

*Учебное издание*

Н. Г. Бутакова

Н. В. Федоров

КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ И СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ

Подписано в печать 11.05.2016. Формат 60 × 88 1/16. Печать цифровая

Усл. печ. л. 22,9. Тираж 500 экз. Заказ №

ООО «Издательский центр “Интермедия”». Адрес: 198334, Санкт-Петербург, ул. Партизана Германа, 41-218. Отпечатано с готового оригинал-макета в ООО «Арт-экспресс». Адрес: 199155, СПб., В.О., ул. Уральская, д. 17.

|  |  |
| --- | --- |
| ISBN 978-5-4383-0135-6 | © ООО «Издательский Центр “Интермедия”», 2016  © Н.Г. Бутакова, Н.В. Федоров, 2016 |

# Оглавление

Введение ................................................................................................................... 7

ЧАСТЬ 1. КРИПТОГРАФИЧЕСКИЕ

МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ .............................................................. 13

Тема 1. История развития криптографии ............................................................ 14

1.1. Криптография в Античные времена .......................................................... 17

1.2. Криптография в Средневековье ................................................................. 20 1.3. Криптография в Эпоху Возрождения ........................................................ 22

1.4. Криптография в Новое и Новейшее время ............................................... 28

Вопросы и задания для самостоятельной работы ........................................... 33

Тема 2. Основные задачи современной криптографии ..................................... 34

2.1. Конфиденциальность .................................................................................. 35

2.2. Целостность .................................................................................................. 39 2.3. Аутентификация .......................................................................................... 40 2.4. Неотслеживаемость ..................................................................................... 41

2.5. Цифровая подпись ....................................................................................... 43 2.6. Управление ключами .................................................................................. 44

2.7. Общие требования к криптосистемам ....................................................... 47 Вопросы и задания для самостоятельной работы ........................................... 47

Тема 3. Математические основы криптографии. Элементы теории чисел,

абстрактной алгебры и алгебраической геометрии ........................................... 48

3.1. Простые числа и непрерывные дроби ....................................................... 49 3.2. Мультипликативные функции ................................................................... 62 3.3. Сравнение целых чисел по модулю ........................................................... 68

3.4. Решение сравнений первой степени .......................................................... 73 3.5. Группы и их свойства .................................................................................. 76

3.6. Кольца и тела в абстрактной алгебре ........................................................ 77

3.7. Конечные поля ............................................................................................. 78

3.8. Эллиптически кривые над конечными полями ........................................ 83

Вопросы и задания для самостоятельной работы ........................................... 90

Тема 4. Классификация шифров .......................................................................... 91

4.1. Классификация шифров .............................................................................. 91 4.2. Математические модели шифров .............................................................. 96 4.3. Математические модели открытого текста .............................................. 98

4.4. Критерии распознавания открытого текста .............................................. 99

Вопросы и задания для самостоятельной работы ......................................... 101

Тема 5. Шифры перестановки ............................................................................ 101

5.1. Шифр Сцитала ........................................................................................... 101

5.2. Маршрутные перестановки ...................................................................... 102

5.3. Шифры вертикальной перестановки ....................................................... 104

5.4. Поворотные решетки................................................................................. 105 5.5. Задача Эйлера ............................................................................................ 106

5.6. Элементы криптоанализа шифров перестановки ................................... 107

Вопросы и задания для самостоятельной работы ......................................... 109

Тема 6. Шифры замены....................................................................................... 110 6.1. Поточные шифры простой замены .......................................................... 110 6.2. Шифры многоалфавитной замены ........................................................... 113

6.3. Биграммные и n-граммные шифры замены ............................................ 117

6.4. Матричные шифры .................................................................................... 119 6.5. Модель шифров замены ............................................................................ 121 6.6. Классификация шифров замены .............................................................. 122

6.7. Криптоанализ поточного шифра простой замены ................................. 124

Вопросы и задания для самостоятельной работы ......................................... 126

Тема 7. Шифры гаммирования ........................................................................... 126

7.1. Основные требования к гамме ................................................................. 127 7.2. Шифр Виженера ........................................................................................ 128

7.3. Одноразовый блокнот К. Шеннона ......................................................... 130

7.4. Аддитивные методы шифрования ........................................................... 131

7.5. Режим гаммирования ГОСТ 28147-89 .................................................... 132 Вопросы и задания для самостоятельной работы ......................................... 134

Тема 8. Поточные системы шифрования .......................................................... 135

8.1. Регистры сдвига с обратной связью ........................................................ 135

8.2. Скремблеры ................................................................................................ 137

8.3. Методы рандомизации сообщений .......................................................... 143

8.4. Поточные шифрсистемы ........................................................................... 145

Вопросы и задания для самостоятельной работы ......................................... 147

Тема 9. Блочные системы шифрования ............................................................ 148

9.1. Блочные системы шифрования ................................................................ 148 9.2. Конструкция Фейстеля ............................................................................. 149

9.3. Режимы шифрования блочных шифров .................................................. 151

9.4. Алгоритмы блочного шифрования .......................................................... 154

Вопросы и задания для самостоятельной работы ......................................... 162

Тема 10. Системы шифрования с открытыми ключами .................................. 163

10.1. Асимметричные системы........................................................................ 164

10.2. Открытое распределение ключей. Схема Диффи–Хеллмана ............. 166

10.3. Криптосистема RSA ................................................................................ 168 10.4. Схема шифрования Эль-Гамаля ............................................................. 170

10.5. Криптография на эллиптических кривых ............................................. 171

10.6. Криптоанализ шифра RSA ...................................................................... 173

Вопросы и задания для самостоятельной работы ......................................... 175

Тема 11. Электронные цифровые подписи ....................................................... 175 11.1. Механизм действия электронной цифровой подписи ......................... 176

11.2. Функции хэширования ............................................................................ 178 11.3. Алгоритмы цифровой подписи .............................................................. 180

11.4. Криптоанализ односторонних хэш-функций ....................................... 197

Вопросы и задания для самостоятельной работы ......................................... 198 Тема 12. Квантовая криптография ..................................................................... 199

12.1. Природа секретности квантового канала связи .................................... 199

12.2. Основные направления развития квантовой криптографии ............... 200 12.3. Протоколы квантового обмена информацией ...................................... 201 12.4. Квантовый криптоанализ ........................................................................ 208

12.5. Проблемы практической реализации систем квантовой

криптографии ................................................................................................... 209

Вопросы и задания для самостоятельной работы ......................................... 211

ЧАСТЬ 2. СРЕДСТВА КРИПТОГРАФИЧЕСКОЙ

ЗАЩИТЫ ИНФОРМАЦИИ ............................................................................... 212

Тема 1. Нормативно-правовые основы криптографической защиты

информации.......................................................................................................... 213

1.1. Действующие стандарты криптографической защиты информации ... 213

1.2. Федеральный закон об электронной подписи ........................................ 224

1.3. Нормативно-правовые акты ФСБ по обеспечению безопасности

персональных данных с использованием СКЗИ ........................................... 224

1.4. Нормативно-правовые акты ФСБ по обеспечению

функционирования и эксплуатации СКЗИ .................................................... 228

1.5. Виды работ и услуг, составляющих лицензируемую деятельность

с использованием СКЗИ .................................................................................. 229

1.6. Порядок лицензирования деятельности с использованием

СКЗИ .................................................................................................................. 229

1.7. Правила сертификации СКЗИ по требованиям ФСБ ............................. 233

1.8. Таможенные ограничения на ввоз и вывоз СКЗИ.................................. 234

Вопросы и задания для самостоятельной работы .................................... 235 Тема 2. Классификация средств криптографической защиты

информации.......................................................................................................... 237

2.1. Классификация средств криптографической защиты информации

по различным признакам ................................................................................. 237

2.2. Требования к средствам криптографической защиты информации .... 240

2.3. Программные СКЗИ. Особенности и примеры ...................................... 241 2.4. Аппаратные и программно-аппартные СКЗИ.

Особенности и примеры .................................................................................. 245 2.5. Критерии выбора СКЗИ ............................................................................ 248

2.6. Основные принципы построения СКЗИ ................................................. 249

2.7. Принципы построения аппаратных СКЗИ .............................................. 250

2.8. Принципы построения программных и программно-аппаратных

СКЗИ .................................................................................................................. 252

2.9. Основные подходы к обеспечению надежности СКЗИ ......................... 253 Вопросы и задания для самостоятельной работы ......................................... 256

Тема 3. Средства криптографической защиты информации на

персональном компьютере ................................................................................. 258

3.1. Задачи обеспечения информационной безопасности на

персональном компьютере с использованием СКЗИ ................................... 258

3.2. Криптографическая защита жестких дисков и съемных носителей .... 259

3.3. Средства шифрования, встроенные в операционную систему

Windows ............................................................................................................. 268

3.4. Шифрование архиваторов ........................................................................ 281 3.5. СКЗИ свободного доступа ........................................................................ 283 3.6. СКЗИ от несанкционированного доступа ............................................... 286

3.6.1. Программные СКЗИ ............................................................................ 286

3.6.2. Аппаратные СКЗИ ............................................................................... 292

3.6.3 Жесткие диски с технологией самошифрования .............................. 295

3.6.4. Практические рекомендации .............................................................. 299

Вопросы и задания для самостоятельной работы ......................................... 301 Тема 4. Средства криптографической защиты сетевого взаимодействия ..... 303

4.1. Криптографические средства создания защищенных виртуальных

сетей (VPN) ....................................................................................................... 303

4.2. Технология построения криптозащищенных туннелей ........................ 304

4.3. Криптографическая защита удаленного доступа к локальной сети .... 305

4.4. СКЗИ для передачи данных в локальных сетях ..................................... 309 4.5. Сетевые протоколы криптографической защиты .................................. 312

4.6. Персональные криптографические средства аутентификации ............ 321

Вопросы и задания для самостоятельной работы ......................................... 325

Приложение 1. Пример зашифрования текста различными шифрами .......... 326

Приложение 2. Надежность криптосистем ....................................................... 335

Приложение 3. Шифровальные машины .......................................................... 344 Приложение 4. Стандарты шифрования ........................................................... 353

Литература ........................................................................................................... 368

# ВВЕДЕНИЕ

*Криптография* – область научных, прикладных, инженерно-технических исследований и практической деятельности, направленных на разработку средств криптографической защиты информации от несанкционированных пользователей.

Слово «*криптография*» образовано из двух греческих слов: *kryptos* – тайный, скрытый и *graphό –* пишу. Способ тайного письма, понятного лишь посвященным, тайнопись. Термин ввел Джон Валлис – английский математик, один из основателей и первых членов Лондонского королевского общества, изучал криптографию и применял свои знания к расшифровке различного рода политической переписки.



**John Wallis** (1616–1703), английский математик

Защищаемая информация в результате криптографического преобразования становится недоступной для ознакомления и непосредственного использования лицами, не имеющими на это соответствующих полномочий. Исследованием и разработкой методов и способов преобразования информации с целью ее защиты и занимается *криптографи*я – прикладная наука, основанная на самых последних достижениях фундаментальных наук и, в первую очередь, математики и ее разделов – Теории чисел, Абстрактной алгебры и Алгебраической геометрии.

Криптографические преобразования обычно связаны с шифрованием и расшифрованием информации. При шифровании с помощью определенных правил, содержащихся в шифре, осуществляется преобразование защищаемой информации (открытого текста) к неявному виду, т.е. в шифрованное сообщение (шифртекст, криптограмму). Расшифрование – процесс, обратный шифрованию, т.е. преобразование шифрованного сообщения в подлежащую защите исходную информацию.

В течение многих лет криптография была засекречена. Агентство национальной безопасности в США, соответствующие ведомства в Англии, России и в других развитых странах применяли криптографию для защиты государственных и военных секретов, для обеспечения безопасности военной и дипло-

матической связи, а также для целей разведывательной и контрразведывательной спецслужб.

На рубеже XX и XXI веков положение изменилось. Этому способствовали развитие информационных технологий и чрезвычайно интенсивные научные исследования в области криптографии, которые привели к тому, что современная компьютерная криптография распространилась широко за стены строго охраняемых военных учреждений. Криптографическая защита стала применяться во всех сферах человеческой деятельности при хранении, использовании и передаче информации по линиям связи информационно-телекоммуникационных сетей. Элементы криптографии обсуждаются на страницах общедоступных журналов, монографий и учебников. Теперь при желании любой человек может зашифровать свои данные таким образом, чтобы защитить их и при хранении, и при передаче по общедоступным каналам связи.

Существуют различные методы защиты информации. Можно, например, физически ограничить доступ к информации путем хранения ее в надежном сейфе или строго охраняемом помещении. При хранении информации такой метод удобен, однако при ее передаче приходится использовать другие средства.

Можно воспользоваться одним из известных методов сокрытия информации, позволяющим скрыть не только смысл хранящейся или передаваемой информации, но и сам факт хранения или передачи закрытой информации (наука, изучающая такие методы, называется *стеганографией*):

* скрыть канал передачи информации, используя нестандартный способ передачи сообщений;
* замаскировать канал передачи закрытой информации в открытом канале связи, например, спрятав информацию в безобидном «контейнере» с использованием тех или других стеганографических способов либо обмениваясь открытыми сообщениями, смысл которых согласован заранее;
* существенно затруднить возможность перехвата противником передаваемых сообщений, используя специальные методы передачи по широкополосным каналам, сигнала под уровнем шумов, либо с использованием «прыгающих» несущих частот и т.п.

В отличие от перечисленных методов стеганографии криптография не «прячет» передаваемые сообщения, а преобразует их в форму, недоступную для понимания противником. При этом обычно исходят из предположения о полном контроле противником канала связи. Это означает, что противник может не только пассивно перехватывать передаваемые сообщения для последующего их анализа, но и активно изменять их, а также отправлять поддельные сообщения от имени одного из абонентов.

Помимо сокрытия существуют и другие проблемы защиты передаваемой информации. Например, при полностью открытом информационном обмене возникает проблема достоверности полученной информации. Для ее решения необходимо обеспечить:

* проверку и подтверждение подлинности содержания и источника сообщения;
* предотвращение и обнаружение обмана и других умышленных нарушений со стороны самих участников информационного обмена.

Криптографические методы защиты информации являются в настоящее время наиболее надежным, а в некоторых случаях – единственным средством противодействия различным угрозам. Они применяются как для защиты информации, обрабатываемой в ЭВМ, так и для закрытия информации, передаваемой по каналам связи между различными элементами информационных систем.

В зарубежной литературе общепринято определять область защиты информации собирательным термином «*криптология*» (от греческих корней *kryptos –* «скрытно» и *logos –* «слово, понятие, учение»). Термин впервые появился в книге англичанина Джона Уилкинса, епископа Честера, основателя и секретаря королевского научного общества «Меркурий, или секретный и быстрый курьер».



**Джон Уилкинс** (1614–1672), британский священник

Криптология вполне естественным образом делится на два направления: криптографию и криптоанализ. Криптограф изыскивает методы гарантированного засекречивания и/или подтверждения подлинности (аутентичности) сообщения. Оригинал сообщения, который криптограф пытается засекретить, называется *открытым текстом*, а плод его труда – *шифртекстом*, или *криптограммой*. Криптограф всегда использует секретный ключ для управления процессом шифрования.

*Криптоанализ* – наука о методах и способах вскрытия шифров. Криптоаналитик ищет способы вскрыть шифр или так подделать код сигнала, чтобы его принимали как подлинный.

Соотношение криптографии и криптоанализа в криптологии очевидно: криптография – это защита, т.е. разработка шифров, а криптоанализ – это атака, нападение на шифры. Однако эти две дисциплины связаны друг с другом, и не бывает хороших криптографов, не владеющих методами криптоанализа.

Почти универсальным предположением в криптографии является допущение, что криптоаналитик противника имеет доступ к криптограмме. Практически столь же единодушно криптографы исходят из предположения, впервые сформулированного голландцем А. Керкгоффсом (Auguste Kerckhoffs, 1835–

1903) и гласящего, что стойкость шифра должна целиком зависеть от секретного ключа. Данное допущение можно выразить таким образом: весь механизм шифрования, кроме значения секретного ключа, известен криптоаналитику противника. Если криптограф исходит только из этих двух допущений, то он разрабатывает криптосистему, стойкую к *атаке по шифртексту*. Если криптограф предполагает, что криптоаналитик сумеет заполучить пару открытого и шифрованного текстов, зашифрованных с помощью неизвестного ему секретного ключа, то шифрсистема разрабатывается стойкой *к атаке с известным открытым текстом*. Криптограф может даже предположить, что криптоаналитик противника способен составить собственный открытый текст и прогнать его через шифрсистему, получив взамен криптограмму на действующем секретном ключе (*атака с подобранным открытым текстом*). Иногда предполагают, что криптоаналитик противника может ввести свою входную информацию в шифрсистему и получить взамен выходной набор символов, «расшифрованных» действующим ключом (*атака с подобранным шифртекстом*). Создатели большинства используемых сегодня шифрсистем разрабатывают их для противостояния как минимум атаке с подобранным открытым текстом (при этом предполагается, что криптоаналитик противника никогда не сможет построить атаку более изощренную, чем атака по шифртексту).

Существует важное различие между терминами *расшифрование* и *дешифрование* шифртекста. Расшифрование – это легитимный процесс преобразования криптограммы в открытый текст с помощью секретного ключа. Дешифрование – это взлом зашифрованного текста нелегитимным участником информационного обмена без знания ключа.

Термин *дешифрование криптосистемы* имеет более широкое значение. Криптографическая система (криптосистема) – это набор преобразований из незашифрованного текста в шифртекст и наоборот. Конкретное преобразование, которое должно использоваться, выбирается ключами. Дешифрованием *криптосистемы* называется результат работы криптоаналитика, приводящий к возможности эффективного раскрытия любого зашифрованного с помощью данной криптосистемы открытого текста. Степень неспособности криптосистемы к раскрытию называется *криптостойкостью*.

В соответствии с общепринятой сегодня терминологией криптосистемы подразделяются на криптосистемы ограниченного использования и криптосистемы общего использования; симметричные, или с секретным ключом, и асимметричные, или с открытым ключом.

Криптографическая система является *криптосистемой ограниченного использования*, если ее стойкость основывается на сохранении в секрете самого характера алгоритмов шифрования и расшифрования. Простейшим историческим примером такой системы можно считать так называемый шифр Цезаря, который является шифром простой замены каждого символа открытого текста третьим следующим за ним символом алфавита с циклическим переносом, когда это необходимо. Например, известное выражение «Veni, vidi, vici» превращается в «Yhgl, ylgl, ylfl».

Криптографическая система называется *криптосистемой общего использования*, если ее стойкость основывается не на секретности алгоритмов шифрования и расшифрования, а на секретности некоторого сравнительно короткого значения, которое называется ее ключом.

Одна из проблем организации секретной связи – проблема передачи ключей между абонентами. В процессе решения этой проблемы было сделано наблюдение, которое привело к появлению *криптографии с открытым ключом*. Оно заключалось в следующем: тот, кто зашифровывает сообщение, не обязан иметь возможность его расшифровывать. В таких системах каждый пользователь Y выбирает свой собственный секретный ключ, на основании которого получает пару алгоритмов. Затем он делает один из них доступным каждому из возможных абонентов, объявляя этот алгоритм своим открытым алгоритмом шифрования, в то время как другой, соответствующий первому и являющийся его личным алгоритмом расшифрования, хранит в строгом секрете. Это позволяет даже совершенно незнакомому пользователю, например абоненту сети по имени Х, применять его общедоступный алгоритм шифрования, чтобы зашифровать предназначенное для Y сообщение; однако лишь сам Y сможет расшифровать его после получения с помощью своего секретного алгоритма расшифрования. Такие системы могут быть стойкими только при условии, что свойства общедоступного алгоритма шифрования делают невозможным «вычисление» или подбор соответствующего ему алгоритма расшифрования.

Шафи Гольдвассер (Shafi Goldwasser) и Сильвио Микэли (Silvio Micali) ввели понятие *вероятностного шифрования*. Для такой схемы шифрования открытый текст и ключ не определяют шифртекст однозначно, т.к. в процессе преобразований используется некоторый случайный параметр – *рандомизатор*. Подобные криптографические системы называются вероятностными, поскольку шифрование в них сообщений, имеющих один и тот же исходный текст и шифрующихся с использованием одного и того же ключа, может в разное время привести к совершенно различным шифртекстам. При криптоанализе этих шифртекстов трудно получить о сообщении любую информацию, позволяющую восстановить весь его открытый текст.

Предлагались и другие подходы к проблеме передачи ключей. Например, бесключевая криптография Альперна (Alpern) и Шнейдера (Schneider) может эффективно использоваться в сети связи, которая скрывает происхождение (но не содержание) сообщений. В идентификационной криптосистеме Шамира (Shamir) отпадает необходимость в распределении ключей, но требуется наличие некоего центра, которому должна быть доверена генерация секретных ключей.

Сегодня уже созданы алгоритмы и протоколы для надежной защиты информации. Правда, все не так просто – практическая реализация данных протоколов требует значительных усилий и опыта. Те области обеспечения безопасности, в которых участвуют люди, – управление ключами, интерфейс человек/компьютер, контроль доступа – часто не поддаются анализу. А такие области, как инфраструктура открытых ключей, защита программного обеспечения, компьютерная и сетевая безопасность, защита аппаратуры от проникновения, находятся в процессе изучения.

Процесс криптографического закрытия данных может осуществляться как *программно*, так и *аппаратно*. Аппаратная реализация отличается существенно большей стоимостью, однако ей присущи и преимущества: высокая производительность, простота, защищенность и т.д. Программная реализация более практична, допускает известную гибкость в использовании.

Использование современных программных и аппаратных средств криптографии невозможны без изучения теоретических основ криптографии, которые систематически и доступно изложены в данном пособии.

Предлагаемое пособие представляет собой переработку курсов, которые читались авторами в нескольких вариантах в Московском государственном мАшиностроительном университете и Московском национальном исследовательском университете «МИЭТ» для бакалавров, магистров, специалистов и аспирантов, обучающихся по направлению подготовки 10.03.01 «Информационная безопасность».

# ЧАСТЬ 1

**КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ**

**ИНФОРМАЦИИ**

## ТЕМА 1. ИСТОРИЯ РАЗВИТИЯ КРИПТОГРАФИИ

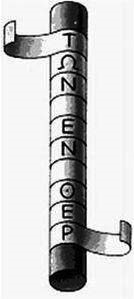
История криптографии берет свое начало в глубокой древности. Так, при раскопках в Месопотамии был найден один из самых древних шифртекстов, относящийся к XX в. до н.э. Он был составлен клинописью на глиняной табличке и содержал рецепт глазури для покрытия гончарных изделий, что, повидимому, было в то время коммерческой тайной. Известны также зашифрованные древнеегипетские религиозные тексты и медицинские рецепты. Даже появившаяся в дальнейшем письменность первоначально рассматривалась как способ закрытия информации, так как владение письменностью было уделом лишь избранных.

Долгое время криптография рассматривалась как искусство, которым владели лишь отдельные умельцы. Среди них были одаренные ученые, дипломаты, полководцы, священнослужители. История криптографии связана с различными дипломатическими и военными тайнами и поэтому окутана туманом легенд. Известны даже случаи, когда криптография считалась черной магией.

Этот период развития криптографии длился с незапамятных времен вплоть до начала ХХ века. Заметный след в этой истории оставили многие известные личности. Приведем лишь несколько примеров.

Первые сведения об использовании шифров в военном деле связаны с именем спартанского полководца Лисандра (шифр «Сцитала»). Этот шифр известен со времен войны Спарты против Афин в V веке до н.э. Для его реализации использовался сцитала-жезл, имеющий форму цилиндра.

Греческий историк Плутарх так описывает способ шифровки:

«*Отправляя к месту службы начальника флота или сухопутного войска, эфоры вручают отъезжающему круглую палку. Другую, совершенно одинаковой длины и толщины, оставляют себе. Эти палки и называют сциталами. Когда эфорам нужно сообщить какую-нибудь важную тайну, они вырезают длинную и узкую, вроде ремня, полосу папируса, плотно, без промежутков наматывают ее на свою сциталу и пишут на нем текст. Затем снимают полосу и без палки отправляют ее военачальнику. Так как буквы на ней стоят без всякой связи, разбросаны в беспорядке, прочитать написанное он может, только взяв свою сциталу и намотав на нее вырезанную полосу, чтобы, водя глазами вокруг палки и переходя от предыдущего к последующему, иметь перед собой связное сообщение».*

Отметим, что в этом шифре преобразование открытого текста в шифрованный заключается в определенной *перестановке* букв открытого текста. Поэтому класс шифров, к которым относится и шифр «Сцитала», называется *шифрами перестановки*.

В 50-х гг. до н.э. во время войны с галлами Ю. Цезарь использует другую разновидность шифра. Под алфавитом открытого текста записывался тот же алфавит, циклически сдвинутый на определенное число позиций. У Ю. Цезаря такой сдвиг осуществлялся на три позиции. Главное, чтобы тот, кому посылается шифрованное сообщение, знал эту величину сдвига. Таким образом, при шифровании буквы открытого текста *заменяются* буквами сдвинутого алфавита. Хотя подобный шифр был известен до Ю. Цезаря, тем не менее, он был назван его именем.

Класс шифров, к которым относится и шифр Цезаря, называется *шифрами замены*. Более сложным шифром замены является греческий шифр – «квадрат Полибия». В этом шифре алфавит записывается в виде квадратной таблицы. При шифровании открытого текста его буквы заменяются парой чисел, соответствующих номерам столбцов и строк данной буквы в таблице. Идея квадрата Полибия была реализована в более сложных шифрах, применявшихся во время Первой мировой войны.

Некоторые сведения о применении шифров можно найти в художественной литературе: в приключенческой, детективной и военной. Подробное объяснение особенностей одного из простейших шифров замены содержится в двух известных рассказах: «Золотой жук» Э. По и «Пляшущие человечки» А. Конан Дойла.

Крах Римской империи в V веке н.э. сопровождался, как известно, кризисными процессами в области искусства и в сфере научной деятельности. Это отразилось и на криптографии. Церковь в те времена преследовала тайнопись, которую она считала чернокнижием и колдовством. Сокрытие мыслей за шифрами не позволяло церкви контролировать эти мысли.

Эпоха средневекового Возрождения знаменует очередной этап в развитии криптографии, отмеченный новыми именами и оригинальными идеями.

Одну из первых книг по криптографии написал живший в Германии аббат Иоганнес Тритемий (1462–1516). Им был предложен оригинальный шифр многозначной замены под названием «Ave Maria», в котором каждая буква открытого текста имела не одну замену, а несколько, по выбору шифровальщика.

В 1550 году итальянский математик, философ, врач Джероламо Кардано (1506–1576) опубликовал работу с описанием изобретенной им системы шифрования («решетка Кардано»), на основе которой, например, был создан один из военно-морских шифров Великобритании во время Второй мировой войны. «Решетка Кардано» является примером шифра перестановки.

В XVI в. в работах французского дипломата Блеза де Виженера получили также дальнейшее развитие шифры замены. Система шифрования Виженера в том или ином виде используется до настоящего времени.

Франция XVI века оставила в истории криптографии шифры короля Генриха IV и кардинала Ришелье, причем последний создал первую в мире шифрослужбу.

В XVII веке были изобретены так называемые словарные шифры. При шифровании открытого текста его буквы обозначались двумя числами – номером строки и номером буквы в строке на определенной странице какой-нибудь выбранной книги.

Среди множества российских шифров можно отметить «цифирную азбуку» 1700 года, автором которой был Петр Великий.

Впечатляющий путь, пройденный криптографией в процессе многовековой эволюции, окончательно сформировал ее как науку только в середине ХХ века. Именно в эти годы вышли в свет фундаментальные работы американского математика К. Шеннона по основам теории информации, в которых был разработан математический аппарат для решения криптографических задач. В работе К. Шеннона «Теория связи в секретных системах» криптография была поставлена на научную основу.

Эволюция криптографической деятельности в различных странах обычно не является прямолинейной. Как правило, здесь имеются взлеты и падения, вызванные конкретной исторической обстановкой. Нередко оригинальные методы защиты информации забывались и изобретались заново через столетия. Иногда плодотворные идеи возникали параллельно в различных странах. Криптография в историческом аспекте развивалась не сама по себе. Внимание, уделяемое развитию криптографии, зависело от активности деятельности государства в различных сферах: политической, военной, экономической и так далее. Криптография выполняла заказы государства и развивалась при его соответствующей поддержке. Среди внутренних противоречий, стимулирующих развитие криптографии, в первую очередь следует выделить противостояние двух сторон кpиптoгрaфичecкой деятельности: защиты и нападения. Успехи в дешифровании шифров приводили к разработке новых шифров; в свою очередь, разработка новых шифров – к поиску новых методов их дешифрования.

Огромное влияние на развитие криптографии во всей истории ее существования оказывали достижения научнo-тeхничecкого прогресса.

Криптография (в современном понимании этого слова) появилась практически сразу же после появления письменности. Мощный импульс ее развитию дало изобретение алфавитной письменности. Широкому распространению криптографических способов защиты информации способствовало совершенствование тeхнoлoгичecкой базы обмена письменными сообщениями: от записи на камнях – к глиняным табличкам (Месопотамия), затем к папирусу (Египет), бересте (Россия), шелковой ткани (Китай), пергаменту (Египет, Греция, Рим), деревянным дощечкам (Греция, Рим), к бумаге (Китай, I в. н.э.) и, наконец, к современным носителям информации.

Во все времена учитывались затраты на защиту информации и на реализацию методов нападения. Как защита, так и нападение требуют сопоставления затрат с возможными доходами от успехов в их воплощении.

Наряду с развитием криптографии как искусства и науки развивались и совершенствовались государственные криптографические структуры. Если у истоков криптографической деятельности стояли специалисты-одиночки, то к настоящему времени появились такие мощные государственные организации, как Служба специальной связи и информации Федеральной службы охраны Российской Федерации, Агентство Национальной Безопасности в США.

### 1.1. Криптография в Античные времена

Криптография по возрасту – ровесник египетских пирамид. В документах древних цивилизаций – Индии, Египта, Месопотамии – есть сведения о системах и способах составления шифрованных писем.

В древнеиндийских рукописях описаны 64 способа письма.

В религиозных книгах Индии указывается, что Будда знал несколько десятков способов письма, среди которых (по современной классификации) присутствовал шифр перестановки. В знаменитом учебнике Ватсьяны об искусстве любви («Камасутра»), среди 64 искусств, которыми должна овладеть женщина, на 45 месте упоминается тайнопись в виде шифра простой замены.

Один из самых старых шифрованных текстов из Месопотамии представляет собой глиняную табличку, написанную клинописью и содержащую рецепт для изготовления глазури в гончарном производстве (ХХ в. до н.э.). Для его написания были использованы редко употребляемые клинописные знаки, игнорировались некоторые гласные и согласные и употреблялись числа вместо имен.

Шифровальные тексты древнего Египта – это чаще всего религиозные тексты и медицинские рецепты.

Древние египтяне использовали и символический язык. Так, в 1998 году был дешифрован текст, записанный на каменных плитах. Этому тексту более 6000 лет, и он получил название Великие Арканы Таро. В нем в символической форме трактуются принципы мироздания, говорится об абсолютной и относительной истине.

В Древнем Китае изобретались изощренные способы физической защиты посланий. Сообщения запрятывались в тайных уголках тела.

Европейская криптография появилась в Греции. Мощным толчком к ее развитию послужил тот факт, что в Греции впервые окончательно сформировалось европейское алфавитное письмо (VIII в. до н.э.). Греческий алфавит в дальнейшем породил латинский и русский алфавиты. До этого момента были распространены письменности, слабо пригодные для применения шифров.

В криптографии древних времен использовалась два вида шифров: замена и перестановка.

Историческим примером шифра замены является *шифр Цезаря*.



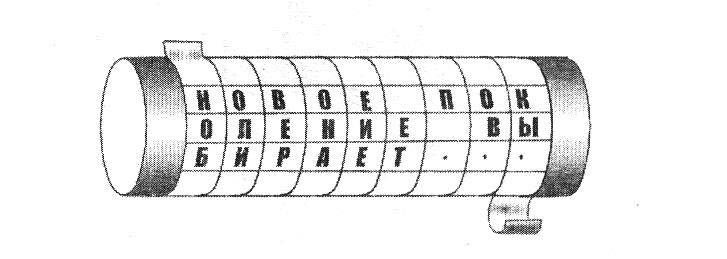
***Рис. 1.1.*** *Цезарь Гай Юлий (100–44 гг. до н.э.), полководец, римский император в 46–44 гг. до н.э.*

Он использовал в своей переписке шифр собственного изобретения. Выписывался алфавит, затем под ним тот же алфавит, но с циклическим сдвигом на 3 буквы влево.

И при зашифровке текста первая буква алфавита становилась четвертой, и т.д. со сдвигом на 3 буквы. Преемник Юлия Цезаря – Август использовал тот же шифр, но с другим ключом – сдвиг на 1 букву.

С тайнописью был знаком древнегреческий философ Платон, которого способам тайнопись обучили при посвящении в Мистерию Великого Ордена в Египте.

Одним из первых физических приборов, реализующих шифр перестановки, был прибор Сцитала-цилиндр обусловленного диаметра. В качестве такого прибора могли использоваться жезлы, рукоятки мечей, копий, кинжалов и т.п. Изобретение дешифровального устройства приписывается Аристотелю. Он предложил использовать конусообразное копье, на которое наматывается перехваченный ремень. Этот ремень передвигается по оси до того положения, пока не появляется открытый текст.



***Рис. 1.2.*** *Модель Сциталлы с текстом «НОВОЕ ПОКОЛЕНИЕ ВЫБИРАЕТ»*

В области тайнописи два изобретения принадлежат Энею. Первое из них – *диск Энея*. Его принцип был прост. На диске диаметром 10–15 см и толщиной 1–2 см высверливались отверстия по числу букв алфавита. В центре диска помещалась катушка с намотанной на ней ниткой достаточной длины. При зашифровании нитка вытягивалась с катушки и последовательно протягивалась через отверстия в соответствии с буквами шифруемого текста. Диск и являлся посланием. Получатель послания последовательно вытягивал нитку из отверстий, что позволяло ему получать передаваемое сообщение, но в обратном порядке следования букв. При перехвате диска недоброжелатель имел возможность прочитать сообщение тем же образом, что и получатель. Но Эней предусмотрел возможность легкого уничтожения сообщения при угрозе захвата диска. Для этого было достаточно выдернуть катушку с закрепленным на ней концом нити до полного выхода всей нити из отверстий диска.

Другим шифровальным приспособлением времен Спарты была *табличка Энея* (Эней Тактик – полководец Спарты IV века до н.э.). На небольшой табличке (рис. 1.3) горизонтально располагался алфавит, а по ее боковым сторонам имелись выемки для наматывания нити. При зашифровании нить закреплялась у одной из сторон таблички и наматывалась на нее. На нити делались отметки (например, узелки) в местах, которые находились напротив букв данного текста. По алфавиту можно было двигаться лишь в одну сторону, то есть делать по одной отметке на каждом витке. После зашифрования нить сматывалась и передавалась адресату. Этот шифр представляет собой шифр замены букв открытого текста знаками, которые означали расстояния между отметками на нити. Ключом являлись геометрические размеры таблички и порядок расположения букв алфавита. Это был довольно надежный шифр, история не сохранила документов, подтверждающих сведения о методах его вскрытия. Интересно, что индейцы Центральной Америки передавали свои сообщения также в виде нитки, на которой завязывались разноцветные узелки, определявшие содержание сообщения.

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

***Рис. 1.3.*** *Табличка Энея*

Еще одно изобретение древних греков – так называемый *квадрат Полибия* (Полибий – греческий государственный деятель, историк, живший в 200–120 гг. до н.э.). Применительно к современному латинскому языку из 26 букв шифрование по этому квадрату заключалось в следующем. В квадрат размером 5 на 5 клеток выписываются все буквы алфавита, при этом буквы I, J учитываются за одну.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | A | B | C | D | E |
| 2 | F | G | H | I,J | K |
| 3 | L | M | N | O | P |
| 4 | Q | R | S | T | U |
| 5 | V | W | X | Y | Z |

***Рис. 1.4.*** *Квадрат Полибия применительно к латинскому алфавиту*

Шифруемая буква заменялась на координаты квадрата, в котором она записана: парами чисел (i,j), где i – номер строки, j – номер столбца. Пары (i,j) передавались с помощью факелов. Например, для передачи буквы О нужно было взять 3 факела в правую руку и 4 факела – в левую.

Усложненный вариант квадрата Полибия заключается в записи букв в квадрат в произвольном порядке. Этот порядок является ключом.

Аналогом квадрата Полибия можно считать тюремную азбуку, которая позволяла путем перестукивания сообщаться заключенным разных камер. Количество ударов по стене соответствовало номерам строки и столбца каждой буквы.

Иногда с целью защиты информации использовались редкие, экзотические языки, например языки древней Азии, Африки и так далее. Эта практика дошла до наших дней. Так, во время Второй мировой войны американцы использовали язык индейцев племени навахо для передачи информации по незащищенным каналам связи, а в 1960 году ирландские войска в Конго, направленные туда по решению ООН, вели радиопереговоры на гэльском языке.

С древних времен широкое распространение получили так называемые *акроаматические шифры*. Их содержание заключается в использовании притч и аллегорий для передачи истинного смысла. Эти шифры встречаются в священных писаниях Платона и Аристотеля, баснях Эзопа и так далее. Расшифрование такого шифра связано с истолкованием текстов, что порождает различные варианты предполагаемого авторского замысла.

### 1.2. Криптография в Средневековье

После гибели Римской империи Европа погрузилась во мрак средневекового невежества. Почти исчезли грамотность, наука и искусство. Не стала исключением и криптография, хотя шифрование как таковое не было забыто. В VI веке верующий человек нацарапал на стене коптского монастыря в Египте зашифрованную шифром простой замены просьбу: «Во имя Бога, я, Виктор, бедный человек, прошу – помните меня!»

Шифрами забавлялись многие, включая коронованных особ. Эти шифры по содержанию оставались на уровне Древней Греции и Рима. Они использовались не для сохранения тайны, а для придания обыденным фразам удивительной формы.

Применявшиеся в Средневековье шифры были предельно просты. Фразы писались либо по вертикали, либо в обратном порядке, гласные заменялись точками, использовались иностранные алфавиты (греческий, древнееврейский, армянский). В шифрах замены использовался упрощенный шифр Цезаря, где каждая буква заменялась на последующую, а в более сложных системах буквы заменялись на специально придуманные знаки.

Из времени Средневековья в истории сохранились лишь несколько имен, связанных с европейской криптографией.

Миссионер Св. Бонифаций (Германия) в XIII веке привез в Западную Европу «Сборник криптозагадок», где гласные буквы заменялись точками.

Английский монах-францисканец, профессор в Оксфорде, универсальный ученый, математик, оптик, астроном Роджер Бэкон в середине VIII века написал книгу «Тайные опыты и недействительность магии». В предисловии он заметил: «Дурак тот, кто пишет о тайне каким-либо способом, но не так, чтобы скрыть ее от простонародья». Здесь же Бэкон приводит несколько методов сокрытия тайны: пропуск гласных букв, использование метафор, букв из иностранных алфавитов, стенография.

За свои научные работы Роджер Бэкон был осужден церковным судом и провел 14 лет в заточении. Этим примером объясняет широкое распространение тайнописи среди ученых Средневековья.

Одним из известных европейских криптографов был английский писатель, астроном-любитель, таможенный чиновник Джеффри Чосер. В 70-х годах XIII века в звании эсквайра Чосер, как человек надежный и сведущий, выполнял секретные дипломатические поручения своего короля в Италии и Франции. Всю тайную переписку он вел, используя шифр простой замены. В свои стихи он включал зашифрованные строфы.

Также шифры появлялись и в регионах, которые в те времена не относились к развитым. Сектанты в Ираке пользовались таинственными знаками из-за страха перед мусульманами. Аналогичная картина наблюдается у тибетцев, у членов тайных сообществ в Нигерии и Таиланде. Эти шифры представляли собой простую замену знаков открытого текста на другие, в том числе и специально придуманные. Аналогичные шифры появлялись и в северных странах (Британия, Скандинавия). Они обычно использовались в религиозных целях. В этом смысле представляют интерес так называемые рунические надписи, написанные с использованием не господствующего в то время в Европе латинского языка, а древненационального (рунического) алфавита. Использованные шифры интересны тем, что в них нередко буквы заменялись на экзотически написанные координаты их положения в специальном расположении букв алфавита – некоторое подобие шифра Полибия.

В арабском мире в рассматриваемое время была создана одна из самых развитых цивилизаций. Процветала наука, медицина и математика. Естественно, что и наилучшие условия для развития криптографии появились именно там. В арабских книгах впервые были описаны содержательные методы криптоанализа. Одна из первых крупных книг, в которой содержательно описывается криптография, – это труд, созданный в 855 году Абу Вакр Ахмед Бен Али Бен Вахшия ан-Набати – «Книга о большом стремлении человека разгадать загадки древней письменности». В ней описано несколько систем шифров арабского происхолждения и само слово *шифр*.

В 1412 г. Шехаб аль-Кашканди написал 14-томную энциклопедию «Шауба аль-Аша». В этой работе имеется раздел о криптографии «Относительно сокрытия в буквах тайных сообщений». В нем дано систематическое описание различных шифров замены и перестановки. Здесь же приводится и криптоанализ этих шифров на основе подсчета частотности букв в арабском языке по тексту Корана; развиваются методы дешифрования, базирующиеся на разночастотности букв. В этой же книге Кашканди пишет о частотном методе дешифрования шифра простой замены: «Если вы хотите разгадать сообщение, которое вы получили в зашифрованном виде, то прежде всего начинайте подсчет букв, а затем сосчитайте, сколько раз повторяется каждый знак, и подведите итог в каждом отдельном случае…» Такой метод в современной криптографии называется *методом частотного анализа встречаемости букв*.

Арабы первыми обратили внимание на возможность использования стандартных слов и выражений для дешифрования. Так, первый широко известный среди арабов филолог VIII века Абу аль-Ахмади дешифровал криптограмму на греческом языке, которую переслал ему византийский император с просьбой о дешифровании, так объяснил свой метод: «Я сказал себе, что письмо должно начинаться со слов «Во имя Бога» или какого-нибудь в этом роде. Итак, я составил на основе этого первые буквы, и все оказалось правильным». На основе открытого им метода дешифрования он написал книгу «Книга тайного языка».

Со времен Цезаря до XV века шифровальное дело претерпело много изменений, однако мало известно о методах и системах шифрования, применяемых в Европе в этот период времени. В мрачные годы Средневековья практика шифрования сохранялась в строжайшей тайне. Так, в годы крестовых походов шифровальщики, служившие у Папы Римского, после года работы подлежали физическому уничтожению – так сохранялась тайна шифра.

### 1.3. Криптография в Эпоху Возрождения

Эпоха Возрождения стала временем не только всемирных открытий, но и великих тайн. Кто знал чужые секреты, тот вершил судьбы мира.

В Испании первые системы шифрования – преобразования открытого текста в римские цифры – появились в XV веке.

Еще один значительный шаг вперед криптография сделала благодаря труду Леона Альберти. Известный философ, живописец, архитектор, он в 1466

г. написал труд о шифрах. В этой работе был предложен шифр, основанный на использовании *шифровального диска* (рис. 1.5). Сам Альберти назвал его шифром, «достойным королей».

В Германии в XV–XVI веках значительный вклад в криптографию внес Иоганнес Тритемий, аббат монастыря в городе Вюрцбург. Это был образованный по меркам того времени человек, находившийся под личным покровительством императора Максимилиана I. Ученый богослов, натурализатор и историк, он заслужил славу чернокнижника своим сочинением «Полиграфия» – в те времена так называлась тайнопись, – написанным в 1499 году и вышедшем в

1518-м. Это была *первая печатная книга* по криптографии.



***Рис. 1.5.*** *Аббат Иоганнес Тритемий*

Тритемий – библиограф, алхимик, тайнописец – сделал два новаторских предложения в области криптографии: он предложил шифр «Аве Мария» и шифр, построенный на основе периодически сдвигаемого ключа.

Шифр «Аве Мария» основывался на принципе замены букв шифруемого текста на заранее оговоренные слова.

Шифр с периодически сдвигаемым ключом впервые представлялся в виде квадратной таблицы – *таблицы Тритемия* (рис. 1.6):

В первой строке был выписан сам алфавит. Следующая строка писалась со смещением на одну букву, следующая еще на одну и так до конца алфавита. Первая строка являлась одновременно и строкой букв открытого текста. Процесс зашифрования производился следующим образом: первая буква текста – по первой строке, вторая буква по второй и так далее, после использования последней строки вновь возвращались к первой.

Шифр Тритемия является первым примером *периодического многоалфавитного шифра*.

В дальнейшем усложнения шифра пошли по двум путям:

* введение произвольного порядка расположения букв исходного алфавита шифрованного текста вместо упорядоченного;
* применение усложненного порядка выбора строк таблицы при шифровании.

Следующий шаг в развитии предложенного Тритемием способа шифрования был сделан итальянцем Джованни Батиста Белазо. В 1553 году выходит в свет его брошюра «Шифр синьора Белазо». В этом шифре Белазо предложил использовать для многоалфавитного шифра буквенный, легко запоминаемый ключ, который он назвал *паролем*. Паролем могло служить слово или фраза. Пароль периодически записывался над открытым текстом. Буква пароля, расположенная над буквой текста, указывала на алфавит таблицы, который использовался для зашифрования этой буквы. Например, это мог быть алфавит из таблицы Тритемия, первой буквой которого являлась буква пароля. Однако Белазо, как и Тритемий, использовал в качестве шифралфавитов обычные алфавиты.

Воскресить смешанные алфавиты, которые применял Альберти, и объединить идеи Альберти с идеями Тритемия и Белазо в современную концепцию *многоалфавитной замены* выпало на долю итальянца Джованни де ла Порта. Ему было 28 лет, когда он в 1563 г. опубликовал книгу «О тайной переписке». По сути, эта книга являлась учебником по криптографии, содержащим криптографические познания того времени. Порта предложил использовать квадратную таблицу с периодически сдвигаемым смешанным алфавитом и паролем. Он советовал выбирать длинный ключ. Впервые им был предложен *шифр простой биграммной замены*, в котором пары букв представлялись одним специальным графическим символом. Символы заполняли квадратную таблицу размером 20х20, строки и столбцы которой были занумерованы буквами алфавита A B C D E F G H I L M N O P Q R S T U Z.

Известен многоалфавитный шифр Порта также в виде таблицы, но уже размером 24х24. Строки такой таблицы нумеровались буквами алфавита по порядку. Алфавит разбивался на полуалфавиты, которые в произвольном порядке были записаны в строки этой таблицы так, что каждая нечетная строка представляла собой первый полуалфавит, а каждая четная – второй. При этом порядок букв во втором полуалфавите не соблюдался. Шифрование осуществляется при помощи лозунга, который пишется над открытым текстом. Буква лозунга определяет алфавит (заглавные буквы первого столбца), расположенная под ней буква открытого текста ищется в первом полуалфавите и заменяется соответствующей ей буквой второго полуалфавита. Например, фраза, начинающаяся словами HUNC CAVETO VIRUM..., будет зашифрована при помощи лозунга DE LA PORTA в XFHP YTMOGA FQEAS.

В начале XVI века основатель разведывательной службы Англии Оливер

Кромвель учредил в рамках разведывательной службы на правах отделения специальную дешифровальную службу, которая оказала ему серьезные услуги в борьбе со сторонниками короля.

Крупнейший английский философ и ученый XVII века лорд-канцлер Френсис Бэкон уделял серьезное внимание шифрам. Он выдвинул главные требования к ним: «Они не должны поддаваться дешифрованию, не должны требовать много времени для написания и чтения и не должны возбуждать никаких подозрений». Эти требования сохраняются и сегодня.

Интерес к криптографии проявляли представители высшей знати. Так, в XVII веке Август II, герцог Ганновера, под псевдонимом Густав Селен опубликовал свою книгу «Секретные сообщения и криптография».

Прибегали к тайнописи и ученые того времени. Опасаясь преследования церкви за свои открытия и в то же время стараясь сохранить свой приоритет, они шифровали некоторые результаты своих исследований. К этому прибегал великий астроном Галилей.

В XIII–XIV веках методы криптографической защиты информации широко использовались в государствах Италии. Уже с ранних времен венецианское правительство имело особых специалистов-шифровальщиков, а за всей криптографической деятельность было поручено следить «Совету Десяти» – верховному органу исполнительной власти. Шифры обычно заключались в замене букв латинского алфавита либо другими буквами, либо арабскими цифрами, черточками, точками, произвольными фигурами, причем для одной буквы нередко использовали две-три замены. В шифрованный текст нередко вводились знаки, не имевшие никакого значения, для того, чтобы усложнить шифр. Эти знаки впоследствии стали называться *пустышками*.

Из дешифровальщиков особо выделялся Джованни Соро (XVI век) – один из первых известных крупных аналитиков-дешифровальщиков. Он дешифровал депеши Марка Антония Колонны – командующего армией императора Священной Римской империи; его услугами пользовалась папская курия – папа Климент VII лично направлял перехваченные депеши для дешифрования.

Во Флоренции жил искусный криптоаналитик Пирро Музефили, граф Сассетский, дешифровавший депеши французского короля Генриха II, кардинала де Мендосы и других. Среди его клиентов были герцог Альба и даже король Англии.

Шифрованная дипломатическая переписка часто вызывала неудовольствие со стороны заинтересованных дворов. Отказавшись от употребления шифров, дипломаты прибегли к стеганографическому способу защиты информации и использовали жаргонные коды, которые были известны ранее. Они заменяли смысл слов, при этом секретное сообщение принимало вид невинного текста, истинный смысл которого мог понять только посвященный в тайну замены слов.

В XIV–XV веках клерк тайной канцелярии папской курии Чикко Симонетти написал книгу о тайнописи. В ней он изложил усовершенствованные шифры замены, в том числе и *шифр многозначной замены (омофоны)*, в котором одной и той же букве (гласной) соответствует несколько шифробозначений. Симонетти обслуживал герцогов Сфорца, олигархов Милана, дешифруя для них перехваченные депеши. Он разработал 13 правил дешифрования шифров простой замены, которые изложил в рукописи на трех кусках пергамента.

Успеха в развитии криптографии в то время достиг Леон Альберти, человек универсального таланта, архитектор и математик. Он впервые выдвинул идею «*двойного шифрования*» – текст, полученный в результате первого шифрования, подвергался повторному шифрованию.

В XVI веке заметный вклад в развитие криптографии внес криптограф папы римского Маттео Ардженти, унаследовавший искусство тайнописи от своего дяди. Он рекомендовал не разделять слова, применять *омофонные* замены, вставлять в шифртекст большое количество пустышек, устранять пунктуацию, не вставлять в шифртекст открытые слова. Для затруднения дешифрования шифров замены он преложил заменить буквы либо на цифры, либо на двухзначные числа, придавая однозначные обозначения наиболее часто встречающимся буквам.

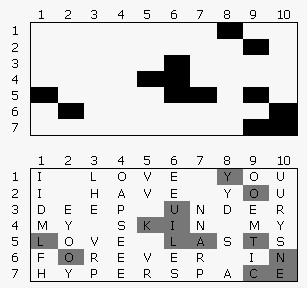


***Рис. 1.6.*** *Итальянский математик, философ, врач Джероламо Кардано (1506–1576)*

В 1550 году появляется книга математика, врача и философа Джероламо Кардано «О тонкостях» с дополнением «О разных вещах», в которой имеются разделы, посвященные криптографии. В ней нашел отражение способ шифрования, который вошел в историю как «*решетка Кардано*» (см. раздел 5.1 и Приложение 4). Для ее изготовления брался лист из твердого материала, представляющий собой квадрат, в котором вырезались «окна». При использовании всех окон решетка поворачивалась на 90 градусов и вновь буквы открытого текста вписывались в окна повернутой решетки. Если в квадрате после снятия решетки образовывались пустые места, то в них вписывались произвольные буквы. Затем буквы квадрата выписывались построчно, что и было шифрованным текстом. Это было не что иное, как *шифр перестановки* с элементами *стеганографии.*

При последовательном использовании (поворачивании) каждая клетка лежащего под ней листа должна была быть занятой. Главное требование к решетке Кардано – при всех поворотах окна не должны попадать на одно и то же место в квадрате, в котором образуется шифртекст. Такой шифр представлял собой шифр перестановки.

Кардинал Ришелье использовал идею квадрата Кардано. Из плотного материала вырезался прямоугольник, в нем проделывались окна. Секретный текст вписывался в эти окна, затем трафарет снимался и оставшиеся клетки заполнялись так, чтобы получалось невинное сообщение. Суровую команду на английском языке «You kill at once», с помощью решетки он спрятал в текст «I love you. I have you. Deep Under my skin. My love. Lasts forever in hyperspace».



***Рис. 1.7.*** *Трафарет Ришелье*

Следует заметить, что использованный Ришелье трафарет иллюстрирует скорее *стеганографическую* защиту информации, а не криптографическую. Подобным стеганографическим методом маскировки сообщения пользовались многие известные исторические лица, например русский дипломат и писатель А. Грибоедов. Так, Ришелье использовал прямоугольник размера 7х10. Для длинных сообщений прямоугольник использовался несколько раз.

Еще одно важное усовершенствование многоалфавитных систем, состоящее в идее использования в качестве ключа текста самого сообщения или же шифрованного текста, принадлежит Дж. Кардано и Блезу де Виженеру, французскому послу в Риме. Такой шифр получил название *самоключ*. В книге «Трактат о шифрах» Виженер представил самоключ в виде таблицы Тритемия с добавленными алфавитами к ней в естественном порядке в качестве первой строки и первого столбца. Позже такая таблица стала называться *таблицей Виженера*. Первая строка служила открытым алфавитом, а первый столбец – алфавитом ключа. В качестве ключевой последовательности использовался открытый текст с добавленной к нему первой буквой, известной и отправителю, и получателю. Такой механизм шифрования явился прообразом *шифров гаммирования*, широко используемых в настоящее время.

### 1.4. Криптография в Новое и Новейшее время

XVII–XVIII века вошли в историю криптографии как эра «черных кабинетов» – специальных государственных органов по перехвату и дешифрованию переписки. В штат черных кабинетов входили криптографы-дешифровальщики, агенты по перехвату почты, специалисты по вскрытию пакетов, писцы-копировальщики, переводчики, специалисты – граверы по подделке печатей, химики, специалисты по подделке почерков и так далее. Таким образом, «черные кабинеты» состояли из высококвалифицированных специалистов в различных областях деятельности. Эти специалисты ценились «на вес золота» и находились под особым покровительством властей. От них требовалось строгое сохранение тайны и преданность монарху. Предателей сурово наказывали. Криптографические службы работали в тесном контакте с агентурно-разведывательными органами.

Первый «черный кабинет» был образован во Франции в XVII веке. Его возглавил первый профессиональный криптограф Франции Антуан Россиньоль при Людовике XIV. Расцвет его таланта приходится на времена кардинала Мазарини. Помимо криптографии, он увлекался математикой, что, безусловно, сказалось на эффективности его криптографической деятельности. На основании ряда высказываний Россиньоля можно сделать вывод о том, что именно он впервые сформулировал *концепцию временной стойкости шифров*. Так, говоря о военных шифрах, он отмечал, что эти шифры должны быть просты, удобны для применения в боевых условиях и должны хранить тайну приказа до его выполнения. Такое требование к шифрам существенно упрощало и удешевляло решение проблемы надежной защиты быстро стареющей информации и позволяло применять в условиях боевых действий достаточно простые и дешевые шифры. После исполнения приказа допустимо его дешифрование противником.

Усилия по дешифрованию выполненного приказа становятся напрасными.

В конце XVIII века и в России был создан «черный кабинет». Дешифрованная корреспонденция докладывалась непосредственно императрице Екатерине II. Первым профессиональным дешифровальщиком в России стал Христиан Гольдбах – немецкий математик, приглашенный Петром I в начале XVIII века. А в середине того же века появились первые отечественные криптографы. Среди них выделялись Ерофей Каржавин (1719–1772) и его племянник Федор Каржавин (1745–1812). Каржавины внесли весомый вклад в развитие криптографической службы в России.

В XVIII веке одним из самых сильных в Европе был «черный кабинет» в Вене (Австрия). Его курировал сам император Карл VI и императрица Мария Тереза. Были разработаны первые «нормативные акты», регулирующие работу дешифровальщиков. Они имели следующие положения:

* предусмотреть денежную компенсацию дешифровальщикам за вынужденную безработицу;
* помимо текущей зарплаты предусмотреть солидные денежные премии за серьезные разовые успехи в дешифровании.

Были созданы *специальные курсы,* на которые направлялись юноши двадцатилетнего возраста. К абитуриентам предъявлялись особые требования: высокие моральные качества, знание иностранных языков, знакомство с математикой. В процессе обучения предусматривались особые тесты, предназначенные для определения способностей обучаемого к деятельности в области криптографии. К преподаванию на курсах привлекались криптографы, находящиеся на государственной службе. За эту работу предусматривалось дополнительное денежное вознаграждение.

Англичанин Джон Уилкинс, епископ Честера, основатель и секретарь королевского научного общества, издал книгу «Меркурий, или секретный и быстрый курьер», в которой впервые в английском языке появился термин *криптография* («секретность при письме») и *криптология* («секретность в речи»). В этой же книге он предложил разновидность *геометрического шифра* (рис. 1.8).

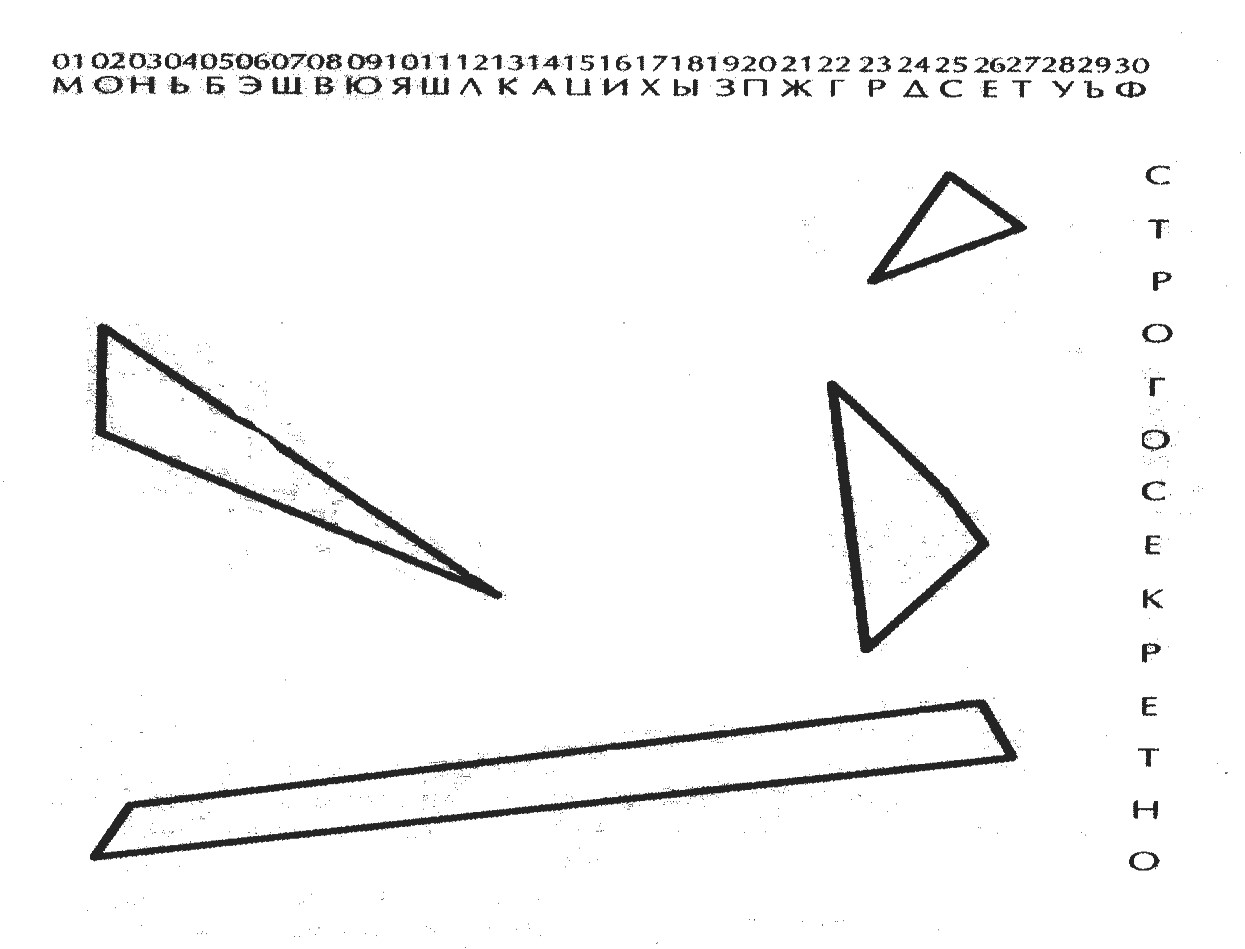
По сути дела это был шифр простой замены, но с применением стеганографии для передачи шифрованного текста: этот текст имел вид невинных геометрических фигур.

Передавалась только картинка. Ключом шифра является верхняя последовательность букв и расстояний между ними. Геометрическая фигура получалась путем произвольного соединения выделенных вершин прямыми линиями. Такой смешанный подход к защите информации активно используется и в настоящее время.

Уилкинс предложил также оригинальный музыкальный шифр. В нем секретные послания имели вид исполняемых музыкальных мелодий. Музыкальный шифр, как и геометрический, не был забыт и дошел до наших дней. Многие композиторы XIX–XX вв. использовали этот прием.

В начале XIX века французы перехватили и дешифровали письмо английского агента, написанное сплошь нотными знаками и по внешнему виду представляющее собой невинное музыкальное произведение.

Во время Второй мировой войны немецкий шпион в Англии был принят в качестве пианиста на радиостанцию BBC. Его основная цель заключалась в передаче секретных сведений в Германию в исполняемых им музыкальных произведениях.



***Рис. 1.8.*** *Пример зашифрования словосочетания СТРОГОСЕКРЕТНО геометрическим шифром*

Несмотря на появление криптографии в литературно-художественных произведениях, появление учебных учреждений для подготовки профессиональных криптографов, появление первых юридических актов, регулирующих криптографическую деятельность государства, криптография являлась скорее искусством одиночек, чем объективной наукой. Научные основы криптографии появились лишь после возникновения и развития серьезных математических методов исследования.

Требования к криптосистеме были впервые изложены в книге О. Керкгоффса «Военная криптография» (издана в 1883 году). Шесть основных требований к криптосистеме, все из которых до настоящего времени определяют проектирование криптографически стойких систем, в переводе с французского звучат так:

|  |  |  |  |
| --- | --- | --- | --- |
| Система должна быть нераскрываемой, если не математически, то практически. | | | |
|  | |  | |
|  |  | 1. Компрометация системы не должна причинять неудобств корреспондентам (*принцип Керкгоффса*). 2. Ключ можно легко передать и запомнить без каких-либо записей; у корреспондентов должна быть возможность по собственной воле менять ключ. 3. Система должна быть применима к телеграфной связи. 4. Система должна быть портативной; для ее обслуживания должно быть достаточно одного человека. 5. Наконец, необходимо, чтобы система была простой в использовании, и ее применение не требовало ни соблюдения длинного списка правил, ни большого умственного напряжения. |  |
|  |  | |  |
|  | |

***Рис. 1.9.*** *Огюст Керкгоффс*

Доктор Огюст Керкгоффс (Auguste Kerckhoffs, 1835–1903) – голландский лингвист и криптограф, профессор Парижской высшей школы коммерции во второй половине XIX века. Керкгоффс родился в городе Нют, Голландия. При крещении получил имя Жан Вильгельм Губерт Виктор Франсуа Александр Огюст Керкгоффс фон Ньювенгоф. Огюст закончил Льежский университет, преподавал в Голландии и во Франции. В своей книге «Военная криптография» сформулировал ответы на вопросы, актуальность которых для криптографии обозначилась только в XX веке. В сжатой, системной форме Огюст Керкхофф изложил требования к криптографическим системам, а также показал важнейшую роль криптоанализа для их проверки и подтверждения стойкости. Одно из требований теперь известно как принцип Керкгоффса.

Выдающиеся результаты в применении математических методов в криптографии принадлежат Клоду Шеннону. Он получил образование по электронике и математике в Мичиганском университете, где и начал проявлять интерес к теории шифров.

К 1944 году Шеннон завершил разработку теории секретной связи. В 1945м им был подготовлен секретный доклад «Математическая теория криптографии», который был рассекречен и издан в 1949 году.

В данной работе излагается теория секретных шифров, служащих фактически математической моделью шифров.

Разработанные Шенноном концепции теоретической и практической секретности позволяют количественно оценивать криптографические качества шифров и пытаться строить идеальные или *совершенные шифры*.

Центральной в работах Шеннона является концепция *избыточной информации*, содержащейся в текстовых сообщениях. Избыточность означает, что в сообщении содержится больше символов, чем в действительности требуется для передачи содержащейся в нем информации. Если шифрование «стирает» избыточность, то восстановить текст по криптограмме криптоаналитику становится принципиально невозможно.

Также Шеннон ввел понятие рабочей характеристики шифра, подходя к практической оценке стойкости.

Еще задолго до К. Шеннона частотные характеристики языка изучал выдающийся русский ученый А.А. Марков. Сегодня в криптографии часто используются *марковские модели открытых текстов*, учитывающие зависимость букв текста от предыдущих букв. Изучением количества информации в текстах и свойств языка занимались А.Н. Колмогоров, Б.Б. Пиотровский.

XX век в истории криптографии отмечен появлением *телефонных шифраторов*. В России разработкой телефонного шифратора руководил В.А. Котельников, ставший впоследствии академиком, ученым с мировым именем. Ему принадлежит *теорема дискретизации (теорема отсчетов)*, лежащая в основе *цифровой обработки данных*.

Во второй половине XX века с развитием вычислительной техники появились *электронные шифраторы и программные реализации криптографических алгоритмов*.

В 70-х годах в Америке был принят и опубликован *первый стандарт шифрования* данных (DES).

В России установлен единый алгоритм криптографического преобразования данных для систем обработки информации в сетях ЭВМ, отдельных вычислительных комплексах и ЭВМ, определяемый ГОСТ 28147-89. Алгоритм предназначен для аппаратной и программной реализации и не накладывает ограничений на степень секретности.

В 1976 году американцы Уитфилд Диффи (Whitfield Diffie) и Мартин Хеллман (Martin Hellman), два инженера-электрика и, независимо от них, Ральф Меркль (Ralf Merkle), бывший в то время студентом, опубликовали в статье

«Новые направления в криптографии» новый принцип построения криптосистем, не требующий передачи ключа и не требующий сохранения в тайне метода шифрования – принцип *криптографии с открытым ключом*. А в 1978 г. последовала первая практическая реализация криптографии с открытым ключом, предложенная Рональдом Райвистом (Ronald Rivest), Ади Шамиром (Adi Shamir) и Леонардом Адлеманом (Leonard Adleman) и получившая название *RSA* (по первым буквам фамилий ее создателей), в которой для зашифрования и расшифрования использовались разные ключи (открытые и секретные). В связи с несимметричным использованием ключей появился термин *асимметричные шифрсистем*ы, в то время как традиционные шифрсистемы стали называться *симметричными*.

Появление систем шифрования с открытым ключом сделало возможным решить проблему *цифровых подписей*: вычисление подписи может выполняться автором с использованием секретного ключа, а проверка подписи производится с открытым ключом и общедоступна. В настоящее время уже имеются стандарты схем цифровой подписи, и этот механизм все шире распространяется во все сферы информационного обмена.

Стивен Визнер (Stephen Wiesner), являясь студентом Колумбийского университета, в 1970 году подал статью по теории кодирования в журнал IEEE Information Theory. В ней была описана идея возможности использования квантовых состояний для защиты денежных банкнот. Спустя более 10 лет ученые Чарльз Беннет (Charles Bennett) из фирмы IBM и Жиль Брассард (Gilles Brassard) из Монреальского университета на основе принципов работы С. Визнера разработали способ кодирования и передачи сообщений. Ими был сделан доклад на тему «Квантовая криптография: распределение ключа и подбрасывание монет» на конференции IEEE International Conference on Computers, Systems, and Signal Processing. Описанный в докладе протокол впоследствии стал базовым протоколом квантовой криптографии и был назван в честь его создателей BB84.

В настоящее время действующая аппаратура позволяет распределять ключи через квантовый канал на расстояния, превышающие 100 км, со скоростями, достаточными для передачи ключей шифрования, но не достаточными для шифрования. Однако высокая стоимость квантовых систем распределения ключей ограничивает их массовое применение для организации конфиденциальной переписки.

#### Вопросы и задания для самостоятельной работы

1. В чем состоит сущность шифрования и расшифрования информации?
2. Соотнесите между собой понятия «криптография», «криптоанализ» и «криптология».
3. Приведите исторические примеры зарождения криптографии.
4. Охарактеризуйте донаучный период развития криптографии. Проанализируйте первые шифры замены и перестановки.
5. С какого времени начался научный период развития криптографии? Какие события стали для этого определяющими?
6. Чем определяется криптостойкость шифрования? Какие другие требования предъявляются к шифрованию?
7. Какие шифры называются омофонами?
8. В чем состоит правило Керкгоффса? Почему это правило является общепринятым в криптографии?
9. Когда родилась криптография с открытыми ключами и первая реальная система шифрования?
10. Каких выдающихся криптографов ХХ в. вы знаете?

## ТЕМА 2. ОСНОВНЫЕ ЗАДАЧИ СОВРЕМЕННОЙ КРИПТОГРАФИИ

Криптография возникла как наука о шифровании данных. Первые шифры были изобретены еще в глубокой древности, и в течение целых тысячелетий единственная задача, которая ставилась перед криптографией, состояла в обеспечении конфиденциальности.

В последние десятилетия в результате бурного развития вычислительной техники и средств связи потребность в защите информации резко возросла. При этом во многих прикладных областях основной задачей становится обеспечение целостности информации, под которой понимается гарантия поступления информации из достоверного источника и в неискаженном виде. Обеспечение целостности информации – вторая задача криптографии.

Автоматизированные системы банковских расчетов – один из примеров прикладной области, для которой обеспечение целостности оказалось более важной задачей, чем обеспечение конфиденциальности. К моменту разработки первых автоматизированных банковских систем криптография не могла предложить для защиты информации никаких других методов, кроме достаточно хорошо разработанных средств шифрования данных. Поэтому первые попытки обеспечения целостности в банковских системах основывались на использовании шифраторов. Однако в начале 80-х годов прошлого столетия было осознано, что для обеспечения целостности требуются иные, чем для обеспечения конфиденциальности, криптографические средства. Так возникла новая отрасль криптографии – *криптографические протоколы.*

Под криптографическим протоколом обычно понимается распределенный алгоритм решения двумя или более участниками некоторой криптографической задачи (см. главу 10). Для криптографических протоколов характерным является изменение представлений о противнике: если при разработке криптосистем предполагается внешний противник, пытающийся получить конфиденциальную информацию, передаваемую между честными, доверяющими друг другу участниками, то в криптографических протоколах противником может быть также один или несколько участников протокола. Иными словами, криптографические протоколы предназначены, прежде всего, для защиты их участников от нечестных действий партнеров.

Первые криптографические протоколы (схемы электронной подписи и протоколы аутентификации) были разработаны в 70-х годах. С тех пор эта отрасль развивалась очень бурно. По крайней мере, в теоретической криптографии протоколы стали основной областью исследований.

Таким образом, современная криптография дает возможность решить такие проблемы безопасности информации, как конфиденциальность, целостность, аутентификация и невозможность отказа сторон от авторства. Достижение этих требований безопасности в процессе информационного взаимодействия составляет пять основных задач криптографии:

* обеспечение *конфиденциальности* – это решение проблемы защиты информации от ознакомления с ее содержанием со стороны лиц, не имеющих права доступа к ней (в зависимости от контекста вместо термина «конфиденциальная» информация могут выступать термины «секретная», «частная», «информация ограниченного доступа»);
* обеспечение *целостности* – это гарантирование невозможности несанкционированного изменения информации (т.е. несанкционированных вставки, удаления и замены данных);
* обеспечение *аутентификации* – это разработка методов подтверждения подлинности сторон и самой информации в процессе информационного взаимодействия (при этом информация, передаваемая по каналу связи, должна быть аутентифицирована по источнику, времени создания, содержанию данных, времени пересылки и т.д.);
* обеспечение *невозможности отказа от авторства или приписывания авторства* – это предотвращение возможности отказа субъектов от некоторых из совершенных ими действий;
* обеспечение *неотслеживаемости* – это предотвращение возможности получения противником или нарушителем сведений о действиях участников информационного обмена.

Еще одной проблемой, вызванной решением перечисленных задач, является *управление ключами* (распределение ключей, сертификация и построение схем разделения секрета и т.п.). Эта проблема также решается средствами современной криптографии.

### 2.1. Конфиденциальность

Обеспечение конфиденциальности информации при передаче сообщений по контролируемому противником каналу связи является традиционной задачей криптографии. В простейшем случае эта задача описывается взаимодействием трех субъектов (рис. 2.1). Владелец информации, называемый обычно *отправителем,* осуществляет преобразование исходной *(открытой)* информации (сам процесс преобразования называется *шифрованием*)в форму передаваемых *получателю* по открытому каналу связи *шифрованных* сообщений с целью ее защиты от противника.

Открытое

сообщение

Шифрованное

сообщение

Открытое

сообщение

Шифрование

Расшифрование

Отправитель

Получатель

Противник

***Рис. 2.1.*** *Передача шифрованной информации*

Под *противником* понимается любой субъект, не имеющий права ознакомления с содержанием передаваемой информации. В качестве противника может выступать *криптоаналитик,* владеющий методами раскрытия шифров. Законный получатель информации осуществляет *расшифрование* полученных сообщений.

Противник пытается овладеть защищаемой информацией (его действия обычно называют атаками)*.* При этом он может совершать как пассивные, так и активные действия.

*Пассивные атаки* связаны с прослушиванием, анализом трафика, перехватом, записью передаваемых шифрованных сообщений, *дешифрованием,* т.е. попытками «взломать» защиту с целью овладения информацией.

При проведении *активных атак* противник может прерывать процесс передачи сообщений, создавать поддельные сообщения (такие действия называются *имитацией*) или модифицировать передаваемые шифрованные сообщения (такие действия называются *подменой*).

Под *шифром* обычно понимается семейство обратимых преобразований, каждое из которых определяется некоторым параметром, называемым ключом, а также порядком применения данного преобразования, называемым *режимом шифрования.*

*Ключ* – это важнейший компонент шифра, отвечающий за выбор преобразования, применяемого для зашифрования конкретного сообщения. Обычно ключ представляет собой некоторую буквенную или числовую последовательность. Эта последовательность «настраивает» алгоритм шифрования.

Каждое преобразование однозначно определяется ключом и описывается некоторым *криптографическим алгоритмом* (способом шифрования)*.* Один и тот же криптографический алгоритм может применяться для шифрования в различных режимах. Тем самым реализуются различные способы шифрования.

Например, простая замена (см. раздел 6.1) или гаммирование (см. раздел 7).

Каждый режим шифрования имеет как свои преимущества, так и недостатки. Поэтому выбор режима зависит от конкретной ситуации. При расшифровании используется криптографический алгоритм, который в общем случае может отличаться от алгоритма, применяемого для зашифрования сообщения.

Соответственно, могут различаться ключи зашифрования и расшифрования.

Пару алгоритмов зашифрования и расшифрования обычно называют *шифрсистемой,* а реализующие их устройства – *шифртехникой*.

Если обозначить через ***М*** открытое, а через ***С*** шифрованное сообщения, то процессы зашифрования и расшифрования можно записать в виде равенств:

*Ек1 (М)* = *С,*

*Dк2 (C)* = *M,*

где *k1* и *k2* – ключи зашифрования и расшифрования соответственно.

Алгоритмы зашифрования ***Е*** и расшифрования ***D***должны удовлетворять равенству:

*Dк2 (Ек1 (М)) = M.*

Различают *симметричные* и *асимметричные* шифрсистемы. В симметричных системах знание ключа зашифрования *k1* позволяет легко найти ключ расшифрования *k2* (в большинстве случаев эти ключи просто совпадают). В асимметричных криптосистемах знание ключа *k1* не позволяет определить ключ *k2.* Поэтому для симметричных шифрсистем оба ключа должны сохраняться в секрете, а для асимметричных – только один, ключ расшифрования *k2*,а ключ *k1* можно сделать открытым (общедоступным). В связи с этим их называют еще *шифрами с открытым ключом.*

Симметричные шифрсистемы принято подразделять на поточные и блочные системы. *Поточные* системы осуществляют зашифрование отдельных символов открытого сообщения. *Блочные* системы производят зашифрование блоков фиксированной длины, составленных из подряд идущих символов сообщения.

Асимметричные шифрсистемы, как правило, являются блочными. При их использовании можно легко организовать передачу конфиденциальной информации в сети с большим числом пользователей. Для того чтобы послать сообщение, отправитель открыто связывается с получателем, который либо передает свой ключ отправителю, либо помещает его на общедоступный сервер. Отправитель зашифровывает сообщение на открытом ключе получателя и отправляет его получателю. При этом никто, кроме получателя, обладающего ключом расшифрования, не сможет ознакомиться с содержанием передаваемой информации. В результате такая система шифрования с общедоступным ключом позволяет существенно сократить объем хранимой каждым абонентом секретной ключевой информации.

Возможна и другая симметричная ситуация, когда открытый и секретный ключи меняются местами. Предположим, например, что для проведения контроля соблюдения выполнения каждой стороной договора об ограничении испытаний ядерного оружия создаются пункты контроля, которые ведут запись и конфиденциальную передачу сторонам, участвующим в договоре, сейсмологической информации. Поскольку на каждом таком пункте контролируемая сторона одна, а участников договора может быть очень много, то необходимо обеспечить такое шифрование информации, при котором зашифровать сообщение мог бы только один отправитель, а расшифровать мог бы каждый.

Не существует единого шифра, подходящего для всех случаев жизни. Выбор способа шифрования (то есть криптографического алгоритма и режима его использования) зависит от особенностей передаваемой информации (ее ценности, объема, способа представления, необходимой скорости передачи и т.д.), а также возможностей владельцев по защите своей информации (стоимость применяемых технических устройств, удобство использования, надежность функционирования и т.п.). Имеется большое разнообразие видовзащищаемой информации: текстовая, телефонная, телевизионная, компьютерная и т.д., причем у каждого вида информации имеются свои существенные особенности, которые надо учитывать при выборе способа шифрования.

Большое значение имеют объемы и требуемая скорость передачи шифрованной информации, а также помехозащищенность используемого канала связи. Все это существенным образом влияет на выбор криптографического алгоритма и организацию защищенной связи.

Наличие надежного криптографического алгоритма и правильный выбор режима еще не гарантируют владельцу защищенность передаваемой информации. Немаловажную роль играет правильность их использования. Поскольку даже самые стойкие шифры при неправильном использовании существенно теряют свои качества, то конфиденциальность передаваемой информации во многом зависит от того, какие *ошибки* допускает ее владелец при использовании криптографической защиты. А то, что все пользователи допускают ошибки, – неизбежно и является непреложным и важным (для криптоаналитика) фактом, поскольку любые криптографические средства, какими бы они ни были удобными и прозрачными, всегда мешают пользователям в работе, а различные тонкости известны только криптоаналитикам и, как правило, непонятны пользователям этих средств.

В качестве субъектов взаимодействия могут выступать не только люди, но и различные процессы, осуществляющие обработку информации в автоматизированной системе без участия человека. Поэтому защищенность информации в системе существенно зависит от того, насколько правильно там реализована криптографическая подсистема, отвечающая за выполнение криптографических функций.

Для разных шифров задача вскрытия имеет различную сложность. Уровень сложности этой задачи и определяет главное свойство шифра – способность противостоять попыткам противника завладеть защищаемой информацией. В связи с этим говорят о *криптографической стойкости* шифра (или просто *стойкости*),различая более стойкие и менее стойкие шифры.

### 2.2. Целостность

Обеспечение целостности информации, то есть неизменности ее в процессе передачи или хранения, является второй важнейшей задачей криптографии.

Решение этой задачи предполагает разработку средств, позволяющих обнаруживать не столько случайные искажения (для этой цели вполне подходят методы теории кодирования с обнаружением и исправлением ошибок), сколько целенаправленное навязывание противником ложной информации. Для этого в передаваемую информацию вносится *избыточность*. Как правило, это достигается добавлением к сообщению некоторой проверочной комбинации, вычисляемой с помощью специального алгоритма и играющей роль контрольной суммы для проверки целостности полученного сообщения.

Главное отличие такого метода от методов *теории кодирования* (науки, изучающей методы защиты информации от случайных искажений в каналах связи) состоит в том, что алгоритм выработки проверочной комбинации является «криптографическим», то есть зависящим от секретного ключа. Без знания секретного ключа вероятность успешного навязывания противником искаженной или ложной информации мала. Такая вероятность служит мерой *имитостойкости* шифра, то есть способности самого шифра противостоять активным атакам со стороны противника.

Итак, для проверки целостности к сообщению ***М***добавляется проверочная комбинация***S****,* называемая *кодом аутентичности сообщения*, или *имитовставкой.* В этом случае по каналу связи передается пара

***С* = *(М,S).***

При получении сообщения ***М***пользователь вычисляет значение проверочной комбинации и сравнивает его с полученным контрольным значением ***S****.* Несовпадение говорит о том, что данные были изменены.

Как правило, код аутентичности сообщения является значением некоторой (зависящей от секретного ключа) криптографической *хэш-функции* от данного сообщения:

***hk(М) = S.***

К ключевым хэш-функциям предъявляются определенные требования:

* невозможность вычисления значения ***hk(М) = S***для заданного сообщения ***М***без знания ключа***k****;*
* невозможность подбора для заданного сообщения ***М***с известным значением ***hk (М) = S***другого сообщения ***M*1** с известным значением ***hk (M1) = S1*** без знания ключа ***k****.*

Первое требование направлено против создания поддельных (сфабрикованных) сообщений при атаках типа *имитация;* второе – против модификации передаваемых сообщений при атаках типа *подмена.*

### 2.3. Аутентификация

*Аутентификация* – установление подлинности. В общем случае этот термин может относиться ко всем аспектам информационного взаимодействия: сеансу связи, сторонам, передаваемым сообщениям и т.д.

Установление подлинности (то есть проверка и подтверждение) всех аспектов информационного взаимодействия является важной составной частью проблемы обеспечения достоверности получаемой информации. Особенно остро эта проблема стоит в случае не доверяющих друг другу сторон, когда источником угроз может служить не только третья сторона (противник), но и сторона, с которой осуществляется взаимодействие.

Аутентификация *сеанса связи* (транзакции) означает проверку: - целостности соединения;

- невозможности повторной передачи данных противником; - своевременности передачи данных.

Для этого, как правило, используют дополнительные параметры, позволяющие «сцепить» передаваемые данные в легко проверяемую последовательность. Это достигается, например, путем вставки в сообщения некоторых специальных чисел или *меток времени.* Они позволяют предотвратить попытки повторной передачи, изменения порядка следования или обратной отсылки части переданных сообщений. При этом такие вставки в передаваемом сообщении необходимо защищать (например, с помощью шифрования) от возможных подделок и искажений.

Аутентификация *сторон взаимодействия* означает проверку одной из сторон того, что взаимодействующая с ней сторона – именно та, за которую она себя выдает. Это достигается присвоением каждой стороне уникального системного имени-идентификатора (*идентификация*). Идентификация заключается в предъявлении этого имени и предшествует аутентификации, то есть подтверждению правильности идентификации.

Основным средством для проведения идентификации являются *протоколы идентификации,* позволяющие осуществлять идентификацию (и аутентификацию) каждой из участвующих во взаимодействии и не доверяющих друг другу сторон. Различают *протоколы односторонней* и *взаимной идентификации.*

Протокол, как уже упоминалось в начале главы, представляет собой распределенный алгоритм, определяющий последовательность действий каждой из сторон. В процессе выполнения протокола идентификации каждая из сторон не передает никакой информации о своем секретном ключе, а хранит его у себя и использует для формирования ответных сообщений на запросы, поступающие при выполнении протокола.

Аутентификация самой *информации* означает проверку того, что информация, передаваемая по каналу, является подлинной:

* по содержанию;
* по источнику;
* по времени создания;
* по времени пересылки и т.д.

Проверка подлинности *содержания* информации сводится, по сути, к проверке ее неизменности (с момента создания) в процессе передачи или хранения, то есть проверке целостности.

*Аутентификация источника данных* означает подтверждение того, что исходный документ был создан именно заявленным источником. Если стороны доверяют друг другу и обладают общим секретным ключом, то аутентификацию сторон можно обеспечить применением кода аутентификации. Действительно, каждое успешно декодированное получателем сообщение может быть создано только отправителем, так как только он знает их общий секретный ключ. Для не доверяющих друг другу сторон решение подобных задач с использованием общего секретного ключа становится невозможным. Поэтому при аутентификации источника данных нужен механизм *цифровой подписи*. В целом аутентификация источника данных выполняет ту же роль, что и протокол идентификации. Отличие заключается только в том, что в первом случае имеется некоторая передаваемая информация, авторство которой требуется установить, а во втором требуется просто установить сторону, с которой осуществляется взаимодействие.

### 2.4. Неотслеживаемость

Пятая задача криптографии – обеспечение неотслеживаемости. Она была поставлена в начале 80-х годов. Определяется для криптографических систем с большим количеством участников: систем электронных платежей, доступа к электронным информационным фондам и т.п.

В самом общем виде проблема связана со все возрастающей компьютеризацией различных сфер деятельности и, главным образом, с тем, что все большее и большее количество организаций, как государственных, так и коммерческих, предоставляют доступ к ресурсам и услугам через компьютерные сети. Но при обращении в организацию, как правило, каждый клиент должен доказать свое право на получение данной услуги или на доступ к информации. Обычно это связано с тем, что услуги платные. Для доказательства своих прав клиент в ныне используемых системах должен идентифицировать себя. Во многих странах у всех граждан имеются универсальные идентификационные номера, проставленные в их удостоверениях личности. Разработчики большинства систем компьютерного доступа к услугам и ресурсам не находят ничего лучшего, чем использование этих номеров для идентификации клиентов. В результате все действия отдельного клиента могут быть отслежены по его идентификационному номеру, и организация может вести досье на каждого клиента. Дальнейшее внедрение компьютерных систем в новые сферы человеческой деятельности создаст возможности для беспрецедентной тотальной слежки за всеми гражданами.

Для предотвращения подобной угрозы предлагаются два подхода. Первый предполагает наказывать тех, кто использует компьютерные системы для незаконного создания досье (во всех цивилизованных странах такая деятельность без санкции прокурора является преступлением). Слабость этого метода очевидна, поскольку злоупотребления в компьютерных системах практически необнаружимы и недоказуемы.

Второй подход состоит в использовании криптографических средств обеспечения неотслеживаемости. Например, клиент может обратиться в некоторую организацию и получить право на доступ к определенной информационной базе данных сроком на год. Для получения такого права клиент должен себя идентифицировать. Под правом (credential) здесь понимается информационная строка специального вида, подписанная организацией и содержащая идентификатор клиента. Особенность математического аппарата состоит в том, что клиент может свое право преобразовать таким образом, чтобы впоследствии обращаться к базе данных под псевдонимом, никак не связанным с его *идентификационным номером*. Разумеется, организация, контролирующая доступ к базе данных, может проверять законность прав клиентов. При этом клиенты могут создавать права самостоятельно (без участия выдающей их организации) лишь с пренебрежимо малой вероятностью.

Таким образом, можно сказать, что имеются две крайних ситуации: тотальная слежка и абсолютная неотслеживаемость. Последняя также должна рассматриваться как крайность, поскольку цивилизованное общество никогда не согласится с полной неотслеживаемостью действий преступников. Если здесь будет найдено приемлемое решение, то обеспечение неотслеживаемости станет основной задачей криптографии, поскольку в современном компьютеризированном обществе проблемы неотслеживаемости касаются буквально каждого.

В последние годы интерес к задаче обеспечения неотслеживаемости заметно возрос. Это отчасти связано с попытками обеспечения неотслеживаемости в системах *электронных платежей*. Клиентам необходимо нечто, аналогичное свойству анонимности обычных бумажных денег. Хотя каждая бумажная купюра имеет уникальный номер, определить, кто ее использовал и в каких платежах, практически невозможно. Аналог этого свойства в криптографии называется *неотслеживаемостью*.

Если задача обеспечения конфиденциальности решается с помощью криптосистем, то для обеспечения неотслеживаемости разрабатываются криптографические протоколы. Отличия криптографических протоколов от криптосистем можно выделить следующие:

* протоколы могут быть интерактивными, т.е. подразумевать многораундовый обмен сообщениями между участниками;
* в протоколе может быть более двух участников;
* участники протокола, в общем случае, не доверяют друг другу. Поэтому криптографические протоколы должны защищать их участников не только от внешнего противника, но и от нечестных действий партнеров.

### 2.5. Цифровая подпись

*Цифровая подпись* для сообщения – это число, полученное в результате криптографических преобразований, примененных к этому сообщению. Поэтому цифровая подпись зависит от самого сообщения и от некоторого секретного, известного только подписывающему субъекту, ключа.

В некоторых ситуациях, например в силу изменившихся обстоятельств, отдельные лица могут отказаться от ранее принятых обязательств. В связи с этим необходим некоторый механизм, препятствующий подобным попыткам.

Так как в данной ситуации предполагается, что стороны не доверяют друг другу, то использование общего секретного ключа для решения поставленной проблемы становится невозможным. Отправитель может отказаться от факта передачи сообщения, утверждая, что его создал сам получатель *(отказ от авторства).* Получатель легко может модифицировать, подменить или создать новое сообщение, а затем утверждать, что оно получено от отправителя *(приписывание авторства).* Ясно, что в такой ситуации арбитр при решении спора не будет иметь возможность установить истину.

Основным механизмом решения этой проблемы является *цифровая подпись.*

Хотя цифровая подпись и имеет существенные отличия, связанные с возможностью отделения от документа и независимой передачей, а также возможностью подписывания одной подписью всех копий документа, она во многом аналогична обычной «ручной» подписи.

*Схема цифровой подписи* включает два алгоритма: - для вычисления подписи; - для проверки подписи.

Вычисление подписи может быть выполнено только автором подписи. Алгоритм проверки должен быть общедоступным, чтобы проверить правильность подписи мог каждый.

Для создания схемы цифровой подписи можно использовать симметричные шифрсистемы. В этом случае подписью может служить само зашифрованное на секретном ключе сообщение. Однако основной недостаток таких подписей состоит в том, что они являются одноразовыми: после каждой проверки секретный ключ становится известным. Единственный выход из этой ситуации в рамках использования симметричных шифрсистем – это введение доверенной третьей стороны, выполняющей функции *посредника*, которому доверяют обе стороны. В этом случае вся информация пересылается через посредника, он осуществляет перешифрование сообщений с ключа одного из абонентов на ключ другого. Естественно, эта схема является крайне неудобной.

При использовании шифрсистем с открытым ключом возможны два подхода к построению системы цифровой подписи.

Первый подход состоит в преобразовании сообщения в форму, по которой можно восстановить само сообщение и тем самым проверить правильность «подписи». В данном случае подписанное сообщение имеет, как правило, ту же длину, что и исходное сообщение. Для создания такого «подписанного сообщения» можно, например, произвести зашифрование исходного сообщения на секретном ключе автора подписи. Тогда каждый может проверить правильность подписи путем расшифрования подписанного сообщения на открытом ключе автора подписи.

При втором подходе подпись вычисляется и передается вместе с исходным сообщением. Вычисление подписи заключается в преобразовании исходного сообщения в некоторую цифровую комбинацию (которая и является подписью). Алгоритм вычисления подписи должен зависеть от секретного ключа пользователя. Это необходимо для того, чтобы воспользоваться подписью мог бы только владелец ключа. В свою очередь, алгоритм проверки правильности подписи должен быть доступен каждому. Поэтому, как правило, этот алгоритм зависит от открытого ключа пользователя. В данном случае длина подписи не зависит от длины подписываемого сообщения.

Одновременно с проблемой цифровой подписи возникла проблема построения бесключевых криптографических *хэш-функций.* Дело в том, что при вычислении цифровой подписи оказывается более удобным осуществить сначала *хэширование*, то есть свертку текста в некоторую комбинацию фиксированной длины, а затем уже подписывать полученную комбинацию с помощью секретного ключа. При этом функция хэширования, хотя и не зависит от ключа и является открытой, должна быть «криптографической». Имеется в виду свойство *односторонности* этой функции: по значению комбинации-свертки никто не должен иметь возможность подобрать соответствующее сообщение.

В настоящее время имеются стандарты на криптографические хэшфункции, утверждаемые независимо от стандартов на криптографические алгоритмы и схемы цифровой подписи (см. главу 10).

### 2.6. Управление ключами

Система управления ключами состоит из подсистем управления открытыми и секретными ключами.

### *Открытые ключи*

Задача подсистемы открытых ключей заключается в обеспечении целостности и аутентичности открытых ключей.

Перед тем, как использовать открытый ключ некоторого абонента для отправки ему конфиденциальной информации, отправитель должен быть уверен, что открытый ключ действительно принадлежит этому абоненту и что злоумышленник не мог подменить этот открытый ключ и выступить от его имени. Для защиты открытых ключей создаются специальные *центры сертификации*, которые выступают доверенной третьей стороной и заверяют ключи каждого абонента своими цифровыми подписями.

Центр сертификации предназначен для регистрации абонентов, изготовления *сертификатов* открытых ключей, хранения изготовленных сертификатов, поддержания в актуальном состоянии справочника действующих сертификатов и выпуска списка досрочно отозванных сертификатов.

Сертификат – это набор данных, заверенных цифровой подписью центра сертификации и включающий открытый ключ и список дополнительных атрибутов, принадлежащих абоненту. К таким атрибутам относятся: имя пользователя, имя сертификационного центра, номер сертификата, время действия сертификата, предназначение открытого ключа (цифровая подпись, шифрование) и т.п.

Центры сертификации объединяются в иерархическую (древовидную) структуру, в корне которой находится главный центр сертификации, выдающий сертификаты подчиненным ему отраслевым центрам, обеспечивая доверие к открытым ключам этих центров. Каждый центр вышестоящего уровня делегирует аналогичным образом право выпуска сертификатов подчиненных ему центров. Таким образом, доверие сертификату открытого ключа каждого центра основано на заверении его сертификата ключом вышестоящего центра. Сертификаты главного центра подписывает сам главный центр.

### *Секретные ключи*

Управление секретными ключами осуществляют системы *установки ключей и управления* ключами. В совокупности они определяют порядок использования криптографической системы.

Система установки ключей решает следующие задачи: определяет алгоритмы и процедуры генерации, распределения, передачи и проверки ключей.

Система управления ключами определяет порядок использования, смены, хранения и архивирования, резервного копирования и восстановления, замены или изъятия из обращения скомпрометированных, а также уничтожения старых ключей.

Самый надежный способ первоначального распределения ключей – это личная встреча всех взаимодействующих сторон. При большом числе последних требуется предварительная рассылка значительного объема ключевой информации и последующее ее хранение. Для этого применяются специальные *системы предварительного распределения ключей*, предусматривающие распределение и хранение не самих ключей, а некоторой меньшей по объему информации, на основе которой каждая сторона может вычислить ключ для взаимодействия с другой стороной.

Система предварительного распределения ключей включает два алгоритма:

* алгоритм генерации исходной открытой информации (которая передается всем сторонам или хранится на общедоступном сервере) и секретной части каждой стороны;
* алгоритм вычисления действующего значения ключа по исходной информации (открытой и секретной частей) для взаимодействия между абонентами.

Система предварительного распределения ключей должна быть *устойчивой и гибкой.* Устойчивость системы означает учет возможности раскрытия части ключей при компрометации, обмане, сговоре некоторых абонентов. Гибкость системы определяет возможности быстрого восстановления путем исключения скомпрометированных и подключения новых абонентов.

Пересылка ключей может осуществляться с помощью шифрования с использованием ранее полученных ключей или через специальные доверенные центры, выполняющие функции центров распределения или перешифрования ключей.

Для сетей с большим количеством абонентов распределение ключей наиболее просто может осуществляться в *системах открытого распределения секретных ключей*. В такой системе каждая из сторон изначально имеет свой секретный параметр. Стороны обмениваются некоторыми сообщениями по открытым каналам связи и по результатам этого обмена вычисляют секретный связной ключ. Протоколы такого взаимодействия называются *протоколами выработки общего ключа (авторы Диффи и Хелман)*.

#### Схема разделения секрета

Для многих практически важных приложений, связанных с запуском или активизацией критических процессов или определяющих порядок получения доступа к значимым данным, ответственное лицо должно ввести секретный ключ. Чтобы обезопасить процедуру принятия решения и не отдавать все на волю одного человека, являющегося обладателем ключа, используют метод разделения секрета.

Метод разделения секрета состоит в назначении определенной группы лиц, которая имеет право принимать решение. Каждый член группы владеет определенной долей секрета (точнее, специально выбранным набором данных), полная совокупность которых позволяет восстановить секретный ключ. При этом *схема разделения секрета* выбирается с таким условием, что для восстановления секретного ключа требуется обязательное присутствие всех членов группы, так как в случае отсутствия хотя бы одного из участников объединение долей оставшихся членов группы гарантированно не позволяет получить никакой информации о секретном ключе.

Таким образом, схема разделения секрета определяется двумя алгоритмами, удовлетворяющими сформулированному выше условию:

* первый алгоритм определяет порядок вычисления значений долей по заданному значению секретного ключа;
* второй алгоритм предназначен для восстановления значения секрета по известным долям.

Задачу построения схемы разделения секрета можно обобщить тремя способами:

* введением *структуры доступа,* когда решение может приниматься не одной, а несколькими различными группами, причем часть из участников может наделяться правом вето;
* добавлением механизмов, позволяющих обнаружить обман или сговор участников;
* введением специального протокола распределения долей между участниками с подтверждением правильности полученной информации и аутентификацией сторон.

#### 2.7. Общие требования к криптосистемам

Для современных криптографических систем защиты информации сформулированы следующие общепринятые *требования*:

1. зашифрованное сообщение должно поддаваться чтению только при наличии ключа;
2. число операций, необходимых для определения использованного ключа шифрования по фрагменту шифрованного сообщения и соответствующего ему открытого текста, должно быть не меньше общего числа возможных ключей;
3. число операций, необходимых для расшифрования информации путем перебора всевозможных ключей должно иметь строгую нижнюю оценку и выходить за пределы возможностей современных компьютеров (с учетом возможности использования сетевых вычислений);
4. знание алгоритма шифрования не должно влиять на надежность защиты;
5. незначительное изменение ключа должно приводить к существенному изменению вида зашифрованного сообщения;
6. если в текст в процессе шифрования вводятся дополнительные биты, то они должны быть полностью и надежно скрыты в шифрованном тексте;
7. не должно быть простых и легко устанавливаемых зависимостей между ключами, последовательно используемыми в процессе шифрования;
8. любой ключ из множества возможных ключей должен обеспечивать надежную защиту информации;
9. алгоритм должен допускать как программную, так и аппаратную реализацию.

##### Вопросы и задания для самостоятельной работы

1. Чем отличаются подходы к обеспечению безопасности информации в криптографии и в стеганографии?
2. Что общего и в чем отличие криптографического преобразования информации от кодирования ее при защите от случайных угроз безопасности?
3. Какими методами обеспечивается конфиденциальность информации?
4. Что такое целостность информации?
5. Дайте определение имитостойкости шифра.
6. Что такое имитовставка? Для каких целей она используется?
7. Для каких аспектов информационного взаимодействия необходима аутентификация?
8. Два основных требования к хэш-функциям. Против каких атак они направлены?
9. Какие средства используются для обеспечения невозможности отказа от авторства?
10. Что означает свойство односторонности криптографической хэшфункции?
11. В чем суть предварительного распределения ключей?
12. Что такое сертификат открытого ключа?
13. Для чего используется схема разделения секрета?

## ТЕМА 3. МАТЕМАТИЧЕСКИЕ ОСНОВЫ КРИПТОГРАФИИ. ЭЛЕМЕНТЫ ТЕОРИИ ЧИСЕЛ, АБСТРАКТНОЙ АЛГЕБРЫ И АЛГЕБРАИЧЕСКОЙ ГЕОИЕТРИИ

В криптографии сплелись несколько разделов математики: теория чисел, теория сложности, теория информации, теория вероятности, абстрактная алгебра, алгебраическая геометрия, формальный анализ. В силу присущей методам криптографии специфики, большой интерес представляет множество целых чисел и различные алгебраические структуры на его базе. Поэтому основное внимание будет уделено работе с целыми числами.

Большой вклад в развитие современных методов криптографии принадлежит *теории чисел*. Теория чисел занимается изучением свойств целых чисел. Целые числа – это числа натурального ряда (1, 2, 3... – положительные целые), нуль и отрицательные целые (-1,-2,-3...). Расположив целые числа в возрастающем порядке, получаем ряд, в котором разность между большим и меньшим соседними членами везде равна единице.

В криптографических методах используются только положительные целые числа, поэтому при изложении теоретического материала темы 3 ограничимся рассмотрением только *положительных целых чисел.* Их будем обозначать буквами латинского алфавита.

Алгебраические структуры, используемые в криптографии, такие как *группы, кольца и поля,* являются объектами *абстрактной алгебры*. Методы абстрактной алгебры необходимы для понимания основ современной криптогра-

фии, для изучения криптографических алгоритмов и умения оценивать их сильные и слабые стороны.

Основной объект изучения *алгебраической геометрии* – *алгебраические многообразия*, то есть геометрические объекты, заданные как множества решений систем алгебраических уравнений. В криптографии используются *абелевы многообразия*. Основной их пример – *эллиптические кривые*, имеющие очень обширную теорию. Они стали инструментом доказательства Великой теоремы Ферма. Плоские *эллиптические кривые над конечными полями* используются в эллиптической криптографии.

Для современной криптографии актуальна проблема повышения стойкости и уменьшения размера блоков данных путем модификации уже существующих криптосистем.

Самый очевидный путь решения вышеупомянутой проблемы – представление блоков информации в криптографических алгоритмах не только в виде чисел (или элементов конечных полей), но и в виде иных алгебраических объектов большей сложности, например *точек эллиптических кривых*.

Использование эллиптических кривых для криптографии было независимо предложено Нилом Коблицем (Neal Koblitz) и Виктором Миллером (Victor Miller) в середине 80-х годов прошлого века (Koblitz, 1987; Miller, 1985). На сегодняшний день это одно из самых перспективных направлений развития *криптографии с открытыми ключами*.

### 3.1. Простые числа и непрерывные дроби

Сумма *а + b*, разность *а – b* и произведение *ab* двух целых *а* и *b* являются *a*

также целыми. Но частное от деления *а* на *b* (если *b* не равно нулю) может *b*

быть как целым, так и не целым. *a*

В случае, когда частное от деления *а* на *b –* целое, обозначая его бук*b*

вой *q,* имеем *a = bq*, т.е*. а* представляется произведением *b* на целое. Говорим тогда, что *а* делится на *b* или, что *b* делит а. При этом *а* называем *кратным* числа *b*, a *b* – *делителем* числа *а*. То обстоятельство, что *b* является делителем числа *а*, записывается так: *b \ а.* П р и м е р ы:

Имеем: 21 = 7·3; 0 = 9·0; 85= 17·5.

Поэтому можем сказать: 21 делится на 7, 0 делится на 9, 85 делится на 17, или: 7 делит 21, 9 делит 0, 17 делит 85.

***Основные теоремы:***

1. *Если а кратно т, т кратно b, то а кратно b.*

Действительно, из *a**ma*1, *m**bm*1следует*a**ba*1 *m*1*.* Таким образом, *а* представляется произведением *b* на целое число *am* и тем самым делится на *b.*

1. *Если в равенстве вида k+l+…+n=p+q+…+s относительно всех членов, кроме какого-либо одного, известно, что они кратны b, то и этот один член кратен b.*

Действительно, пусть таким одним членом будет *k.* Имеем

*l**bl*1, …, *n**bn*1, *p**bp*1 ,*q**bq*1, …, *s**bs*1 *k p q*      ... *s l* ... *n b p q*( 1 1    ... *s l*1 1 ... *n*1 (1)

Таким образом, *k* представляется произведением *b* на целое число *p q*1 1  ... *s*1 1 *l*1 ... *n*1 и тем самым делится на *b.* 3. Теорема о делении с остатком.

*Всякое целое а представляется единственным способом с помощью положительного целого b равенством вида*

*a = bq + r;* 0*r* <*b*. (2)

Действительно, одно представление числа *а* равенством такого вида получим, взяв *bq* равным наибольшему кратному числа *b,* не превосходящему *а.* Допустив же существование представления числа *а* еще одним равенством того же вида:*a**bq*1*r*1 ;0 *r* <*b,* и вычитая почленно это последнее равенство из

предыдущего, получим

0   *b q q*( 1) *r r*1 (3)

Отсюда убедимся (теорема **2**), что разность *r r* 1кратна *b.* С другой стороны, легко видеть, что та же разность, как разность двух неотрицательных чисел, меньших *b,* сама будет численно меньше *b,* числом же, кратным *b* и численно меньшим *b,* является лишь число 0. Поэтому*r r* 10 *,* а отсюда и из равенства (1) будет следовать, что и *q q* 10 *.* Таким образом, второе представление числа *а* тождественно первому.

Число *q* называется *неполным частным,* а число *r* – *остатком* от деления *а* на *b.* Очевидно, что при *r=0* понятия «неполное частное» и «частное» совпадают.

П р и м е р ы.: Пусть b=14. Имеем

177 = 14·12 + 9, 0 < 9< 14,

154 = 14·11 + 0, 0 = 0 < 14.

#### Наибольший общий делитель (НОД)

В дальнейшем будем рассматривать лишь положительные делители чисел. Всякое целое, делящее одновременно целые *а, b, ..., l,* называется их *общим делителем.* Наибольший из общих делителей называется общим наибольшим делителем и обозначается символом *(а, b,* ..., *l*). Если *(а, b,* ..., *l)* =1, то *а, b,* ..., *l* называются *взаимно простыми.* Если каждое из чисел *а, b, ..., l* взаимно просто с каждым другим из них, то *а, b, ..., l* называются *попарно простыми.* Очевидно, числа попарно простые всегда и взаимно простые. В случае же двух чисел понятия «попарно простые» и «взаимно простые» совпадают.

Примеры. Числа 6, 10, 15, ввиду (6, 10, 15) = 1, – взаимно простые. Числа 8, 13, 21, ввиду (8, 13) = (8, 21) = (13, 21) = 1, – попарно простые.

1. *Если а кратно b, то совокупность общих делителей чисел а и b совпадает с совокупностью делителей одного b; в частности (a, b) = b.*

Действительно, всякий общий делитель чисел *а* и *b* является делителем и одного *b.* Обратно, раз *а* кратно *b,* то (1) всякий делитель числа *b* является также делителем числа *а,* т.е. является общим делителем чисел *b* и *а.* Таким образом, совокупность общих делителей чисел *а* и *b* совпадает с совокупностью делителей одного *b.* А так как наибольший делитель числа *b* есть само *b,* то *(а, b) = b.*

1. *Если*

*a = bq + c, (4)*

*то совокупность общих делителей чисел a u b совпадает с совокупностью общих делителей чисел b и с; в частности (а, b) = (b, c).*

Действительно, написанное равенство показывает, что всякий общий делитель чисел *а* и *b* делит также и *с* и, следовательно, является общим делителем чисел *b* и *c*. Обратно, то же равенство показывает, что всякий общий делитель чисел *b* и *с* делит *а* и, следовательно, является общим делителем чисел *а* и *b.* Таким образом, общие делители чисел *а* и *b* суть те же, что и общие делители чисел *b* и *с*; в частности, должны совпадать и наибольшие из этих делителей, т.е. *(a, b) = (b, с).*

Для разыскания общего наибольшего делителя, а также для вывода его важнейших свойств применяется *алгоритм Евклида:*

Пусть *а* и *b –* положительные целые и *а>b.* Согласно теореме 3 находим ряд равенств:

*a**bq*1 *r* 2 ,0 *r b*2

*b r* 2 *q*2 *r*3 ,0*r*3 *r* 2(5) *r r*2  3 *q*3 *r* 4 ,0*r* 4 *r*3

………………………

*rn*2*rn*1*qn*1*rn*,0 *rn*0 *rn*1, *rn*1*rnqn*,

заканчивающийся, когда получается некоторое*rn*10 . Последнее неизбежно, так как ряд *b r r*, ,2 3*,* ... как ряд убывающих целых не может содержать более чем *b* положительных.

Рассматривая равенства (2), идя сверху вниз, убеждаемся, что общие делители чисел *с* и *b* одинаковы с общими делителями чисел *b* и*r*2, далее одинаковы с общими делителями чисел *r*2 и*r*3*,* чисел *r*3и*r*4, ..., чисел *rn*1и *rn* , наконец, с делителями одного числа*rn ,* являющегося последним не равным нулю остатком алгоритма Евклида. Одновременно с этим имеем:

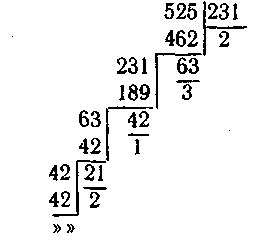
(*ab*, ) ( , *br*2) ( *r*2,*r*3) ... (  *rn*1,*rn*)*rn* (6)

и приходим к следующим результатам.

1. *Совокупность общих делителей чисел а и b совпадает с совокупностью делителей их общего наибольшего делителя.*
2. *Этот общий наибольший делитель равен последнему не равному нулю остатку алгоритма Евклида.*

П р и м е р. Применим алгоритм Евклида к отысканию (525, 231). Находим

(вспомогательные вычисления приведены слева)



525 = 231\*2 + 63,

231= 63\*3 + 42, 63 = 42\*1+21, 42 = 21\*2.

Здесь последний положительный остаток есть *r*421 . Значит,

(525, 231) = 21.

1. *Обозначая буквою т любое положительное целое, имеем (am, bm)=(a, b)m.*
2. *Обозначая буквою*  *любой общий делитель чисел а и b, имеем a b* (*ab*, ) *a b*

( , )  *; в* *частности, имеем* ( , ) 1, *т.е. частные от деле-*

  (*ab*, ) (*ab*, )

*ния двух чисел* *на их общий наибольший делитель суть числа взаимно простые.*

Действительно, умножив соотношения (1) почленно на *т,* получим новые соотношения, где вместо *abr*, , ,...,2 *rn* будут стоять *ambm*, ,*r m*2 ,...,*rn m .* Поэтому (*ambm*, )*rnm* и, таким образом, верно утверждение **8**.

Применяя утверждение **8**, находим:

#### a b a b

(*ab*, )  ( , )  ( , )**.** (7)

  

Отсюда следует утверждение **9.**

**10**. *Если (а, b)=1, то (ас, b) = (c, b).*

Действительно, *(ас, b)* делит *ас* и *bc,* значит, оно делит и *(ас, bc),* ввиду **8** равное *с.* Но *(ас; b)* делит и *b,* поэтому оно делит и *(с, b).* Обратно, *(с, b)* делит *ас* и *b,* поэтому оно делит и *(ас, b).* Таким образом, *(ас, b)* и *(с, b)* взаимно делят друг друга и, следовательно, равны между собой. **11.** *Если (а, b) =1* *и ас делится на b, то с делится на b.*

Действительно, в соответствии с теоремой **4**, при *ас,* делящемся на *b,* имеем *(ас, b) = b* и получаем *b =* *(с, b).* А этим и доказывается делимость *с* на *b.* **12.** *Если каждое a*1,*a*2,...,*am взаимно просто с каждым b*1, ,...,*b b*2 *n , то и произведение aa*1 2,...,*am взаимно просто с произведением bb*1 2,...,*bn.*

Действительно, согласно 1 находим

(*a*1*a a a*2 3... *m*,*bk*) ( *a*2*a a*3... *m*,*bk*) ( *a*3...*a bm k*) 1

и далее, полагая ради краткости *a*1,*a*2,...,*am = А,* точно таким же путем выводим

(*b*1, ,...,*b b*2 *n , A) = (b*2,*b b*3..., *n , A) = (b b*3..., *n A) = (bn , A) =* 1.

##### Общее наименьшее кратное

Всякое целое, кратное всех данных чисел, называется их *общим кратным.* Наименьшее положительное общее кратное называется *общим наименьшим кратным.* Будем рассматривать только общие кратные двух положительных чисел.

Пусть (*ab d a*, ) , *da*1,*b**db*1и, следовательно (согласно **9**), (*a*1,*b*1) 1 *.*

Пусть *М –* какое-либо общее кратное чисел *а* и *b*. Так как *М* кратно *а,* то *M=ak,* где *k* – целое. Но *М* кратно и *b.* Поэтому

*M ak a k*1

 (8)*b b b*1

должно быть целым и, следовательно (в соответствии с **11**), *к* должно делиться на *b*1*.* Поэтому *k bt* 1 *,* где *t –* целое, причем для *М* получается формула

M = *abt* . (9) *d*

Обратно, очевидно, что *М*, представляемое формулой (3) при любом целом *t,* будет общим кратным *а* и *b,* и, таким образом, формула (3) дает общий вид всех общих кратных чисел *a* и *b.*

Наименьшее положительное из этих общих кратных, т.е. общее наименьшее кратное, получаем при *t =* 1*.*

Оно будет

*m*  *ab* (10)

*d*

Теперь формулу (3) можно переписать так:

*M = mt.* (11)

Формулы (4) и (5) приводят к теоремам:

1. *Совокупность общих кратных двух чисел совпадает с совокупностью кратных их общего наименьшего кратного.*
2. *Это общее наименьшее кратное двух чисел равно их произведению, деленному на их общий наибольший делитель.*

##### Простые числа

Число 1 имеет только один положительный делитель, именно 1. В этом отношении число 1 в ряде натуральных чисел стоит совершенно особо. Всякое целое, большее 1, имеет не менее двух делителей, именно 1 и самого себя; если этими делителями исчерпываются все положительные делители целого числа, то оно называется *простым.* Целое, большее 1, имеющее кроме 1 и самого себя другие положительные делители, называется составным.

1. *Наименьший отличный от единицы делитель целого, большего единицы, есть число простое.*

Действительно, пусть *q* – наименьший отличный от 1 делитель целого *а,* большего 1. Если бы *q* было бы составным, то оно имело бы некоторый делитель *q*1с условием 1 *q q*1 причем число *а*, делясь на *q,* должно делиться на *q*1*.* А это противоречило бы нашему предположению относительно числа *q.*

1. *Наименьший отличный от единицы делитель составного числа а (согласно предыдущему утверждению* *он будет простым) не превосходит a* Действительно, пусть *q* – этот делитель, тогда *a**qa*1,*a*1 *q* ,откуда, перемножая и сокращая на *a*1, получим *a q* 2,*q a*. **17.** *Простых чисел бесконечно много.*

Справедливость этой теоремы следует из того, что каковы бы ни были различные простые *p p*1, 2,...,*pk* , можно получить новое простое, среди них не находящееся. Таковым будет простой делитель суммы *p p p*1 2... *k*1 , который, деля всю сумму, не может совпадать ни с одним из простых *p p*1, 2,...,*pk* .

Для составления таблицы простых чисел, не превосходящих данного целого *N,* существует простой способ, называемый *решетом Эратосфена.* Он состоит в следующем. Выписываем числа

*1, 2, .... N.* (12)

Первое большее 1 число этого ряда есть 2. Оно делится только на 1 и на самого себя и, следовательно, оно простое.

Вычеркиваем из ряда (12) (как составные) все числа, кратные 2, кроме самого

2. Первое следующее за 2 невычеркнутое число есть 3. Оно не делится на 2 (иначе оно оказалось бы вычеркнутым). Следовательно, 3 делится только на 1 и на самого себя, а потому оно также будет простым.

Вычеркиваем из ряда (6) все числа, кратные 3, кроме самого 3. Первое следующее за 3 невычеркнутое число есть 5. Оно не делится ни на 2, ни на 3 (иначе оно оказалось бы вычеркнутым). Следовательно, 5 делится только на 1 и на самого себя, а потому оно также будет простым.

И так далее.

Когда указанным способом уже вычеркнуты все числа, кратные простых, меньших простого *р,* то все невычеркнутые, меньшие *p*2*,* будут простые. Действительно, всякое составное *а,* меньшее *p*2, нами уже вычеркнуто, как кратное его наименьшего простого делителя, который  *a* *p .* Отсюда следует:

1. *Приступая к вычеркиванию кратных простого р, это вычеркивание следует начинать с* *p*2.
2. *Составление таблицы простых чисел, не превосходящих N, закончено, как только вычеркнуты все составные кратные простых, не превосходящих N .*

##### Единственность разложения на простые сомножители

*Всякое целое а или взаимно просто с данным простым р, или же делится на р.* Действительно, *(а, р),* будучи делителем *р,* может быть равно или *1*, или *р.* В первом случае *а* взаимно просто с *р,* во втором *а* делится на *р*.

*Если произведение нескольких сомножителей делится на данное простое р, то, по крайней мере, один из сомножителей делится на р.*

Действительно (а), каждый сомножитель или взаимно прост с *р,* или же делится на р. Если бы все сомножители были взаимно просты с *р,* то и их произведение (3, f, § 2) было бы взаимно просто с *р.* Поэтому хоть один сомножитель делится на *р.*

**22.** *Всякое целое, большее единицы, разлагается на произведение простых сомножителей и притом единственным способом (если отвлечься от порядка следования сомножителей).*

Действительно, пусть *а* – целое, большее 1; обозначая буквою *p*1его наименьший простой делитель, имеем *a p* 1*a*1 *.* Если *a*11 , то, обозначая буквою *р2* его наименьший простой делитель, имеем *a*1*p*2 *a*2 *.* Если *a*21 , то подобно этому находим *a*2 *p*3 *a*3 и т.д., пока не придем к какому-либо *an ,* равному 1. Тогда получим *an*1*pn .* Перемножив все найденные равенства и произведя сокращение, получим следующее разложение *а* на простые сомножители:

*a p p p p* 1 2 3... *n* . (13)

Допустим, что для того же самого *а* существует и второе разложение на простые сомножители*a q q q* 1 2 3...*qs* .

Тогда найдем

*p p p*1 2 3...*pn* *q q q*1 2 3...*qs*. (14)

Правая часть этого равенства делится на *q*1*.* Следовательно (*b*), по крайней мере один из сомножителей левой части должен делиться на *q*1*.* Пусть, например, *p*1делится на *q*1(порядок следования сомножителей в нашем распоряжении);

тогда найдем *p q*1 1 *( p*1 кроме 1 делится только на *p*1 ). Сократив обе части равенства на *p q*1 1*,* получим *p p p*2 3... *n* *q q q*2 3... *s*.

Повторив прежние рассуждения применительно к этому равенству, получим *p p qq*3... *n* 3 *s*и т.д., пока, наконец, в одной части равенства, например в левой, не сократятся все сомножители. Но одновременно должны сократиться и все сомножители правой части, так как равенство 1*qn*1...*qs* при *qn*1, ...,*qs*, превосходящих 1, невозможно. Таким образом, второе разложение на простые сомножители тождественно первому.

**23.** В разложении числа *а* на простые сомножители некоторые из них могут повторяться. Обозначая буквами *p p*1, 2,...,*pk* различные из них и буквами 1, 2,..., *k* кратности их вхождения в *а,* получим так называемое *каноническое разложение* числа *а* на сомножители:

##### a p p p  11 22... kk . (15)

Пример. Каноническое разложение числа 588 000 будет: 588000 2\*3\*5\*7 5 3 2.

*Пустьa p p p*  11 22... *kk* – *каноническое разложение числа а. Тогда все делители*

*числа суть все числа вида*

*d**p p*1 1 22...*p**kk*(16)

011,022,0*k**k* .

Действительно, пусть *d* делит *а.* Тогда (b, § 1) *а = dq* и, следовательно, все простые делители числа *d* входят в каноническое разложение числа *а* с показателями, не меньшими тех, с которыми они входят в каноническое разложение числа *d.* Поэтому *d* имеет вид (16). Обратно, всякое *d* вида (16) делит *а.*

П р и м е р. Все делители числа 720 2 \*3 \*5 4 2 получим, если в выражении

213 5 2 3 заставим 1, 2, 3, независимо друг от друга пробегать значения

10,1,3,4; 20,1; 30,1. Поэтому указанные делители будут: 1, 2, 4, 8, 16, 3, 6, 12, 24, 48, 9, 18, 36, 72, 144, 5, 10, 20, 40, 80, 15, 30, 60, 120, 240, 45, 90, 180, 360, 720.

1. *Общий наибольший делитель нескольких чисел является произведением степеней вида p**, где р – общий простой делитель всех этих чисел, а* *– наименьший из показателей, с которыми р входит в их канонические разложения.*
2. *Совокупность общих делителей нескольких чисел совпадает с совокупностью делителей их общего наибольшего делителя.*

Действительно, пусть *d –* общий делитель чисел *а,..., l.* Тогда имеют место равенства вида *a**da*1,...,*l**dl*1, (которые показывают, что: а) всякий простой делитель *р* числа *d* должен быть делителем и каждого из чисел *a,* ..., *l*, а также что: b) этот делитель *р* должен входить в каноническое разложение числа *d* с показателем, не превосходящим наименьшего из тех, с которыми он входит в канонические разложения чисел *а,* ..., *l;* обратно, каждое *d,* подчиненное условиям а) и b), очевидно, является общим делителем чисел

*а, ..., l.*

Общим наибольшим делителем, т.е. наибольшим из общих делителей (а, § 2), является тот из последних, в каноническом разложении которого показатели степеней простых чисел точно равны наименьшим из тех, с какими эти простые числа входят в канонические разложения чисел *а, ..., l.*

А всякий общий делитель, как имеющий в своем каноническом разложении все показатели не превосходящими соответствующих показателей в каноническом разложении общего наибольшего делителя, будет делителем последнего.

П р и м е р. Общий наибольший делитель чисел 6791400 2 \*3 \*5 \*7 \*3 2 2 3 11, 178500 2 \*3\*5 \*7\*3 3 17, 27720  2 \*3 \*5\*7\*3 3 11 равен 2 \*3\*5\*72  420.

1. *Общее наименьшее кратное нескольких чисел является произведением степеней вида p**, где р* – *простой делитель по меньшей мере одного из этих чисел, а* *– наибольший из показателей, с которыми р входит в их канонические разложения.*
2. *Общее наименьшее кратное нескольких попарно простых чисел равно их произведению.*
3. *Совокупность общих кратных нескольких чисел совпадает с совокупностью кратных их общего наименьшего кратного.*

Действительно, пусть *М* – общее кратное чисел 0, *..., l*. Тогда имеют место равенства вида *M=ad',* ..., *М =ll’*, которые показывают, что: а) всякий простой делитель *р* каждого из чисел *а*, ..., *l* должен быть делителем и числа М, а также что: b) этот делитель *р* должен входить в каноническое разложение числа *М* с показателем, не меньшим наибольшего из тех, с которыми он входит в канонические разложения чисел *а*, ..., *l*; обратно, каждое *М*, подчиненное условиям а) и b), очевидно, является общим кратным чисел *а*, ..., *l.*

П р и м е р. Общее наименьшее кратное чисел 1800 = 23 **.** 32 **.**52, 3780 = 22 **.** 33 **.** 5 **.** 7, 8910 = 2 3 4  5 11 равно 23 **.** З4 **.** 52 **.** 7 **.** 11 = 1 247 400.

##### Непрерывные дроби и их связь с алгоритмом Евклида

**30.** Пусть  – любое вещественное число. Обозначим буквой *q1* наибольшее целое число, не превосходящее .

При нецелом  имеем  *q*1  1 ; 2 1. Точно так же при нецелых  2,..., 5 1

2

имеем 1

2  *q*2  ; 3 1,

3

. . . . . . . . . . . . . . .

1

*S*1  *qS*1  ; *S* 1, (17) *S*

ввиду чего получаем следующее *разложение*  *в непрерывную дробь:* 1

 *q*1

2

1



*q*

*q*3 ...

**.**

**.**

**.**

1

 (18)

*q*

1

*S*1 

*S*

Если  – иррациональное, то и всякое 1 – иррациональное (при рациональном

*S* ввиду (18) рациональным оказалось бы и ) и указанный процесс может быть неограниченно продолжен.

Если же  – рациональное и, следовательно, может быть представлено рацио*a*

нальной несократимой дробью  с положительным знаменателем, то ука*b*

занный процесс будет конечен и может быть выполнен с помощью алгоритма Евклида. Действительно, имеем:

1. *bq* 1  *r*2; *a*  *q*  *b*1 , *b*

2

1

*r*

1

*b*

1.  *r q*2 2  *r*3; *r*  *q*2 *r*2

2 *r*3

. . . . . . . . . . . . . . . . . . . . . . . . . . .

*rn* 2  *r qn* 1 *n* 1 *rn*; 

*   *rrn*2 *qn*1  *rn*11 ,

*n* 1 *rn*

*rn* 1  *r qn* 1 *n* 1 *rn* ; 

*   *rrn*2 *qn*1  *rn*11 ,

*n* 1

*rn*

*rn*1 *r qn n* ; *rn*1 *qn*,

*rn a* 1

 *q*

*b q*2 

1

1

*q*3 ...

**.**

**.**

**.**

 1 (19)

*qn*1 

1

*q*13

Числа *q q*1, 2,..., участвующие в разложении числа в непрерывную дробь, называются *неполными частными* (в случае рационального  это будут, согласно *b* , неполные частные последовательных делений алгоритма Евклида), дроби же

1  *q*1,2  *q*1 1 ,  *q*1 1 , . . . (20)*q*2

2

1

*q*



*q*3

называются *подходящими дробями.*

Простой *закон вычисления подходящих дробей* получим, заметив, что *S* (*s* 1) получается из *S*1заменой в буквенном выражении для *S*1 числа *qS*1 числом

1

*qS*1  . Действительно, полагая ради единообразия *Р0=*1*,* Qo = 0, мы можем

*qS*

последовательно представить подходящие дроби в следующем виде (здесь ра-

1. *PS* пишем, желая обозначить *А* символом *РS*, а *В* –символом *QS*):

венство 

1. *QS*

*q*1 *P*1 ,

1  

1` *Q*1

1 *q*1 

2  *q*2  *q q*2 1 1  *q P*2 1  *P*0  *P*2 ,

1 *q*2  1 0 *q Q*2 1  *Q*0 *Q*2

1

(*q*2  )*P P*1  0

3  (*q*2  *q*13)*Q Q*1  0  *q Q Qq P P*33 22  11 *QP*33

*q*3

и т.д., и вообще при *S* 1

*S*  *q PS S*1 *PS*2  *PS* (21)

*q QS S*1 *QS*2 *QS*

Таким образом, числители и знаменатели подходящих дробей мы можем последовательно вычислять по формулам:

*PS* *q PS S*1 *PS*2

*QS* *q QS S*1 *QS*2 (22)

Эти вычисления удобно делать по следующей схеме (два последних столбца пишем лишь в случае, когда  – несократимая дробь с положительным зна*a*

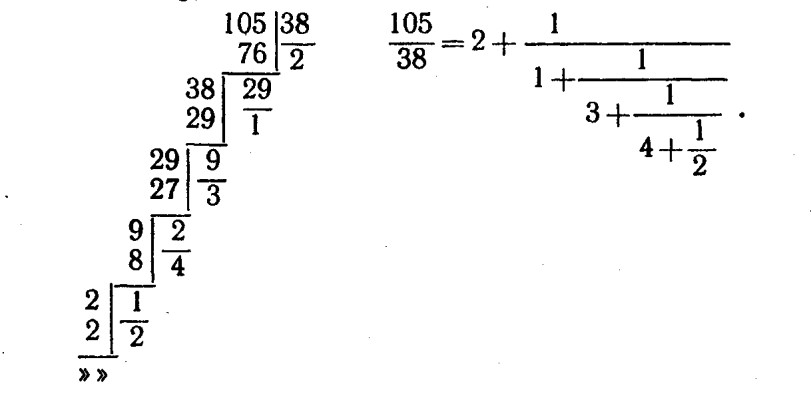
менателем:  ):

*b*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | *q*1 | *q*2 | … |  |  |  | … | *qn*1 | *qn* |



П р и м е р . Разложим в непрерывную дробь несократимую дробь . Здесь имеем



Поэтому указанная выше схема дает

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *qS* |  | 2 | 1 | 3 | 4 | 2 |
| *PS* | 1 | 2 | 3 | 11 | 47 | 105 |
| *QS* | 0 | 1 | 1 | 4 | 17 | 38 |

*При s>0 имеем PQS S*1 *Q PS S*1  ( 1)*S*

( 1) *S*

*При S>1 имеем*  *S*  *S*1 

*Q QS S*1

Действительно, приняв обозначение h3= P*s* Q *s*1– Q*s* P *s*1, мы при s=1 получим h1= q1 **.** 0 – 1**.**1=-1, a при s=1 с помощью равенства (3) найдем h*s* = -h*s*1. Отсюда получим h *s* =(-1) *s* . Пользуясь же этим равенством при s >1, легко найдем

*Ps* – *Ps* 1 = *hs* = (-1)S (23)

*s* –*s*1= 

*Qs Qs*1 *Q Qs s*1 *Q Qs s*1

Пусть 1<s, а если – рациональная несократимая дробь = *a* с положительным *b*

знаменателем, то пусть также s<n. Тогда  лежит между *s*1 и *s* , причем ближе к *s* , нежели к *s*1. Действительно, заменив в равенстве (2) число q*s* числом q*s* + 1 , получим

*s*1

= *s*1*Ps*  *Ps*1 ,

*s*1*Qs*  *Qs*1

*s*1Q*s* + Q *s*1- *s*1P *s* – P *s*1=0,

*s*1 Q *s* (– *Ps* )+ Q *s*1(– *Ps*1 ) =0, (24)

*Qs Qs*1

откуда убеждаемся, что первая из разностей, стоящих в скобках, и по знаку противоположна второй, и численно (ввиду Q *s* > Q *s*1) меньше последней. А этим и доказываются наши утверждения.

### 3.2. Мультипликативные функции

*Функция* *(а) называется мультипликативной, если она удовлетворяет двум следующим условиям:*

1. *Эта функция определена для всех целых положительных а и не равна нулю, по меньшей мере, при одном таком а.*
2. *Для любых положительных взаимно простых a1 и а2 имеем:*

(*aa*1 2)  (*a*1)  *a*2(25)

П р и м е р.Нетрудно видеть, что мультипликативной является функция *aS,* где s – любое вещественное или комплексное число.

**31.** *Для всякой мультипликативной функции*  *(а) имеем*  (1) = 1. Действительно, пусть (*ао*) не равно нулю. Находим

(*a*0) (*a*0 1)  (*a*0) (1), 1(1)(26)

Свойство **32**мультипликативной функции  (*a*) распространяется и на случай *k* > 2 попарно простых чисел *а1, аг, а3,* ..., *ak.* Действительно, имеем:

(*a a a*1 2 3...*aK* )  (*a*1) (*a a*2 3...*aK* )   (*a*1) (*a*2) (*a*3...*aK* ) ...   (*a*1) (*a*2) (*a*3)... (*aK* ). В частности, находим

(*p p p*1  2 3 ...*pK**K* )   (*p*11 ) (*p*22 ) (*p*33 )... ( *pK**K* ) (26)

**33.** Обратно, *мы всегда построим некоторую мультипликативную функцию*  (*a*), *если положив*  (1) = 1 *и назначив произвольно значения для* (*Р*), *отвечающих положительным степеням простых чисел, в общем случае определим эту функцию равенством* (26).

Действительно, *a*  *p*1 ...*pk k* , если представлено в виде произведения *aa*1 2двух взаимно простых чисел *а1* и *аг,* то справедливо тождество

  ( )*a*  (*a*1) (*a*2) , (27)

левая часть которого является произведением чисел (*pS**S* ), отвечающих всем сомножителям вида *pS**S* числа *а,* а правая часть является тем же произведением, но разбитым на два взаимно простых произведения, одно из которых (*a*1) является произведением чисел (*pS**S* ), отвечающих всем сомножителям вида *pS**S* числа *a*1, другое же (*a*2) является произведением чисел (*pS**S* ), отвечаю-

щих всем сомножителям видам *pS**S* числа *a*2.

П р и м е р.Мультипликативную функцию можно построить, взяв  (1) =1 и  (р) = 2, если  > 0. Тогда при *k* > 0 будем иметь (*p*11...*pK**K* )  2*K* . В частности, найдем: (1) 1, (2)  2, (3)  2

(28)

(4)  2, (5)  2, (6)  4

**34.** *Произведение*  *(а)* = 1 *(а)* 2 *(а) двух мультипликативных функций* 1*(а) и* 2 *(а) также является мультипликативной функцией.*  Действительно, имеем   (1)  1(1) 2(1) 1 .

Кроме того, при (*a a*1, 2) 1 находим

(*a a*1 2)  1 1 2(*a a* )2(*a a*1 2)  1 1 1(*a* ) (*a*2)2(*a*1)2(*a*2)  1 1(*a* )2(*a*1 1) (*a*2)2(*a*2)   (*a*1) ( *a*2)

Доказанная теорема обобщается и на случай любого числа *к* > 2 мультипликативных функций:

1(*a*), 2(*a*), 3(*a*),...,*K* ( )*a* (29)

Действительно, пользуясь ею последовательно, убедимся в мультипликативности произведений:

  1( )*a* 2 ( )*a* 3 ( )*a*  (  1( )*a* 2(*a*)) 3 (*a*),

   1( )*a* 2 ( )*a* 3 ( )*a* 4 ( )*a*  (   1( )*a* 2 ( )*a* 3 (*a*)) 4 (*a*), (30)

............................................................... .....

   1( )*a* 2 ( )...*a K*1( )*a K* ( )*a*  (  1( )*a* 2 ( )...*a K*1(*a*))*K* ( )*a*

**35.** *Пусть*( )*a* – *мультипликативная функция и а = p*1 ...*pk k* – *каноническое разложение числа а. Тогда, обозначая символом*  *сумму, распространенную*

*d a*

*на все делители d числа a, будем иметь:*

( )*d*  (1 (*p*1)  (*p*12 ) ...(*p*11 ))...(1(*pK* ) (*p*2*k* ) ... (*pK**K* )) *(31)*

*d a*\

(В случае *a=1 правая часть считается равной 1.)*

Чтобы доказать, что это тождество, раскроем скобки в его правой части. Тогда получим сумму всех (без пропусков и повторений) слагаемых вида

 (*p*11 ) (*p*22 )... ( *pK**K* ) (*p p*1 1 2 2 ...*pK**K* )

0   1 1, 0   2 2,...,0  *K*  *K* (32)

А это (теорема **24**) как раз и будет то, что стоит в левой части тождества.

#### Число делителей и суммы делителей

При ( )*a* 1 тождество **35** примет вид ( )*a*  (1 1)...(*K* 1), где ( )*a* – число делителей числа *a.*

П р и м е р.(720) (24 32 5)  (41)(21)(1 1)  30.

( )*a – мультипликативная функция, для которой при* 0 *имеем*

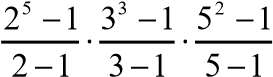
(*p*) 1(33)

Это следует из найденной для ( )*a* формулы и теоремы **33.**

При ( )*a*  *a* тождество **33** примет вид

*S a*( )  *p*111 1... *pK**K*1 1 (34) *p*1 1 *pK* 1

где S(a) – сумма делителей числа *а*.

П р и м е р.*S*(720)  *S*(24 32 5)    2418

*S (a) – мультипликативная функция, для которой при* 0 *имеем*

 *p*1 1

*S p*( )  *.* (35) *p*1

Это следует из найденной для S (a) формулы и теоремы **33.**

#### Функция Мебиуса

**36.** *Функция Мебиуса – мультипликативная функция, определенная равенствами:*( )*p*  1, (*p*)  0*, если* 1.

Из этого определения, в частности, следует, что:

1. Если в каноническом разложении *a p* 1 ...*pk k* числа *а* по меньшей мере один из показателей *a p p* 1... *K* превосходит l (если *а* делится на квадрат, отличный от l), то имеем ( )*a*  0.
2. В противном случае, т.е. в случае, если каноническое разложение числа *а* имеет вид *a p* 1 ...*pk* , имеем ( )*a*  ( 1)*K* .

|  |  |  |
| --- | --- | --- |
| П р и м е р ы. |  |  |
| (1) 1  (2) 1  (3) 1 (4)  0 | (5) 1  (6) 1  (7) 1  (8)  0 | (9)  0 (10) 1    (11) 1  (12)  0 |

**37.** *Пусть*  *(а) – мультипликативная функция и a*  *p*11 ...*pK**K – каноническое разложение числа а. Тогда имеем:*

 ( ) ( )*d d*  (1 (*p*1 ))...(1(*pK* ))(36)

*a*

*b*

(в случае *а=*1правую часть считаем равной 1).

Действительно, функция 1( )*a*  ( ) ( )*a a* , как произведение мультипликативных функций ( )*a* и ( )*a* , сама является мультипликативной функцией. Применяя к ней тождество f, § 2 и имея в виду, что 1( )*p* ( )*p* и что 1(*p*)  0, если

1, мы и убедимся в справедливости нашего утверждения.

В частности, полагая ( )*a* 1, из ) получим

( )*d*  0, *если a* 1

*d* ( )*d*  0, *если a* 1 (37)

*a*

1. Полагая же ( )*a*  , *получим*

*a*

1

( )*d* 1 1

 (1 )...(1 ), *еслиa* 1,

*d p*1 *pK*  (38)

*d* ( )*d*

*a* 1, *еслиa* 1 *d*

1. *Пусть целым положительным*    1,..., *n* *отвечают любые вещественные или комплексные f*  *f*1,..., *fn. Тогда, обозначая символом S' сумму значений f, отвечающих значениям* , *равным 1*, *и символом Sd сумму значений f, отвечающих значениям* , *кратным d, будем иметь*

*S* ,  ( )*d Sd ,* (39)

*d*

*где d пробегает целые положительные числа, делящие хотя бы одно значение* *.*

Действительно (2, b), имеем

*S* ,  *f*1 ( )*d* ... *fn* ( )*d* . (40)

*d*1 *d**n*

Собирая же вместе члены с одними и теми же значениями *d* и вынося при этом ( )*d* за скобки, в скобках получим сумму тех и только тех значений *f*, которые отвечают значениям , кратным *d,* т.е. как раз и получим сумму *Sd.*

#### Функция Эйлера

1. *Функция Эйлера* *(a) определяется для всех целых положительных а и представляет собою число чисел ряда*

*0, 1, …, а–1,* (41)

*взаимно простых с а.*  П р и м е р ы.

(1) = 1, (4) = 2,

(2) = 1, (5) = 4, (3) = 2, (6) = 2.

1. *Пусть*

*a= p*11 *p*22 *… p* *k k* (42)

* *каноническое разложение числа а. Тогда имеем*

(а)=а (1**-** *p*1 **)** (1**-** *p*1 **)…** (1**-** *p*1 **),** (43)

1 2 *k*

*или также*

1 11 2 21 *k*

|  |  |
| --- | --- |
| *В частности, будем иметь* |  |
|   1  *(р )=р - р ,* *(p)=p-1.* | (45) |

(а) = (p1 - p1 )(p 2 - p 2 )… (p *k*

* p*k k* *k* ). (44)

Действительно, применим теорему **39**. При этом числа  и числа *f* определим так: пусть ***x*** пробегает числа ряда (1); каждому значению ***x*** приведем в соответствие число = (***x,*** *а*) и число f=1.

Тогда S ' обратится в число значений = (***x,*** *а*), равных 1, т.е. в (a). А *S d* обратится в число значений = (***x,*** *а*), кратных *d.* Но (***x,*** *а*) может быть кратным *d* лишь при условии, что *d* – делитель числа *a.* При наличии этого же условия *a*

*S d* обратится в число значений *x,* кратных *d,* т.е. в *d* . Поэтому

(а)=(d) *da* , (46)

*d a*\

откуда (ввиду **38**) следует формула (43), а из последней (ввиду 42) следует формула (44). П р и м е р ы.

(60) = 60(1–  )(1–  )(1–  ) = 16, (81) = 81–27=54,

(5) = 5–1=4.

1. *(a) – мультипликативная функция, для которой при*  *> 0 имеем* *(р**)= р**– р*1 *.*

Это следует из формулы (44) и теоремы **33**.

1. *Имеем*

*(d)=а.* (47)

*d a*\

В справедливости этой формулы убедимся, применяя тождество **35**, которое при (а)=(a) дает

(а)=(1+(р1)+ … +(p11 ))…(1+(р*k* )+…+(р*k k* )). (48)

*d a*\

Ввиду (5) правая часть окажется равной

(1+(p1 – 1)+ … + (p11 – p111 ))…(1+(p *k* – 1)+…+(p*k k* – p*k k* *k* )), (49)

что после приведения в каждой большой скобке подобных членов обратится в

p11 … p *k k* = а. (50)

П р и м е р. Полагая а=12, находим

(1)+ (2) +(3) +(4)+ (6)+ (12)=1+1+2+2+2+4=12. (51)

### 3.3. Сравнение целых чисел по модулю

Рассмотрим целые числа в связи с остатками от деления их на данное целое положительное *т,* которое называется *модулем.*

Каждому целому числу отвечает определенный остаток от деления его на *т* (теорема **3**).

1. Если двум целым *а* и *b* отвечает один и тот же остаток **r**, то они называются *равноостаточными по модулю* *т* или *сравнимыми по модулю т.* Сравнимость чисел *а* и *b* по модулю *т* записывается так:

*a**b(*mod *m*), (52)

или так:

*а* сравнимо с *b* по модулю *т.*

1. *Сравнимость чисел а и b по модулю т равносильна:* 1) в*озможности представить а в виде а = b+тt, где t* – *целое;*

2) д*елимости а – b на m*.

Действительно, из *a**b(*mod *m*) следует

a=mq+r, b=mq1+ r; 0r<*m*,

откуда a – b=m(q – q1), *а = b+тt, t= q –* *q*1.

Обратно, из *а = b+тt, представляя b в виде*

b=mq1+ r, 0r<*m,*

выводим a=mq+r; q= q1+t, т.е. *a**b(*mod *m*).

Поэтому верно утверждение 1.

Из 1 непосредственно следует утверждение 2

#### Свойства сравнений, подобные свойствам равенств

1. *Два числа, сравнимые с третьим, сравнимы между собою.*

Следует из **44**.

1. *Сравнения можно почленно складывать.*  Действительно, пусть

a1  b1*(*mod *m*), a2  b2 *(*mod *m*), …, a*k*  b*k (*mod *m*), (53)

Тогда (из **45**)

a1=b1 + mt1, a2 *=* b2 *+* mt2 , …, a*k*  b*k* + mt *k* , (54)

откуда a1 + a2 +…+ a*k* b1 + b2 +…+ b*k* + m (t1 + t2 +…+ t*k* )

или (из **45**)

a1 + a2 +…+ a*k* b1 + b2 +…+ b*k* + (mod *m*).

*Слагаемое, стоящее в какой-либо части сравнения, можно переносить в другую часть, переменив знак на обратный.*

Действительно, складывая сравнение *a+b**c*(mod *m*) с очевидным сравнением -*b**-b(*mod *m*), получим а *c – b*(mod *m*).

*К каждой части сравнения можно прибавить любое число, кратное модуля.* Действительно, складывая сравнение *a**b*(mod *m*), с очевидным сравнением *mk*0(*mod*), получим *a+mk**b*(mod *m*).

1. *Сравнения можно почленно перемножать.*

Действительно, рассмотрим снова сравнения (1) и вытекающие из них равенства (2). Перемножая почленно равенства (2), получим

a1a2 … a*k* = b1b2 … b*k* + mN,

где N – целое. Следовательно (**1, с, § 1**),

a1a2 … a*k*  b1b2 … b*k* + (mod m),

Обе части сравнения можно возвести в одну и ту же степень.

Действительно, перемножив сравнение *a**b(*mod *m*) с очевидным сравнением k*k(*mod *m*), получим *ak**bk (*mod *m*).

1. Свойства **47** и **48** (сложение и умножение сравнений) обобщаются следующей теоремой.

*Если в выражении многочлена с целыми коэффициентами*

*S=*  *A*1*, …,**k x*11 *… x**k k заменим A*1*, …,**k , x*1*,…, x k числами В*1*, …,**k , y*1*,…, y k , сравнимыми с прежними по модулю m, то новое выражение S будет сравнимо с прежним по модулю m.*

Действительно, из

A1, …,*k*  В1, …,*k* (mod m), x1  y1(mod m), …, x *k*  y*k* (mod m)

находим (с) x11  y11 (mod m), …, x*k k* y *k k* (mod m),

A1, …,*k* x11 … x*k k*  В1, …,*k* y11 … y*k k* (mod m),

откуда, суммируя, получим

 A1, …,*k* x11 … x*k k*  В1, …,*k* y11 … y*k k* (mod m).

Если

ab(mod m), a1  b1 (mod m), …, a *n*  b *n* (mod m), xx1(mod m),

то

ax *n* + a1x*n*1+…+ a*n*  bx *n* + b1x*n*1+…+ b*n* (mod m).

Это утверждение есть частный случай предыдущего.

1. *Обе части сравнения можно разделить на их общий делитель, если последний взаимно прост с модулем.*

Действительно, из *a**b(*mod *m*), a=a1d, b= b1d, (d, m)=1 следует, что разность a–b, равная (a1–b1)d, делится на m. Поэтому (из **11**) a1–b1 делится на m, т.е. a1  b1(mod m).

**Особые свойства сравнений**

1. *Обе части сравнения и модуль можно умножить на одно и то же целое.* Действительно, из *a**b(*mod *m*) следует:

*а = b + тt, ak = bk + mkt* (55)

и, следовательно, *ak*  *bk(*mod *mk*).

1. *Обе части сравнения и модуль можно разделить на любой их общий делитель.*

Действительно, пусть *a*  *b(*mod *m*), a = a1d, b = b1d, m = m1d.

Имеем a = b+mt, a1d = b1d,+m1dt, a1= b1+ m1t

и, следовательно, а1 = b1*(*mod *m*1).

1. *Если сравнение а*  *b имеет место по нескольким модулям, то оно имеет место и по модулю, равному общему наименьшему кратному этих модулей.* В самом деле, из *a**b(*mod *m*1), *a**b(*mod *m*2 ), …, *a**b(*mod *mk* )следует, что разность *а–b* делится на все модули *m*1*, m*2 *,…, mk .* Поэтому (из **29**) она должна делиться и на общее наименьшее кратное *т* этих модулей, т.е.

*a*  *b(*mod*).*

**54*.*** *Если сравнение имеет место по модулю т, то оно имеет место и по модулю d, равному любому делителю числа т.*

В самом дело, из *a*  *b(*mod *m*) следует, что разность a – b должна делиться на m; поэтому (из **1**) она должна делиться на любой делитель *d* числа *т,* т.е.

*a*  *b(*mod *d*).

1. *Если одна часть сравнения и модуль делятся на какое-либо число, то и другая часть сравнения должна делиться на то же число.*

Действительно, из *a*  *b(*mod *m*) следует *а = b + mt;* если *а* и *m* кратны *d,* то

**(**из теоремы **2)** и *b* должно быть кратным *d,* что и утверждалось.

1. *Если a*  *b(*mod *m*), то *(a, m)= (b, m).*

Действительно, ввиду теоремы **2** это равенство непосредственно следует из *а = b + тt.*

#### Полная система вычетов

1. *Числа равноостаточные, или, что то же самое, сравнимые по модулю т, образуют класс чисел по модулю т.*

Из такого определения следует, что всем числам класса отвечает один и тот же остаток *r,* и мы получим все числа класса, если в форме *mq+r* заставим *q* пробегать все целые числа.

Соответственно, *т* различным значениям *r* имеем *т* классов чисел по модулю *т.*

1. Любое число класса называется *вычетом по модулю т* поотношению ко всем числам того же класса. Вычет, получаемый при q = 0, равный самому остатку r*, называется* *наименьшим неотрицательным вычетом.*

Вычет , самый малый по абсолютной величине, называется *абсолютно наименьшим вычетом.*

Очевидно при r< *m*2 имеем =r, при r> *m*2 – имеем =r – m;наконец, если *т* четное и r= *m*2 , то за  можно принять любое из двух чисел *m*2 и *m*2 –m = - *m*2 .

Взяв от каждого класса по одному вычету, получим *полную систему вычетов по модулю т.* Чаще всего в качестве полной системы вычетов употребляют наименьшие неотрицательные вычеты 0, 1, ..., m – 1или также абсолютно наименьшие вычеты; последние, как это следует из вышеизложенного, в случае нечетною *т* представляются рядом

* + *m*21 , …, -1, 0, 1, …, *m*21 ,

а в случае четного *т* каким-либо из двух рядов:

* + *m*2 +1, …, -1, 0, 1, …, *m*2 ,
  + *m*2 , …, -1, 0, 1, …, *m*2 -1.

1. *Любые т чисел, попарно несравнимые по модулю т, образуют полную систему вычетов по этому модулю.*

Действительно, будучи несравнимы, эти числа тем самым принадлежат к различным классам, а так как их *т,* т.е. столько же, сколько и классов, то в каждый класс наверно попадет по одному числу.

1. *Если (a, m)=1 и* ***х*** *пробегает полную систему вычетов по модулю т, то ах+b, где b –любое целое, тоже пробегает полную систему вычетов по модулю т.*

Действительно, чисел *ax+b* будет столько же, сколько и чисел ***х****,* т.е. ***m***. Согласно ***с*** остается, следовательно, только показать, что любые два числа *ax*1*+b* и *ax*2 *+b*, отвечающие несравнимым ***x***1 и ***x***2 , будут сами несравнимы по модулю *т.*

Но допустив, что *ax*1*+b*  *ax*2 *+b(*mod *m*),мы придем к сравнению *ax*1  *ax*2 *(*mod *m*), откуда, вследствие *(a, m)=1,* получим *x*1  *x*2 *(*mod *m*)*,* что противоречит предположению о несравнимости чисел *x*1и *x*2 .

#### Приведенная система вычетов

1. Согласно свойству **56** числа одного и того же класса по модулю *т* имеют с модулем один и тот же общий наибольший делитель. Особенно важны классы, для которых этот делитель равен единице, те. классы, содержащие числа, *взаимно простые с модулем*.

Взяв от каждого такого класса по одному вычету, получим *приведенную систему вычетов по модулю т.* Приведенную систему вычетов, следовательно, можно составить из чисел полной системы, взаимно простых с модулем. Обыкновенно приведенную систему вычетов выделяют из системы наименьших неотрицательных вычетов: 0, 1, ..., *т*–1. Так как среди этих чисел число взаимно простых с *т* есть (m), то число чисел приведенной системы, равно как и число классов, содержащих числа, взаимно простые с модулем, есть (m). П р и м е р. Приведенная система вычетов по модулю 42 будет 1, 5, 11, 13, 17, 19, 23, 25, 29, 31, 37, 41.

1. *Любые* (m) *чисел, попарно несравнимые по модулю т и взаимно простые с модулем, образуют приведенную систему вычетов по модулю m*.

Действительно, будучи несравнимыми и взаимно простыми с модулем, эти числа тем самым принадлежат к различным классам, содержащим числа, взаимно простые с модулем, а так как их (m)*,* т.е. столько же, сколько и классов указанного вида, то в каждый класс наверно попадет по одному числу.

1. *Если (а, т) = 1 и х пробегает приведенную систему вычетов по модулю m, то ах тоже пробегает приведенную систему вычетов по модулю т.*

Действительно, чисел *ах* будет столько же, сколько и чисел *х,* т.е. (m).

Согласно b остается, следовательно, только показать, что числа *ах* по модулю *т* несравнимы и взаимно просты с модулем. Но первое доказано в **60** для чисел более общего вида *ах +b*, второе же следует из *(а, т) = 1, (х,m)= 1.*

#### Теоремы Эйлера и Ферма

**64**. *При т* > 1 и *(а, т) =* 1 *имеем (теорема Эйлера):*

a( )*m*  *1(mod m).*

Действительно, если ***х***пробегает приведенную систему вычетов

*x=r*1*, r*2 *, …, rc ; c=**(m),*

составленную из наименьших неотрицательных вычетов, то наименьшие отрицательные вычеты 1, 2, …, *c* чисел *ax* будут пробегать ту же систему, но расположенную, вообще говоря, в ином порядке (согласно **63**). Перемножая почленно сравнения

ar1 1(mod m), ar2 2(mod m), …, ar*c* *c* (mod m),

получим

a*c* r1r2 …r*c* 1 2 …*c* (mod m),

откуда, деля обе части на произведение r1r2 …r*c* = 1 2 …*c* , получим a*c* *1 (mod m).*

***65.*** *При**p простом и а, не делящимся на p, имеем (теорема Ферма):*

*a p*1 *1(mod p).* (56)

Эта теорема является следствием теоремы **64** при m=p. Последней теореме можно придать более удобную форму. Именно, умножая обе части сравнения (1) на *а*, получим сравнение:

*а p*  *а(mod p)*,

справедливое уже при всех целых *а*, так как оно верно и при *а*, кратном *р*.

### 3.4. Решение сравнений первой степени

Сравнение первой степени с одним неизвестным имеет вид:

f (x)0 (mod m); f (х) = ах + аn. (57)

*Решить сравнение* – значит найти все значения *х*, ему удовлетворяющие. Два сравнения, которым удовлетворяют одни и те же значения *х*, называются равносильными.

Если сравнению (57) удовлетворяет какое-либо *x = x1*, то (согласно **49**) тому же сравнению будут удовлетворять и все числа, сравнимые с *x*1, по модулю *m: x**x1 (mod m)*. Весь этот класс чисел считается за *одно решение*. При таком соглашении можно сделать следующий вывод.

**66.** С*равнение* (57) *будет иметь столько решений, сколько вычетов полной системы ему удовлетворяет*. П р и м е р. Сравнению

6x – 4  0 (mod 8)

среди чисел 0, 1,2, 3, 4, 5, 6, 7 полной системы вычетов по модулю 8 удовлетворяют два числа: х = 2 и х = 6. Поэтому указанное сравнение имеет два решения: x2 (mod 8), х6 (mod 8).

Сравнение первой степени перенесением свободного члена (с обратным знаком) в правую часть можно привести к виду

axb (mod m). (58)

Рассмотрим сравнение, удовлетворяющее условию (*а, m*) = 1.

Согласно **66** наше сравнение имеет столько решений, сколько вычетов полной системы ему удовлетворяет. Но когда *x* пробегает полную систему вычетов по модулю *m,* то *ах* пробегает полную систему вычетов (из **60**). Следовательно, при одном и только одном значении *х,* взятом из полной системы, *ах* будет сравнимо с *b.* Итак,

**67.** *при (а, m) = 1 сравнение ax**b* (mod m) *имеет одно решение.*

Пусть теперь *(a, m) = d* > 1. Тогда, чтобы сравнение (2) имело решения, необходимо (из **55**), чтобы *b* делилось на *d,* иначе сравнение (2) невозможно ни при каком целом *х.* Предполагая поэтому *b* кратным *d, положим a = a1d*, *b = b1d, m = m1d.* Тогда сравнение (2) будет равносильно такому (по сокращении на *d*): *a1x**b1* (mod *m*)*,* в котором уже *(а1,m1*) = 1*,* и потому оно будет иметь одно решение по модулю *m*1. Пусть *х*1 – наименьший неотрицательный вычет этого решения по модулю *m*1*,* тогда все числа *х,* образующие это решение, найдутся в виде

*x**x1*(mod *m*1)*.* (59)

По модулю же *m* числа (3) образуют не одно решение, а больше, именно столько решений, сколько чисел (3) найдется в ряде 0, 1, 2, *...,* m *–* 1 наименьших неотрицательных вычетов по модулю *m.* Но сюда попадут следующие числа (3):

*x1, x1 + m1, x1* + *2m1, ..., x1 + (d* – *1) m1*, (60)

т.е. всего *d* чисел (3); следовательно, сравнение (2) имеет *d* решений. Получаем теорему:

1. *Пусть (a, m) = d. Сравнение ax**b (mod m) невозможно, если b не делится на d. При b, кратном d, сравнение имеет d решений.*
2. Способ решения сравнения первой степени, основанный на теории непрерывных дробей.

Разлагая в непрерывную дробь отношение *m:а,*

*m* 1

 *q*1

*a q*2  1 ... 1

*q*3 *qn*

и рассматривая две последние подходящие дроби:

*Pn*1 , *Pn*  *m*,

*Qn*1 *Qn a*

согласно свойствам непрерывных дробей (согласно **30**) имеем:

*mQn*1  *aPn*1  ( 1)*n*, *aPn*1  ( 1)*n*1(mod*m*), *a* ( 1)*n*1*P bn*1  *b*(mod*m*).

Итак, сравнение имеет решение *x*  ( 1)*n*1*P bn*1 (mod*m*),

для разыскания, которого достаточно вычислить *Pn-1* согласно способу, указанному в **30**.

П р и м е р. Решим сравнение

111x = 75 (mod 321). (61)

Здесь (111, 321) = 3, причем 75 кратно 3. Поэтому сравнение имеет три решения.

Деля обе части сравнения и модуль на 3, получим сравнение

37x = 25 (mod 107), (62)

которое нам следует сначала решить. Имеем:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| q |  | 2 | 1 | 8 | 4 |
| P3 | 1 | 2 | 3 | 26 | 107 |



Значит, в данном случае n = 4, Pn-1 = 26, b = 25, и мы имеем решение сравнения (5) в виде

x-26 ∙ 2599 (mod 107).

Отсюда решения сравнения (4) представляются так:

х99; 99 + 107; 99 + 2 ∙ 107 (mod 321), т.е.

х= 99; 206; 313 (mod 321).

#### Вычисление обратного элемента по заданному модулю

**70.** Если целые числа *a* и *n* взаимно просты, то существует число *a’*, удовлетворяющее сравнению *a ∙ a’≡1(mod n)*. Число *a’* называется *мультипликативным обратным к a по модулю n* и для него используется обозначение *a-1(mod n)*.

Вычисление обратных величин по некоторому модулю может быть выполнено решением сравнения первой степени с одним неизвестным, в котором за *x* принимается число *a’*.

Чтобы найти решение сравнения

*a ∙x ≡ 1(mod m)*,

где *(a, m) = 1,*

можно воспользоваться алгоритмом Евклида (**69**) или теоремой Ферма-Эйлера, которая утверждает, что если *(a, m) = 1*, то

aφ(m) ≡ 1(mod m).

Поэтому

x ≡ aφ(m)-1 (mod m).

### 3.5. Группы и их свойства

Группы – один из таксономических классов, используемых при классификации математических структур с общими характерными свойствами. Группы имеют две составляющие: *множество* (*G*) и *операции* (), определенные на этом множестве.

Понятия множества, элемента и принадлежности являются базисными неопределяемыми понятиями современной математики. Любое множество определяется элементами, входящими в него (которые, в свою очередь, тоже могут быть множествами). Таким образом, мы говорим, что множество определено или задано, если для любого элемента мы можем сказать, принадлежит ли он этому множеству или нет.

Для двух множеств A, B записи B A, B A, B ∩ A, B  A, B \ A, A × B означают, соответственно, что B является подмножеством множества A (т.е. любой элемент из B содержится также и в A, например, множество натуральных чисел содержится в множестве действительных чисел; кроме того, всегда A  A), B является собственным подмножеством множества A (т.е. B  A и B ≠ A), пересечение множеств B и A (т.е. все такие элементы, которые лежат одновременно и в A, и в B, например пересечение целых чисел и положительных действительных чисел есть множество натуральных чисел), объединение множеств B и A (т.е. множество, состоящее из элементов, которые лежат либо в A, либо в B), разность множеств B и A (т.е. множество элементов, которые лежат в B, но не лежат в A), декартово произведение множеств A и B (т.е. множество пар вида (a, b), где a A, b B). Через |A| всегда обозначается мощность множества A, т.е. количество элементов в множестве A.



Операция – это правило, согласно которому любым двум элементам множества *G* (*a* и *b*) ставится в соответствие третий элемент из G: *а* *b.*

Множество элементов *G* с операцией  называется *группой*, если удовлетворяются следующие условия:

1. *Ассоциативность*: для любых элементов *a, b, c*  *G* выполняется равенство *a*  (*b*  *c)= (a*  *b)*  *c.*
2. *Единичный элемент:* существует такой элемент *е*  *G,* что при любом *a*  *G* выполняются равенства *a*  *e= e* *a = a.*
3. *Обратный элемент*: для каждого *a*  *G* найдется элемент *a′**G*, удовлетворяющий соотношению *a*  *a′ = a′*  *a = e* .

Элемент *e* из *G* называется *нейтральным* элементом группы, а элемент *a′ – обратным* элементом к *a* . Обратный элемент обозначается *a′ = a -1.*

Группы, в которых операция *коммутативна*, то есть для любой пары *a*, *b*  *G,* выполняется равенство *a*  *b= b*  *a*, называются *коммутативными* группами или *абелевыми* группами.

Число элементов в группе называется ее *порядком.*

С точки зрения решения уравнений основное свойство группы в том, что в ней однозначно разрешены уравнения вида:

*a*  *x = b, y*  *a = b,*

при любых *a, b*  *G*.

П р и м е р ы:

1. Целые, рациональные, действительные, комплексные числа по сложению.
2. Ненулевые рациональные, действительные, комплексные числа по умножению.

Все эти группы являются абелевыми.

### 3.6. Кольца и тела в абстрактной алгебре

В абстрактной алгебре кольцо **́** – естественное обобщение целых чисел. Кольцо – это множество, на котором заданы две операции, «сложение» и «умножение», со свойствами, напоминающими сложение и умножение целых чисел.

Кольцо – это множество *R*, на котором заданы две бинарные операции: + и × (называемые *сложение* и *умножение*), со следующими свойствами:

2.

1. – ассоциативность сложения;



–

коммутативность сложения;



–

1. существование нейтрального

элемента относительно сложения;

1. – существование обратного эле-

мента относительно сложения;

1. – дистрибутивность.

Кольца могут обладать следующими свойствами:

ассоциативность умножения: (ас-

социативное кольцо);

наличие единицы: (кольцо с едини-

цей);

коммутативность умножения: (коммутативное

кольцо); отсутствие делителей нуля: .

Обычно под кольцом понимают ассоциативное кольцо с единицей.

Кольца, для которых выполнены все вышеперечисленные условия, называются целостными (иногда также *областями* целостности или просто областями, хотя условие коммутативности не всегда считается обязательным).

Таким образом, множество *R* с двумя бинарными ассоциативными операциями «+» и «» называется *кольцом*, если выполняются следующие условия:

1. Множество *R* с бинарным сложением «+» является абелевой группой.
2. Операция умножения «» удовлетворяет дистрибутивности относительно операции сложения «+», т.е. (*a + b)*  *c = a*  *c + b*  *c* и *a*  *(b + c) = a*  *b + a*  *c.*

Непустое подмножество называется *подкольцом* *R*, если *A* само является кольцом относительно операций, определенных в *R*. П р и м е р ы: – целые числа (с обычным сложением и умножением).



– кольцо вычетов по модулю натурального числа *n* (множество, образующее полную систему вычетов целых чисел по модулю *n* с операциями «+» и «» по модулю *n* – это коммутативное кольцо).

### 3.7. Конечные поля

Ассоциативное кольцо с единицей, в котором каждый ненулевой элемент обратим, называется *телом.* Тело – это кольцо, в котором ненулевые элементы образуют группу по умножению.

Коммутативное тело называется полем, т.е. *поле* – это коммутативное кольцо, с единицей, отличной от нуля, в котором любой ненулевой элемент обратим. Т.е. полем называется множество элементов, на котором определены две операции: одна из них называется сложением и обозначается «+», а другая – умножением и обозначается «», даже если эти операции не являются обычными операциями сложения и умножения чисел. Для того, чтобы множество элементов, на котором заданы операции сложения и умножения, было полем, необходимо, чтобы по каждой из этих операций выполнялись все групповые аксиомы, а также выполнялся дистрибутивный закон. Кроме того, по каждой операции группа должна быть коммутативной, а групповые свойства по операции умножения справедливы для всех ненулевых элементов поля. П р и м е р ы:

* кольцо рациональных чисел, являющееся полем.



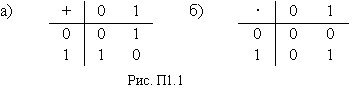
* кольцо вещественных чисел, являющееся полем.
* кольцо многочленов от *n* переменных над полем *R*.
* кольцо вычетов по модулю натурального числа *n* является полем в случае, когда *n= p*, т.е*. n* – простое число.

Поля с конечным числом элементов q называют полями Галуа по имени их первого исследователя Эвариста Галуа и обозначают GF(q).

Число элементов поля *q* называют порядком поля. Конечные поля используются для построения большинства известных кодов и их декодирования.

В зависимости от значения *q* различают простые или расширенные поля. Поле называют простым, если *q* – простое число. Для обозначения простых чисел будем использовать символ p. Простое поле образуют числа по модулю p: 0, 1, 2,…, p–1, а операции сложения и умножения выполняются по модулю p.

Наименьшее число элементов, образующих поле, равно 2. Такое поле должно содержать 2 единичных элемента: 0 относительно операции сложения и 1 относительно операции умножения. Это поле GF(2), или двоичное. Правила сложения и умножений для элементов GF(2) приводятся на рис. 3.1.

Рис. 3.1. Правила сложения и умножений для элементов GF(2)

GF(3) –троичное поле с элементами 0, 1, 2. Для него правила сложения и умножения приводятся на рис. 3.2.

нтов GF(3) Формирование таблиц производится приведением результата сложения или умножения чисел, записанных во главе строк или столбцов, по модулю *p*, т.е. в качестве результата операции принимается остаток от деления полученного числа на *p*.

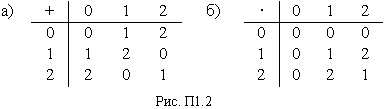


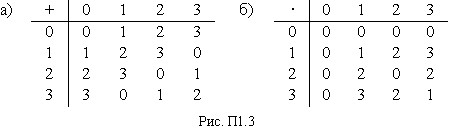
Рис. 3.2

.

Правила сложения и умножений для элеме

Анализируя состав таблиц, легко убедиться, что 0 и 1 как единичные элементы по операции сложения и умножения не изменяют значения других элементов поля по соответствующей операции. Кроме того, видно, что для каждого элемента по операции сложения и для ненулевых элементов по операции умножения имеются обратные.

На рис. 3.3 приведены правила сложения и умножения для элементов GF(4) при попытке построить это поле из чисел 0, 1, 2, 3 по предыдущей конструкции.

Рис. 3.3. Правила сложения и умножения для элементов GF(4)

Из рис. 3.3 б видно, что для элемента 2 по операции умножения отсутствует обратный, т.е. набор чисел 0, 1, 2, 3 не является полем при введении операции по модулю 4. Такой результат объясним тем, что 4 не является простым числом. Для поля GF(5) с элементами 0, 1, 2, З, 4 правила сложения и умножения приводятся на рис. 3.4.

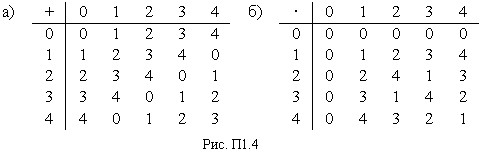


Рис. 3.4. Правила сложения и умножения для поля GF(5) с элементами 0, 1, 2, З, 4

Изучим возможность построения полей с элементами в виде последовательностей чисел.

Определим условия, при которых последовательности длины m с элементами из поля GF(p) образуют поле.

Рассмотрим последовательности длины 4 с элементами из GF(2). Такие последовательности можно складывать как векторы, и нулевым элементом по операции сложения является 0000. Для задания операции умножения сопоставим каждой последовательности многочлен от α:



Умножение таких многочленов может дать степень, большую чем 3, т.е.

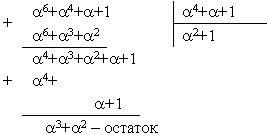
последовательность, не принадлежащую рассматриваемому множеству.

Для того чтобы свести ответ к многочлену степени не более 3, положим, что α удовлетворяет уравнению степени 4, например,

Π(α) = 1+ α + α 4 = 0, или α 4 = 1+ α. Тогда α 5 = α + α 2, α 6 = α 2+ α 3;

1+ α + α 4+ α 6 = 1+ α +1+ α + α 2+ α 3 = α 2+ α 3.

Это эквивалентно делению на многочлен 1+ α + α4 и нахождению остатка от деления:



Таким образом, имеет место аналогия при формировании поля из чисел и последовательностей чисел (многочленов). Эта аналогия распространяется и на то, что для обратимости введенной операции умножения (чтобы система элементов в виде последовательностей длины m или многочленов степени меньшей m, образовывала поле) многочлен Π(α) должен быть неприводим над полем своих коэффициентов (неприводимый многочлен над полем *k* ― это многочлен *p*(*x*1,*x*2,..,*xn*) от *n* переменных над полем *k*, являющийся простым элементом кольца *k*[*x*1,*x*2,..,*xn*], то есть непредставимый в виде произведения *p* = *qr*, где *q* и *r* ― многочлены с коэффициентами из *k*, отличные от константы).

Поле, образованное многочленами над полем GF(р) по модулю неприводимого многочлена p(x) степени m, называется расширением поля степени m над GF(p) или расширенным полем. Оно содержит pm элементов и обозначается GF(pm).

Поле, образованное шестнадцатью двоичными последовательностями длины 4, или многочленами степени 3 и менее с коэффициентами из GF(2) по модулю многочлена x4+ x+1, неприводимого над GF(2), является примером расширенного поля GF(24), которое может быть обозначено также GF(16).

Важнейшим свойством конечных полей является следующее. Множество всех ненулевых элементов конечного поля образует группу по операции умножения, т.е. мультипликативную группу порядка *q–1*.

Группа, которая состоит из всех степеней одного из ее элементов, называется *циклической* группой.

Из рассмотренного свойства конечных полей вытекают два важных следствия.

1. Многочлен xq–1–1 имеет своими корнями все *q–1* ненулевых элементов поля GF(q), т.е.



В поле GF(q) элемент α, имеющий порядок *e = q–1*, называется *примитивным*. Отсюда следует, что любой ненулевой элемент GF(q) является степенью примитивного элемента.

1. Любое конечное поле GF(q) содержит примитивный элемент, т.е.

мультипликативная группа GF(q) является циклической.

В прикладных целях обычно используются задания конечных полей в виде кольца вычетов целых чисел по простому модулю *p* (поле *GF(p)*) или факторкольца кольца многочленов над полем *GF(p)* по модулю неприводимого многочлена (поля *GF(pt)*, *t >* 1).

Наличие в конечном поле примитивного элемента *α* позволяет ввести понятие логарифма для ненулевых элементов этого поля. Логарифм элемента *β* по основанию *α* определяется как наименьшее целое неотрицательное число *k,* удовлетворяющее равенству *β = αk* .

В настоящее время задача вычисления логарифма в конечном поле в общем случае не имеет достаточно эффективных алгоритмов решения и по этой причине, наряду с задачей разложения на множители, используется при построении стойких криптографических алгоритмов и протоколов.

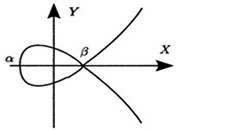
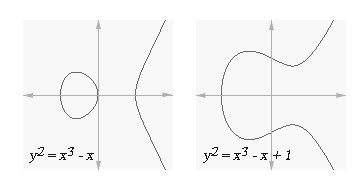
### 3.8. Эллиптически кривые над конечными полями

Первоначально эллиптической кривой называлась гладкая кривая на Декартовой плоскости, описываемая следующим уравнением:

у2+а1ху+а3у=х3+а2х2+а4х+а6. (63)

Если все коэффициенты и неизвестные – действительные числа, то путем замены переменных уравнение (63) может быть преобразовано к следующему, более простому виду (рис. 3.5):

у2=х3+ах+b. (64)



**D < 0**

**D > 0**

**D = 0**



***Рис. 3.5.*** *Эллиптические кривые*

Уравнение (63) может рассматриваться над произвольными полями и, в частности, над конечными полями, представляющими для криптографии особый интерес. В последнем случае решением уравнения (63) является множество отдельных точек, а не линия, но и в этом случае по традиции говорят об эллиптических кривых.

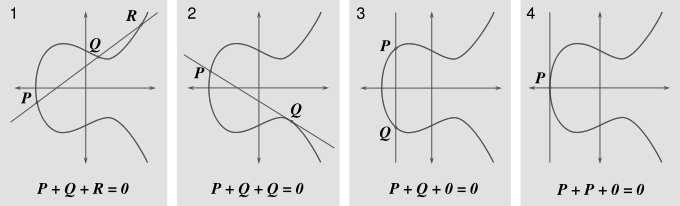
Эллиптические кривые используются в криптографии с середины 80х гг. прошлого века в качестве инструмента решения некоторых теоретикочисловых задач и в качестве основы для построения криптосистем. Любая кривая над полем действительных чисел, задаваемая уравнением общего вида (63), обладает следующими свойствами, в формулировке которых условимся точку касания считать за два пересечения:

1. Любая вертикальная (параллельная оси Оу) прямая не пересекает кривую ни разу или пересекает ее дважды.
2. Любая другая прямая пересекает кривую один или три раза.

Добавим к множеству точек эллиптической кривой фиктивный элемент «О» (называемый обычно «бесконечной точкой»), который будет играть роль нулевого элемента конструируемой аддитивной группы: для любой точки Р эллиптической кривой положим Р + О = О + Р = Р. Операции отрицания и сложения определяем таким образом, чтобы для каждой прямой, пересекающей эллиптическую кривую дважды или трижды, «сумма» всех точек пересечения с учетом их кратности при касании была «равна нулю»:

P + P' = O – для вертикальных прямых,

P + Q + R = O – для невертикальных прямых (рис. 3.6).



***Рис. 3.6.*** *Прямая пересекает эллиптическую кривую*

Первое соотношение определяет правило отрицания – аддитивным обратным элементом для любой точки P эллиптической кривой является другая точка пересечения кривой и вертикальной прямой: P' = -P. Второе соотношение определяет правило сложения точек: суммой двух точек кривой является отрицание третьей точки пересечения кривой и прямой, проведенной через первые две: Р + Q = -R = R' (см. рис. 3.6). Для определенных таким образом аддитивных операций выполняются все требования к операциям в абелевых группах:

1. Операция сложения точек очевидным образом коммутативна: для любых точек Р, Q эллиптической кривой справедливо Р + Q = Q + Р.
2. В проективной геометрии доказывается ассоциативность операции сложения точек: для любых трех точек Р, Q, R эллиптической кривой справедливо (Р + Q) + R = P + (Q+R).

Таким образом, множество точек эллиптической кривой с добавленной к нему бесконечной точкой О и определенными указанным выше способом операциями отрицания и сложения точек образуют аддитивную абелеву группу, элементы которой могут использоваться в качестве «заменителей» обычных чисел.

Определенные выше операции над точками кривых могут быть распространены на случай произвольного поля. Необходимые формулы могут быть получены, если воспользоваться алгебраическими выражениями для геометрических понятий – «прямая», «вертикальная прямая», «касательная». Следует указать, что не для всякого конечного поля уравнение кривой может быть приведено к виду (64).

|  |
| --- |
| Эллиптические кривые определяются над двумя типами конечных полей:  поля нечетной характеристики ( , где *p* > 3 – большое простое число), и поля характеристики 2 ( ). Описание эллиптических кривых в полях нечетной характеристики существенно проще, поэтому остановимся на них подробнее. Рассматривается уравнение двух переменных :    *y*2 = *x*3 + *Ax* + *B*(mod *p*),    где  – константы, удовлетворяющие .  *Множеством точек эллиптической кривой* называется множество пар (*x*,  *y*), удовлетворяющих вышеуказанному уравнению, объединенное с нулевым элементом :    .  *Теорема Хассе об эллиптических кривых* утверждает, что количество точек на эллиптической кривой близко к размеру конечного поля:    .  П р и м е р. Пусть *p* = 5, а уравнение эллиптической кривой: *y*2 = *x*3 + 3*x* +  + 2(mod 5). Тогда (1,1) и (1,4) – точки эллиптической кривой, так как    .  Следует отметить, что в у каждого элемента кроме нуля есть либо два квадратных корня, либо нет ни одного, поэтому точки эллиптической кривой всегда получаются парами (*x*,*y*1) и (*x*,*y*2), так что    . |

Следует указать, что не для всякого конечного поля уравнение кривой может быть приведено к виду (64). Так, для конечного поля характеристики 3 невозможно избавиться от члена а2х2, если а2 отлично от нуля. Для полей характеристики 2 уравнение (63) может быть приведено к одному из следующих видов: у2+ау=х3+Ьх+с (суперсингулярная кривая); (65)

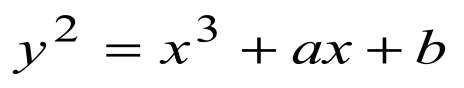
у2+аху=х3+bх2+с (несуперсингулярная кривая). (66)

Заметим, если степень поля есть нечетное число, то заменой переменных можно добиться того, что коэффициенты в уравнении (3.8.3) будут равны либо 0, либо 1.

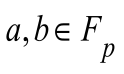
Для криптографии особую важность представляет нахождение кратного точки эллиптической кривой: Q = kР, где k – большое натуральное число. Эта операция аналогична возведению элемента конечного поля в большую натуральную степень. На сегодняшний день не известно эффективных алгоритмов для решения задачи дискретного логарифмирования в полях точек эллиптических кривых.

Эллиптической кривой , определенной в конечном поле , где будем называть множество точек с координатами , удовлетворяющих уравнению:



**,**

где и вместе со специальной точкой ,



называемой точкой бесконечности или, как еще иногда говорят, ***нулевой точкой*** (Коблиц, 2001). В группе точек на эллиптической кривой точка бесконечности  нужна для обеспечения существования обратного элемента в группе и необходимых свойств операций.

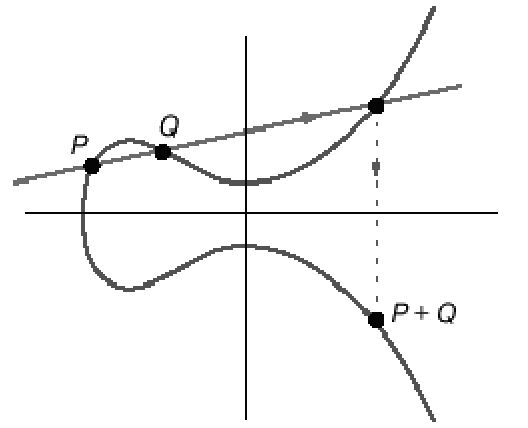
Рассмотрим основные операции над точками и их геометрическую интерпретацию на примере обычных плоских эллиптических кривых, т.е. кривых, определенных над полем вещественных чисел. Полученные в результате аналитические формулы для рассматриваемых операций являются действительными для эллиптических кривых, определенных над любым полем.

***Точкой, обратной к данной точке*** , называется точка с координатами  и обозначается как , т.е. точка, зеркально отраженная от оси абсцисс.

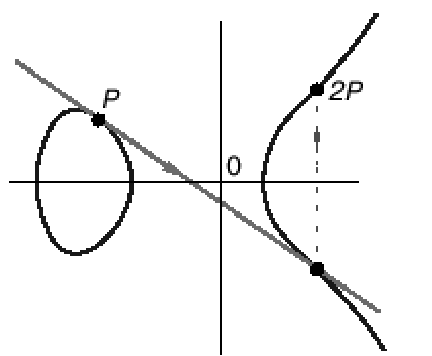


Геометрический смысл операции сложения двух точек  на эллиптической кривой (рис. 3.7) заключается в том, что через них проводится секущая, пересекающая кривую в третьей точке. Обратная точка для точки пересечения дает результат операции сложения.

Если некоторая точка эллиптической кривой  складывается сама с собой (рис. 3.8), то такая процедура называется ***удвоением точки***. Геометрический смысл подобной операции заключается в том, что через точку  проводится секущая линия, которая пересекает кривую в некоторой точке. Обратная точка к точке пересечения и будет результатом удвоения точки .



***Рис. 3.7.*** *Геометрический смысл операции сложения двух точек на эллиптической кривой*



***Рис. 3.8.*** *Геометрический смысл операции сложения точки эллиптической кривой самой с собой – удвоение точки*

Операции сложения и удвоения точек имеют аналитические выражения. ***Сложение точек****:* которых . То-

гда

,

Пусть



–

две

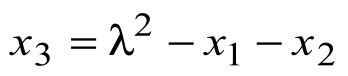
точки,

для

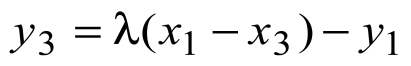


,

где



,



и

.

***Удвоение точки:***

Пусть точка , для которой *y*1 ¹ 0 (если , тогда ,



т.е. ), тогда , где

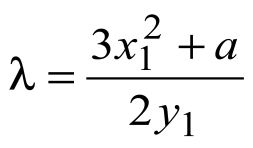
,



,



где

.

***Точка бесконечности****.* Геометрический смысл точки бесконечности заключается в том, что в ней пересекаются все прямые, параллельные оси ординат. Рассмотрим операции с участием точки бесконечности  и некоторой точки :

2) ;

3)

1)



;



.

Еще одним важным понятие эллиптической криптографии является порядок эллиптической кривой, который показывает количество точек кривой над конечным полем.

Теорема Хассе утверждает, что если **N** – количество точек кривой, определенной над полем Zq с q элементами, тогда справедливо равенство:

.

Т.к. бинарное конечное поле



состоит из 2

n

элементов

,

мы м

о-



равен



, где



жем сказать, что порядок кривой .

С числом t связано следующее определение: эллиптическая кривая над бинарным конечным полем называется *суперсингулярной*, если t делится на характеристику поля (в случае бинарного поля характеристика равна 2) без остатка. *Нельзя использовать в схемах ЭЦП суперсингулярные кривые.*

*Порядком (рангом) точки*называется такое минимальное целое положительное число *k*, что *kP = 0*. Ранг точки определяет порядок группы точек эллиптической кривой, с которыми осуществляются криптографические преобразования.

Каждая точка P эллиптической кривой над простым полем *E(Fp)* образует циклическую подгруппу группы точек эллиптической кривой. Порядок циклической подгруппы группы точек эллиптической кривой *(число точек в подгруппе) называется порядком точки эллиптической кривой*. Точка P на E(Fp) называется точкой порядка *k*, если: *kP=O*, где *k* – наименьшее натуральное число, при котором выполняется данное условие.

По теореме Лагранжа порядок точки делит порядок эллиптической кривой. При определении порядка кривой ее можно заменить удобной изоморфной кривой, так как у изоморфных кривых порядки одинаковы. При надлежащем выборе параметров *p*, *а* и *b* порядок кривой может быть простым числом, и в этом случае любая точка, кроме O, является *генератором* всего множества – т.е. такой точкой *G*, что ряд: *G*, 2*G*, 3*G*, …, *kG*, где *k* содержит все точки множества *Ep*(*a*,*b*), причем *kG* = O. Такая кривая предпочтительна для криптографического использования и может быть найдена за практически приемлемое время.

Операция сложения двух точек, принадлежащих эллиптической кривой, определенной на конечном поле, является аналогом операции умножения двух чисел в конечном поле. Операция умножения точки на некоторое число  (на скаляр) называется *композицией* *точки на эллиптической кривой* и является аналогом операции возведения в степень . Операцию умножения точки на число можно представить как процесс сложения точки самой с собой нужное число раз.

Например:



,



и т.д.

Однако существует способ быстрого умножения точки на число, который требует количества операций для умножения точки , определенной в конечном поле на некоторое число . Этот метод аналогичен алгоритму быстрого возведения в степень для элементов конечного поля, действительно:



.

В группе точек, принадлежащих эллиптической кривой, существует задача, имеющая экспоненциальный уровень сложности и получившая название *задачи дискретного логарифмирования на эллиптической кривой*. Суть ее заключается в том, что при известных точках вычислительно трудно найти такое число  (если оно существует), что . Надежность криптографических систем на эллиптических кривых основана на сложности решения этой задачи. В следующей таблице приведены оценки необходимой вычислительной мощности для нахождения дискретного логарифма в группе точек эллиптической кривой (Jonson, Menezes, н.д.).



**Вычислительная мощность для нахождения дискретного логарифма в группе точек эллиптической кривой**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Модуль кривой ,  бит | 163 | 191 | 239 | 359 | 431 |
| Порядок группы точек кривой , бит | 160 | 186 | 234 | 354 | 426 |
| Требуемая вычислительная мощность (M/Y) | 9,6\*1011 | 7,9\*1015 | 1,6\*1023 | 1,5\*1041 | 1052 |

Задача дискретного логарифмирования в группе точек эллиптической кривой является более трудной, чем задача дискретного логарифмирования в конечном поле. В этом заключается основная причина преимущества использования криптосистем на эллиптических кривых, которые обеспечивают такой же уровень стойкости при использовании чисел существенно меньшего размера по сравнению с более традиционными криптосистемами, надежность которых заключается в сложности задачи факторизации или дискретного логарифмирования в конечном поле. Соответственно, при использовании чисел одинаковой размерности уровень стойкости криптосистем на эллиптических кривых значительно выше.

На сегодняшний день лучший из известных алгоритмов – -метод Полларда – дает следующую оценку для решения задачи дискретного логарифмирования в группе точек эллиптической кривой: , где – количество процессоров для параллельных вычислений, а – порядок группы точек эллиптической кривой (Escott and others, 1999). С помощью этого метода в 2000 году была решена задача дискретного логарифмирования в группе точек эллиптической кривой с порядком в 108 бит. В 2002 году была решена задача для эллиптической кривой с порядком в 109 бит. Для сравнения отметим, что криптосистема, использующая преобразования в группе точек эллиптической кривой с порядком в 160 бит, эквивалентна по стойкости криптосистеме RSA с модулем шифрования в 1024 бита.



#### Вопросы и задания для самостоятельной работы

1. Как определить наибольший общий делитель?
2. Как составить таблицу простых чисел, пользуясь способом «Решето Эратосфена»?
3. Выпишите каноническое разложение числа на простые сомножители.
4. Какое число называется Евклидовым числом?
5. Назовите три способа отыскания функции Эйлера.
6. Какова связь непрерывных дробей с алгоритмом Евклида?
7. Как сравнить целые числа по модулю?
8. Какие свойства сравнений подобны свойствам равенств, и какие свойства особенные?
9. Опишите алгоритм решения сравнений первой степени по алгоритму Евклида.
10. Как вычислить обратный элемент по заданному модулю?
11. Какой элемент является мультипликативным обратным по заданному модулю?
12. Как найти мультипликативный обратный элемент по модулю m?

## ТЕМА 4. КЛАССИФИКАЦИЯ ШИФРОВ

### 4.1. Классификация шифров

Существующие способы шифрования могут быть классифицированы по различным признакам.

В качестве первичного признака, по которому производится классификация, используется тип преобразования, осуществляемого с открытым текстом при шифровании.

Если фрагмент открытого текста (отдельные буквы или группа букв) заменяются некоторыми их эквивалентами (буквами, цифрами, символами или их комбинацией) в шифртексте, то соответствующий шифр относится к классу *шифров замены.*

Если буквы открытого текста при шифровании каким-нибудь способом перестанавливаются, то есть изменяется только порядок следования символов открытого текста, то соответствующий шифр относится к классу *шифров перестановки.*

Эффективным средством повышения стойкости шифрования является комбинированное использование нескольких различных способов. С помощью двух или более способов осуществляется последовательное шифрование исходного текста. На практике наибольшее распространение получили следующие комбинации: 1) перестановка + гаммирование; 2) подстановка + гаммирование; 3) гаммирование + гаммирование; 4) подстановка + перестановка. Комбинированное шифрование предусмотрено, например, в национальном стандарте США криптографического закрытия данных. Всевозможные композиции различных шифров приводят к классу комбинированных шифров, которые обычно называются *композиционными*.

Таким образом, все шифры можно разделить на три класса, в результате чего образуется первый уровень классификации шифров (рис. 4.1).

**ШИФРЫ**

**ШИФРЫ ЗАМЕНЫ**

**ШИФРЫ**

**ПЕР**

**Е**

**СТАНОВКИ**

**КОМПОЗИЦИОННЫЕ**

**ШИФРЫ**

***Рис. 4.1.*** *Основные классы шифров*

В современной криптографии широко используется деление шифров по типу ключей. Если ключ зашифрования совпадает с ключом расшифрования (kз = kр), шифры называются *симметричными*, если нет (kз ≠ kр) – *асимметричными*.

При симметричном шифровании (рис. 4.2) используется один ключ (*секретный*), с помощью которого отправитель А зашифровывает открытое сообщение, а получатель В его расшифровывает. Очевидно, что для данного ключа выполняемое шифрпреобразование должно быть обратимым, то есть должно существовать обратное преобразование, которое при выбранном ключе однозначно определяет открытое сообщение. При этом сам ключ передается от отправителя А к получателю В отдельно, используя другой (защищенный или очень надежный) канал связи. Шифрование с симметричным ключом обеспечивает высокую скорость обмена информацией, однако надежность такого шифрования не столь высока из-за возможности перехвата секретного ключа злоумышленником.

Секретный ключ

**К**

**а**

**нал**

связи

Получатель

В

Расшифров

а

н

ие

Шифров

а

ние

Отправитель

А

***Рис. 4.2.*** *Симметричное шифрование*

Асимметричное шифрование (рис. 4.3) сложнее, но надежнее. Для его реализации нужны два взаимосвязанных ключа: *открытый и секретный.* Получатель сообщает всем желающим свой открытый ключ, позволяющий шифровать для него сообщения. Если кому-то необходимо отправить зашифрованное сообщение, то он выполняет шифрование, используя открытый ключ получателя. Получив сообщение, последний расшифровывает его с помощью своего

секретного ключа. Таким образом, отличительной особенностью данного метода шифрования является разделение ключей для зашифрования и расшифрования. При этом ключ для зашифрования не требуется делать секретным, более того, он может быть общедоступным и содержаться в телефонном справочнике вместе с фамилией владельца.

Открытый

ключ

**К**

**а**

**нал**

связи

Получатель

В

Расшифров

а

ние

Шифров

а

ние

Отправитель

А

***Секретный***

ключ

***Рис. 4.3.*** *Асимметричное шифрование*

Основное преимущество асимметричного шифрования перед симметричным состоит в том, что при этом нет необходимости в передаче секретного ключа, который может быть перехвачен злоумышленником. Недостатком асимметричных шифров является большая трудоемкость последних и, как результат, меньшие скорости при шифровании. Поскольку вычисления в случае асимметричного шифрования сложнее, то процедура расшифрования занимает также больше времени.

Механизмы использования асимметричных шифров предполагают различные способы замены, поэтому деление шифров на симметричные и асимметричные целесообразно проводить в классе шифров замены.

Перед шифрованием открытый текст предварительно представляется в виде последовательности «подслов», называемых *шифрвеличинами*. При зашифровании шифрвеличины заменяются некоторыми эквивалентами в шифртексте, которые называются *шифробозначениями*. Если шифрвеличина представляет собой букву открытого текста, то число шифрвеличин ограничено числом букв в алфавите. Это переносит избыточность открытого текста в шифртекст и делает диаграмму повторяемости знаков криптограммы рельефной, а сам шифр слабым по отношению к взлому. Шифры, в которых последовательно шифруется каждый символ открытого текста, называются *поточными*. Если шифрвеличина представляет собой несколько букв открытого текста (две буквы – биграммы, три – триграммы, больше – блоки), то общее количество шифрвеличин увеличивается. Чем больше разница между числом шифрвеличин и числом букв в алфавите, тем более равномерной должна стать диаграмма повторяемости знаков шифртекста и тем более криптостойким сам шифр. Для увеличения количества шифрвеличин используются *блочные шифры*, в которых открытый текст делится при шифровании на блоки фиксированной длины (например, биграммы в шифрах замены или столбцы в шифрах перестановки).

Таким образом, все шифры можно поделить на блочные и поточные шифры (рис. 4.4).

**ШИФРЫ**

**ШИФРЫ ЗАМЕНЫ**

**ШИФРЫ**

**ПЕР**

**Е-**

**СТАНОВКИ**

**КОМПОЗИЦИОННЫЕ**

**ШИФРЫ**

СТРОЧНЫЕ ПЕРЕСТ

А-

НОВКИ

ВЕРТИКАЛЬНЫЕ ПЕР

Е-

СТАНОВКИ

**МАРШРУТНЫЕ**

**ПЕР**

**Е**

**СТАНОВКИ**

**ШИФРЫ ГАММИР**

**О-**

**ВАНИЯ**

МНОГОАЛФАВИТНЫЕ

ШИФРЫ

ОДНОАЛФАВИТНЫЕ

ШИФРЫ

**БЛОЧНЫЕ ШИФРЫ**

**ПОТОЧНЫЕ**

**ШИФРЫ**

**АСИММЕТРИЧНЫЕ**

**ШИФРЫ**

**СИММЕТРИЧНЫЕ**

**ШИФРЫ**

**ОДНОЗНАЧНЫЕ**

**З**

**А**

**МЕНЫ**

**МНОГОЗНАЧНЫЕ**

**З**

**А**

**МЕНЫ**

ПОВОРОТНЫЕ РЕШЕТКИ

ЛАБИРИНТЫ

***Рис. 4.4.*** *Общая схема классификации шифров*

Другой способ выравнивания диаграммы повторяемости знаков шифртекста и, как следствие, повышения криптостойкости шифров состоит в приведении соответствия шифрвеличине группы шифробозначений. Если для этого используется несколько алфавитов, то шифры называются *многоалфавитными*. Если одной и той же шифрвеличине в разных местах шифртекста ставятся в соответствие различные шифробозначения (например, в зависимости от числа повторяемости в открытом тексте), то такие шифры относятся к *шифрам многозначной замены*. Если используется один алфавит и каждой шифрвеличине ставится в соответствие ее шифробозначение, то шифр замены называется *одноалфавитным шифром замены* или *шифром простой замены*.

Следовательно, можно выделить подклассы шифров *одноалфавитных* и многоалфавитных, однозначной замены и многозначной замены. Одноалфавитные и многоалфавитные шифры могут быть как блочными, так и поточными.

Для шифрования информации могут использоваться аналитические преобразования, основанные на преобразованиях матричной алгебры.

Шифрование k-го блока исходной информации, представленного в виде вектора Bk = ║bj║, осуществляется путем перемножения этого вектора на матрицу А = ║aіј║, используемую в качестве ключа. В результате перемножения получается блок шифртекста в виде вектора Сk = ║сі║, где элементы вектора Сk определяются по формуле:

Сi j aijbj .

*Аналитические* методы шифрования используют аналитические преобразования символов или групп символов открытого текста, поэтому относятся к классу шифров замены.

При использовании *аддитивных* методов шифрования осуществляется последовательное суммирование цифровых кодов, соответствующих символам исходной информации, с некоторой специальной последовательностью кодов, которая называется *гаммой*. Поэтому аддитивные методы представляют *шифры гаммирования*. Операция сложения называется *наложением гаммы*, обратная операция – *снятием гаммы*. Шифры гаммирования, образуя подкласс многоалфавитных шифров, относятся к поточным шифрам.

Математическая модель простого шифра гаммирования:

 (*c* ),

где Χ – любая буква шифрованного текста, Γ – любая буква гаммы, C –

любая буква открытого текста.

В шифрах перестановки открытый текст делится при шифровании на блоки фиксированной длины, в каждом из которых производится определенная конкретным видом шифра перестановка. Поэтому все шифры перестановки можно отнести к блочным шифрам.

На рис. 4.4 представлена общая схема классификации шифров, в которой приведены наиболее важные или исторически известные подклассы шифров. На этой схеме не указаны связи композиционных шифров, поскольку в основе они могут иметь любой класс.

### 4.2. Математические модели шифров

Для того чтобы иметь возможность доказывать в криптографии точные результаты, нужны *математические модели* основных исследуемых объектов, к которым относятся в первую очередь *шифр* и *открытый текст*. Введем сначала алгебраическую модель шифра (шифрсистемы), предложенную К. Шенноном.

#### Алгебраическая модель

Пусть *X, Y, K* – конечные множества возможных открытых текстов, ключей и шифрованных текстов соответственно; *Ek: X→Y* – правило зашифрования на ключе *k* *K*. Множество {Ek: kK} обозначим через *E*, а множество *{Ek(x):x**X}* – через *Ek(X)*. Пусть *Dk:Ek(X) → X* – правило расшифрования на ключе *k* *K*, и *D* – это множество *{Dk: k* *K}.* Здесь и далее будем предполагать, что если *k* *K* представляется в виде *k = (kз,kр)*, где *kз –* ключ зашифрования, а *kр –* ключ расшифрования (причем kз ≠ kр), то *Ek* понимается как функция *Ekз,* а *Dk –* как функция *Dkр*.

ОПРЕДЕЛЕНИЕ: *Шифром (шифрсистемой) назовем совокупность*

*∑A = (X, K, Y, E, D) введенных множеств, для которых выполняются следующие свойства:*

*1) для любых x**X и k**K выполняется равенство Dk(Ek(x)) = x; 2) Y =*  *Ek(X).*

*л Л*

*Шифр* – это совокупность множеств возможных открытых текстов (то, что шифруется), возможных ключей (то, с помощью чего шифруется), возможных шифртекстов (то, во что шифруется), правил зашифрования и правил расшифрования.

Условие 1) отвечает требованию однозначности расшифрования. Условие 2) означает, что любой элемент yY может быть представлен в виде Ek(x) для подходящих элементов xX и kK. В общем случае утверждение «для любых kK и yEk(X) выполняется равенство Ek(Dk(y)) = y «является неверным. Иначе говоря, функция Dk – постоянна, а Ek может меняться в процессе шифрования. Из условия 1) следует свойство инъекции функции Ek. Другими словами, если x1, x2 X, причем x1 != x2, то при любом kK выполняется неравенство

Ek(x1) != Ek(x2).

Определение, приведенное выше, вводит математическую модель, отражающую основные свойства реальных шифров. В силу этого будем отождествлять реальный шифр с его моделью *∑A*, которую будем называть *алгебраической моделью* шифра.

#### Вероятностная модель

Следуя К. Шеннону, определим априорные распределения вероятностей *P(X), P(K)* на множествах *P* и *K* соответственно. Тем самым для любого *x**X* определена вероятность *px(x)*  *P(x)* и для любого *k**K* – вероятность *pk(k)*  *P(K)*, причем выполняются равенства:  *px(x) =* 1;  *pk(k) =* 1.

*ч Ч* *ч Ч**k*

В тех случаях, когда требуется знание распределений *P(X), P(K*), будем пользоваться *вероятностной моделью* *∑B*, состоящей из пяти множеств, связанных условиями 1) и 2) определения, и двух вероятностных распределений*: ∑B = (X, K, Y, E, D, P(X), P(K)).*

Вероятностные характеристики шифров используются лишь в криптоанализе, т.е. при вскрытии (взломе) шифров.

#### Модель шифров замены

В большинстве случаев множества X и Y представляют собой объединения декартовых степеней некоторых множеств A и B соответственно, так что для некоторых натуральных L и L1: X = i=1LAi Y = i=1L1Bi. Множества A и B называют, соответственно, алфавитом открытого текста и алфавитом шифрованного текста. Другими словами, открытые и шифрованные тексты записываются привычным образом в виде последовательностей букв.

Введем *шифр простой замены* в алфавите A.

ОПРЕДЕЛЕНИЕ: Пусть X = Y = i=1LAi, K S(A), где S(A) – симметричная группа подстановок множества A. Для любого ключа k K, открытого текста X = (x1,...,xl) и шифрованного текста y = (y1,...,yl) правила зашифрования и расшифрования шифра простой замены в алфавите A определяются формулами: Ek(x) = (k(x1),..., k(xl)), Dk(x) = (k-1(y1),..., k-1(yl)), (67) где k-1 – подстановка, обратная k.

В более общей ситуации для шифра простой замены X = i=1LAi,

Y = i=1L1Bi, причем |A| = |B|, а K представляет собой множество всех биекций множества A на множество B. Правила зашифрования и расшифрования определяются для k K, x X, y Y (и обратной к k биекции k-1) формулами (67).

#### Модель шифров перестановки

Определим еще один шифр, называемый *шифром перестановки*.ОПРЕДЕЛЕНИЕ: Пусть X = Y = AL и пусть K SL, где SL – симметрическая группа подстановок множества {1,2,...,L}. Для любого ключа k открытого текста x = (x1,...,xL) и шифрованного текста y = (y1,...,yL) правила зашифрования и расшифрования шифра перестановки определяются формулами:

Ek(x) = (xk(1),..., xk(L))), Dk(y) = (yk-1(1),..., yk-1(L))), (2)

где k-1 – подстановка, обратная k.

Шифры, введенные выше, являются представителями двух наиболее важных классов *симметричных шифров*, а именно шифров замены и шифров перестановок. Другими симметричными шифрами являются композиции (или последовательные применения) некоторых шифров замены и шифров перестановки.

### 4.3. Математические модели открытого текста

Математические модели открытых текстов служат основой для автоматизации процессов криптоанализа шифртекстов, а также в процессе изучения криптостойкости различных систем шифрования.

Потребность в математических моделях открытого текста продиктована следующими двумя причинами. Во-первых, даже при отсутствии ограничений на временные и материальные затраты по выявлению закономерностей, имеющих место в открытых текстах, нельзя гарантировать того, что такие свойства указаны с достаточной полнотой. Например, хорошо известно, что частотные свойства текстов в значительной степени зависят от их характера. Поэтому при математических исследованиях свойств шифров прибегают к упрощающему моделированию, в частности, реальный открытый текст заменяется его моделью, отражающей наиболее важные его свойства.

Во-вторых, при автоматизации методов криптоанализа, связанных с перебором ключей, требуется «научить» ЭВМ отличать открытый текст от случайной последовательности знаков. Соответствующий критерий может выявить лишь адекватность последовательности знаков некоторой модели открытого текста.

Один из естественных подходов к моделированию открытых текстов связан с учетом их частотных характеристик, приближения для которых можно вычислить с нужной точностью, исследуя тексты достаточной длины. Основанием для такого подхода является устойчивость частот *k*-грамм или целых словоформ реальных языков человеческого общения (то есть отдельных букв, слогов, слов и некоторых словосочетаний). Основанием для построения модели может служить также и теоретико-информационный подход, развитый в работах К. Шеннона.

Учет частот *k*-грамм приводит к следующей модели открытого текста. Пусть *P(k) (А)* представляет собой массив, состоящий из приближений для вероятностей *p(b1b2….bk)* появления *k-*грамм *b1b2….bk* в открытом тексте, *k**N,*  А = {а,,..., аn} – алфавит открытого текста, *b**А,* i = 1… *k*.

Тогда источник «открытого текста» генерирует последовательность

*c1,с2,...,сk,сk+1,...* знаков алфавита *А*, в которой *k*-грамма *с,с2...сk* появляется с вероятностью *р(с2с3...сk)* *Р(K) (A)*, следующая *k*-грамма *с2с3...сk+*1 появляется с вероятность *р(с2с3...сk+1)* *Р(K) (A)* и т.д. Назовем построенную модель открытого текста *вероятностной моделью k-го приближения*.

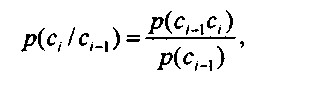
Таким образом, простейшая модель открытого текста – *вероятностная модель первого приближения –* представляет собой последовательность знаков с1, с2,... в которой каждый знак *сi-,I =* 1,2,... появляется с вероятностью *р(сi)* *Р(i) (A),*

независимо от других знаков. Будем называть также эту модель *позначной моделью*

*открытого текста.* В такой модели открытый текст *c1с2...сl* имеет вероятность

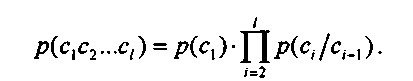


В вероятностной модели второго приближения первый знак *с1* имеет вероятность *р(с1)* *Р(1) (A),* а каждый следующий знак *сi* зависит от предыдущего и появляется с вероятностью



где *р(сi-1сi)* *Р(г)(А), р(сi-1)*  *Р(1) (А), i =* 2,3,....

Другими словами, модель открытого текста второго приближения представляет собой *простую однородную цепь Маркова.* В такой модели открытый текст *с1с2...сl* имеет вероятность



Модели открытого текста более высоких приближений учитывают зависимость каждого знака от большего числа предыдущих знаков. Модели, как и сам открытый текст, обладают выраженной эргодичность. *Эргодичность* – специальное свойство изменяющейся (динамической) системы, состоящее в том, что в процессе эволюции системы почти каждая точка ее с определенной правильностью проходит вблизи любой другой точки системы. И чем выше степень приближения, тем более «читаемыми» являются соответствующие модели. Проводились эксперименты по моделированию открытых текстов с помощью ЭВМ.

С более общих позиций открытый текст может рассматриваться как реализация *стационарного эргодического случайного процесса с дискретным временем и конечным числом состояний*. (*Эргодичность* – специальное свойство изменяющейся системы, состоящее в том, что в процессе эволюции системы почти каждая точка ее с определенной правильностью проходит вблизи любой другой точки системы, иными словами, система «забывает» свое начальное состояние и ведет себя хаотически.)

### 4.4. Критерии распознавания открытого текста

Критерий распознавания открытого текста можно построить, заменив реальный открытый текст его моделью. При этом можно воспользоваться либо стандартными методами различения статистических гипотез, либо наличием в открытых текстах некоторых запретов, таких, например, как биграмма ЪЪ в русском тексте.

Открытый текст представляет собой реализацию независимых испытаний случайной величины, значениями которой являются буквы алфавита *А =* {а1,...,аn}, появляющиеся в соответствии с распре делением вероятностей



.

Требуется определить, является ли случайная последовательность

*C1с2...сl* букв алфавита *А* открытым текстом или нет.

Пусть Н0 – гипотеза, состоящая в том, что данная последовательность – открытый текст, Н1 – альтернативная гипотеза. В простейшем случае последовательность *с1с2...сl* можно рассматривать при гипотезе Н1 как случайную и равновероятную. Эта альтернатива отвечает субъективному представлению о том, что при расшифровании криптограммы с помощью ложного ключа получается «бессмысленная» последовательность знаков. В более общем случае можно считать, что при гипотезе Н1 последовательность *C1с2...сl* представляет собой реализацию независимых испытаний некоторой случайной величины, значениями которой являются буквы алфавита *А =* {a1,…an}, появляющиеся в соответствии с распределением вероятностей

Q(1) *(А)* = (q(а,),..., *q(an*)).

При таких договоренностях можно применить, например, наиболее мощный критерий различения двух простых гипотез, который дает *лемма Неймана–Пирсона* (математические методы статистики).

В силу своего вероятностного характера такой критерий может совершать ошибки двух родов. Критерий может принять открытый текст за случайный набор знаков. Такая ошибка обычно называется *ошибкой первого рода,* ее вероятность равна

.

Аналогично вводится *ошибка второго рода* и ее вероятность

.

Эти ошибки определяют качество работы критерия. В криптографических исследованиях естественно минимизировать вероятность ошибки первого рода, чтобы не «пропустить» открытый текст. Лемма Неймана–Пирсона при заданной вероятности первого рода минимизирует также вероятность ошибки второго рода.

Критерии на открытый текст, использующие запретные сочетания знаков, например k-граммы подряд идущих букв, называются *критериями запретных kграмм.* Они устроены чрезвычайно просто. Отбирается некоторое число s редких kграмм, которые объявляются запретными. Теперь, просматривая последовательно kграмму за k-граммой анализируемой последовательности *c1c2...сl,* мы объявляем ее случайной, как только в ней встретится одна из запретных k-грамм, и открытым текстом в противном случае. Такие критерии также могут совершать ошибки в принятии решения. В простейших случаях их можно рассчитать. Несмотря на свою простоту, критерии запретных k-грамм являются весьма эффективными.

#### Вопросы и задания для самостоятельной работы

1. Приведите классификацию методов шифрования в зависимости от способа преобразования информации.
2. Что такое шифрвеличина, шифробозначение и как эти понятия соотносятся?
3. Какие особенности характерны для методов шифрования с симметричным ключом и несимметричным (открытым) ключом?
4. Чем отличаются симметричные шифрсистемы от асимметричных?
5. Какие шифры называются омофонами?
6. Поясните на примере сущность шифрования методом замены.
7. Приведите примеры шифрования методами перестановки. Что означает маршрутная перестановка?
8. Поясните сущность гаммирования как способа криптографического преобразования информации. Что является при этом ключом шифрования?
9. Почему аддитивные методы шифрования относятся к шифрам гаммирования?
10. К какому классу шифров относятся аналитические способы шифрования?
11. Приведите пример шифра перестановки, который может рассматриваться и как блочный шифр замены?
12. С какими целями в криптографии вводятся модели открытых текстов? 13. Какие подходы используются для распознавания открытых текстов?

## ТЕМА 5. ШИФРЫ ПЕРЕСТАНОВКИ

Шифрование методом перестановки заключается в том, что при этом символы шифруемого текста переставляются по определенным правилам внутри шифруемого блока символов.

Ключом шифра является перестановка номеров букв открытого текста. Зависимость ключа от длины текста во многих шифрах перестановки создает значительные неудобства в использовании шифра.

Рассмотрим ряд частных шифров перестановки, которые можно применять для зашифрования текстов любой длины.

### 5.1. Шифр Сцитала

Упрощение письма стимулировало развитие криптографии. В Древней Греции криптография уже широко использовалась в разных областях деятельности, в особенности в государственной сфере. В Спарте в V–IV веках до н.э., как уже упоминалось в историческом обзоре, использовалось одно из первых шифровальных приспособлений – Сцитала. Это был жезл цилиндрической формы, на который наматывалась полоска пергамента. Вдоль оси цилиндра на пергамент построчно записывался текст, после этого лента сматывалась с жезла и передавалась адресату, который имел точно такую же Сциталу. Такой способ шифрования осуществлял перестановку букв в сообщении, ключом к которому служил диаметр Сциталы.

Метод вскрытия этого шифра приписывается Аристотелю. Он предложил заточить конусом длинный брус и, обернув вокруг него ленту, начать сдвигать ее по конусу от малого диаметра до самого большого. В том месте, где диаметр конуса совпадал с диаметром Сциталы, буквы складывались в слоги и слова.

После этого оставалось только изготовить жезл нужного диаметра.

Шифр Сцитала эквивалентен следующему шифру маршрутной перестановки: в таблицу, состоящую из *m* столбцов, построчно записывается сообщение, после чего выписывают буквы по столбцам.

На реализацию шифра Сцитала накладываются физические ограничения. Естественно предположить, что диаметр жезла не должен превосходить 10 сантиметров. При высоте строки в 1 сантиметр на одном витке такого жезла уместится не более 32 букв (π·10 < 32). Таким образом, число перестановок, реализуемых шифром Сцитала, вряд ли превосходит 32.

### 5.2. Маршрутные перестановки

Широкое распространение получили шифры перестановки, использующие разные пути при записи исходной информации и при считывании зашифрованной информации. Такое шифрование называют *маршрутной перестановкой* или *маршрутной транспозицией*.

В простейшем случае можно, например, прочитать исходный текст задом наперед.

Можно записать исходное сообщение в прямоугольную матрицу, выбрав такой маршрут: по горизонтали, начиная с левого верхнего угла поочередно слева направо и справа налево. Считывать же шифрованное сообщение по другому маршруту: по вертикали, начиная с правого верхнего угла и двигаясь поочередно сверху вниз и снизу вверх.

Зашифруем, например, таким способом фразу «пример маршрутной перестановки», используя прямоугольник размером 4 х 7 (рис. 5.1).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | *П* | *Р* | *И* | *М* | *Е* | *Р* | *М* |
| 2 | *Н* | *Т* | *У* | *Р* | *Ш* | *Р* | *А* |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | *О* | *Й* | *П* | *Е* | *Р* | *Е* | *С* |
| 4 | *И* | *К* | *В* | *О* | *Н* | *А* | *Т* |

***Рис. 5.1.*** *Пример шифра маршрутной перестановки*

Зашифрованная фраза выглядит так:

*МАСТАЕРРЕШРНОЕРМИУПВКЙТРПНОИ.*

Маршруты считывания зашифрованного сообщения в такой матрице могут быть значительно более изощренными, однако запутанность маршрутов усложняет использование таких шифров.

Можно, например, осуществлять считывание по диагоналям матрицы слева направо, начиная с левого верхнего угла. В этом случае получим зашифрованную фразу вида:

*ПНРОТИИЙУМКПРЕВЕШРОРРМНЕААСТ.*

Последняя строка при шифровании не всегда может быть заполнена до конца. В таком случае последняя строка не заполняется «нерабочими» буквами, так как это даст противнику, получившему в свое расположение криптограмму, сведения о длине ключа. Длиной ключа мог бы быть один из делителей длины сообщения.

При расшифровании такой криптограммы в первую очередь определяется число длинных (полностью заполненных) столбцов, то есть число букв в последней строке прямоугольника. Для этого число букв в сообщении следует разделить на длину числового ключа. Остаток от деления и будет искомым числом. Когда число длинных столбцов определено, можно заполнять таблицу буквами криптограммы.

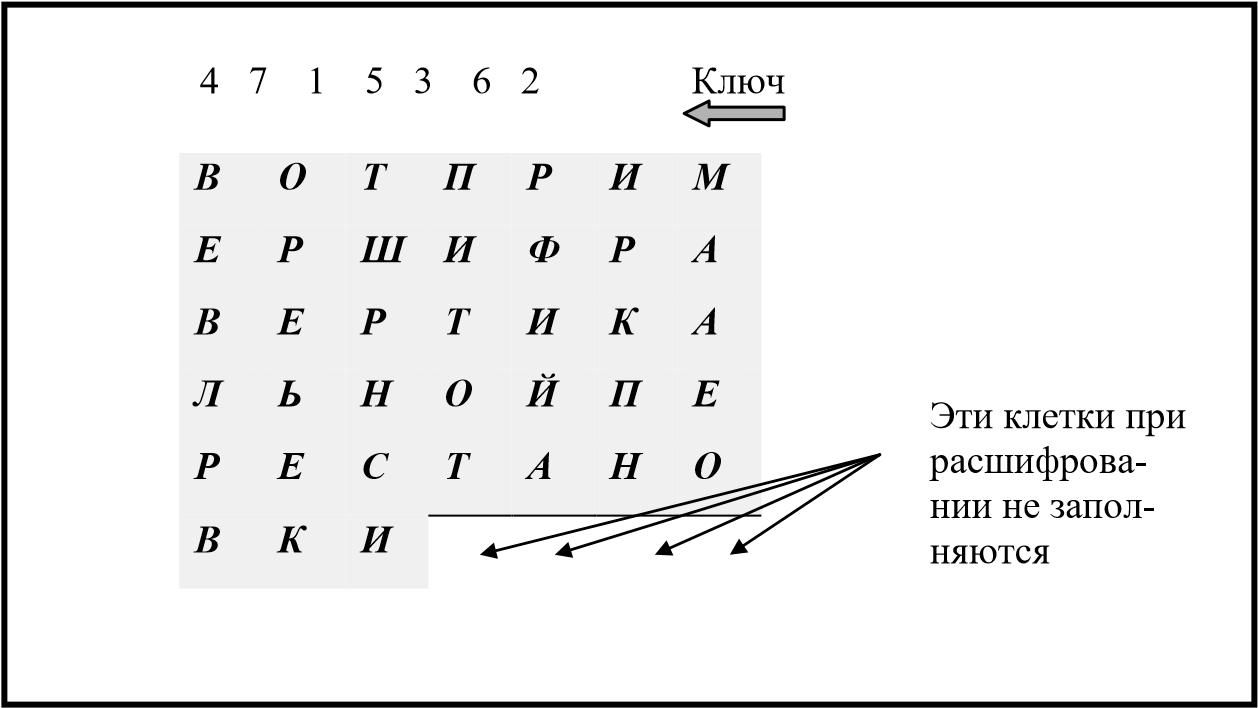
В примере, приведенном на рис. 5.2, использована прямоугольная таблица размером 6х7 и текст длиной 38 символов. Требуется расшифровать криптограмму:

ТШРНСИМААЕОРФИЙАВЕВЛРВПИТОТИРКПНОРЕЬЕК.

Сначала определяем количество длинных столбцов: 38 = 7∙5 + 3.Таким образом, в заполненной таблице имеется 3 длинных и 4 коротких столбца. Далее вписываем буквы криптограммы в таблицу согласно числовому ключу

(4715362) и с учетом длины столбцов: в третий столбик – первые 6 букв, в седьмой – следующие 5 букв, в пятый столбик – … и так далее до конца криптограммы. При построчном чтении получаем открытый текст:

ВОТ ПРИМЕР ШИФРА ВЕРТИКАЛЬНОЙ ПЕРЕСТАНОВКИ.



***Рис. 5.2.*** *Пример шифра вертикальной перестановки*

### 5.3. Шифры вертикальной перестановки

Широко распространена разновидность шифра маршрутной перестановки, называемая *шифром вертикальной перестановки*. В таком шифре реализуется перестановка считываемых столбцов матрицы в соответствии с ключом. Пусть, например, этот ключ будет таким: 5,4,1,7,2,6,3. Теперь, выбирая столбцы в порядке, заданном ключом, и считывая последовательно буквы каждого из них сверху вниз, следующую криптограмму:

*ЕШРНМРЕОПНОИМАСТРТЙКРРЕАИУПВ.*

Ключом может быть слово. Например, *«ОКТЯБРЬ».* Используя расположение букв этого ключа в алфавите, будем иметь набор чисел 3257146 – ключевую последовательность, аналогичную рассмотренной выше, но дающую новый шифртекст:

*ЕШРНРТЙКПНОИРРЕАИУПВМАСТМРЕО*.

Число ключей шифра вертикальной перестановки, независимо от способа задания, не более m!, где m – число столбцов таблицы шифрования.

### 5.4. Поворотные решетки

Частным случаем шифра маршрутной перестановки является также шифр «Поворотная решетка», в котором используется трафарет из прямоугольного листа клетчатой бумаги размером 2m x 2k клеток. В трафарете вырезано m x k клеток так, что при наложении его на чистый лист бумаги того же размера четырьмя возможными способами его вырезы полностью покрывают всю площадь листа.

Буквы сообщения последовательно вписываются в вырезы трафарета (по строкам, в каждой строке слева направо) при каждом из четырех его возможных положений в заранее установленном порядке.

Поворотные решетки называются также решетками Кардано по имени придумавшего их Дж. Кардано в XV веке, о чем говорилось уже в историческом обзоре.

Поясним процесс шифрования на примере. Пусть в качестве ключа используется решетка 6х10, показанная на рис. 5.3. Зашифруем с ее помощью текст «шифр решетка является частным случаем шифра маршрутной перестановки».

Наложив решетку на лист бумаги, вписываем первые 15 (по числу вырезов) букв сообщения. Сняв решетку, увидим на бумаге текст, представленный рис. 5.4.а. Поворачиваем решетку на 180º. В окошечках появятся новые, еще не заполненные клетки. Вписываем в них следующие 15 букв. Получится запись, приведенная на рис. 5.4.б. Затем переворачиваем решетку на другую сторону и зашифровываем остаток текста аналогичным образом (рис. 5.4.в,г).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

***Рис. 5.3.*** *Трафарет шифра «Поворотная решетка».*

Можно показать, что число возможных трафаретов (рис. 5.3), то есть количество ключей шифра «решетка», составляет Т = 4m·k. Уже при размерности решетки 8х8 число возможных решеток превосходит 4 миллиарда.

Если решетка квадратная, то возможен вариант самосовмещенной фигуры, а именно, последовательные повороты вокруг центра квадрата на 90о.

### 5.5. Задача Эйлера

Знаменитый математик Леонард Эйлер в 1759 году нашел замкнутый маршрут обхода всех клеток шахматной доски ходом коня ровно по одному разу. Если вписать в клетки шахматной доски открытый текст по такому маршруту, а затем выписать последовательность букв из полученной таблицы, например, построчно, получим образец шифртекста маршрутной перестановки с очень сложным маршрутом («ходом коня»).

Более сложные маршрутные перестановки могут использовать другие геометрические фигуры и более «хитрые» маршруты, например, при обходе шахматной доски «ходом коня», пути в некотором лабиринте и т.п. Фантазия составителя ограничивается при этом только требованием простоты использования такой шифрсистемы в процессах шифрования и расшифрования.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Ш*** |  |  |  |  |  |  |  |  |
| И |  |  |  | Ф |  | Р | Р |  |  |
|  | Е |  |  |  | Ш |  |  |  | Е |
|  |  |  | Т |  |  |  | К |  |  |
|  | А |  |  |  |  |  |  |  |  |
|  |  | Я |  |  | В | Л |  |  | Я |

а

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Е | ***Ш*** |  | Т | С |  |  | Я |  |  |
| И |  |  |  | Ф |  | Р | Р | Ч |  |
|  | Е | А |  |  | Ш | С |  |  | Е |
| Т |  |  | Т | Н |  |  | К | Ы |  |
|  | А | М | С |  | Л |  |  |  | У |
|  |  | Я |  |  | В | Л |  | Ч | Я |
|  |  |  |  |  |  |  |  |  |  |

б

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Е | ***Ш*** | А | Т | С | Е | М | Я |  | Ш |
| И | И |  |  | Ф |  | Р | Р | Ч |  |
|  | Е | А | Ф |  | Ш | С | Р |  | Е |
| Т | А |  | Т | Н | М |  | К | Ы | А |

в

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Р | А | М | С | Ш | Л | Р | У |  | У |
|  | Т | Я |  |  | В | Л |  | Ч | Я |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Е | ***Ш*** | А | Т | С | Е | М | Я | Н | Ш |
| И | И | О | Й | Ф | П | Р | Р | Ч | Е |
| Р | Е | А | Ф | Е | Ш | С | Р | С | Е |
| Т | А | Т | Т | Н | М | А | К | Ы | А |
| Р | А | М | С | Ш | Л | Р | У | Н | У |
| О | Т | Я | В | К | В | Л | И | Ч | Я |

г

***Рис. 5.4.*** *Последовательность использования поворотной решетки при шифровании*

### 5.6. Элементы криптоанализа шифров перестановки

#### Вскрытие вертикальных перестановок

В процессе шифрования вертикальными перестановками буквы каждого столбца заполненного прямоугольника выписываются в криптограмму подряд, то есть криптограмма разбивается на отрезки, являющиеся столбцами таблицы. Поэтому при дешифровании естественно попытаться соединить две группы последовательных букв криптограммы так, чтобы они образовывали хорошие с точки зрения обычного текста комбинации.

Для этого естественно использовать наиболее частые биграммы открытого текста, которые можно составить из букв рассматриваемого шифрованного текста рис. 5.5.

**СТ НО ЕН ТО НА ОВ НИ РА ВО КО**

**СТО ЕНО НОВ ТОВ ОВО ОВА**

***Рис. 5.5.*** *Наиболее частые биграммы и триграммы*

Если для первой пробы выбрано, скажем, сочетание *СТ* (самая частая биграмма русского языка), то можем по очереди приписывать к каждой букве *С* криптограммы каждую букву *Т* из нее. При этом несколько букв, стоящих до и после данной буквы С, и несколько букв, стоящих до и после данной буквы Т, соединяются в пары, то есть получаются два столбца букв, записанных рядом

(рис. 5.6).

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | **…**    ***С***    **…** |   **I** | |  | | --- | | **…**    ***Т***    ***…*** |   **II** |

***Рис. 5.6.*** *Два столбца букв, записанных рядом*

Конечно, длины столбцов неизвестны, но некоторые ограничения на них можно получить, используя положение конкретных букв. Так, столбцы должны иметь одинаковые длины или первый столбец может быть длиннее второго на одну букву, и тогда эта буква – последняя буква сообщения (рис. 5.7).

**…**

**…**

***С***

**…**

***Л***

***Й***

**…**

**…**

***Т***

***…***

***Б***

*Последняя буква сообщения*

***Рис. 5.7.*** *Определение последней буквы сообщения*

Если приписываемые друг к другу буквы разделены, скажем, только двумя буквами, то, как легко видеть, мы можем составить в соседних столбцах не более трех пар, и длина столбца не превышает четырех. Кроме того, ограничением может послужить появление запретной биграммы (например, гласная – мягкий знак, как показано на рис. 5.8).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **…**  ***С***  **…**  ***И*** | | |  |  |  | | --- | --- | --- | | |  | | --- | | **…**  ***Т***  ***…***  ***Ь*** | |  | | *запретная биграмма* |

***Рис. 5.8.*** *Пример запретной биграммы*

Для выбранного сочетания *СТ* получается по одной паре столбцов для каждого конкретного выбора букв *С* и *Т* из криптограммы, и из них целесообразно отобрать ту пару, которая содержит наиболее частые биграммы.

При автоматизации этого процесса можно приписать каждой биграмме вес, равный частоте ее появления в открытом тексте. Тогда следует отобрать ту пару столбцов, которая имеет наибольший вес. Кстати, появление одной биграммы с низкой частотой может указать на то, что длину столбца надо ограничить по длине. Выбрав пару столбцов, мы аналогичным образом можем подобрать к ним третий (справа или слева) и т.д.

Описанная процедура значительно упрощается при использовании вероятных слов, то есть слов, которые могут встретиться в тексте с большой вероятностью.

#### Вскрытие произвольных шифров перестановки

Допустим, что к двум или более сообщениям (или отрезкам сообщений) одинаковой длины применяется один и тот же шифр перестановки. Тогда буквы, которые находились на одинаковых местах в открытых текстах, окажутся на одинаковых местах и в шифрованных текстах.

Выпишем шифрованные сообщения одно под другим так, что первые буквы всех сообщений окажутся в первом столбце, вторые – во втором и т.д. Если предположить, что две конкретные буквы в одном из сообщений идут одна за другой в открытом тексте, то буквы, стоящие на тех же местах в каждом из остальных сообщений, соединятся подобным же образом.

Это означает, что они могут служить проверкой правильности первого предположения, подобно тому, как комбинации, которые дают два столбца в системе вертикальной перестановки, позволяют проверить, являются ли соседними две конкретные буквы из этих столбцов.

К каждому из указанных двух сочетаний можно добавить третью букву для образования биграммы и т.д. Если располагать не менее чем четырьмя сообщениями одинаковой длины, то можно с уверенностью гарантировать их вскрытие подобным образом.

#### Вопросы и задания для самостоятельной работы

1. Какое правило лежит в основе всех шифров перестановки?
2. Что является ключом шифра перестановки?
3. Назовите основной недостаток шифров перестановки.
4. Приведите пример шифра перестановки, который может рассматриваться и как блочный шифр замены?
5. Как определить по криптограмме, полученной с помощью шифра вертикальной перестановки, число коротких столбцов заполненного открытым текстом основного прямоугольника?
6. Какие свойства открытого текста используются при вскрытии шифра перестановки?
7. Сколько решений имеет шахматная задача Эйлера?

## ТЕМА 6. ШИФРЫ ЗАМЕНЫ

Шифры замены (или подстановки) – это класс шифров, в которых каждой букве алфавита ставится в соответствие шифробозначение: буква, цифра, символ или какая-нибудь их комбинация.

Если каждой букве открытого алфавита соответствует одно шифробозначение, то такая замена называется *однозначной*. Если соответствие предполагает несколько шифробозначений для каждой буквы открытого алфавита, например, в зависимости от частоты повторений в тексте, то такая замена называется *многозначной,* а шифры – *омофонами*.

Если в системе шифрования используется один алфавит для зашифрования, то шифр замены называется *одноалфавитным шифром замены*. В противном случае – *многоалфавитным шифром замены*.

Если зашифрование букв открытого текста происходит по одной по порядку их в тексте, то такая замена называется *поточным шифром замены*. Если текст предварительно разделяется на блоки, которые затем зашифровываются как отдельный элемент, то такая замена называется *блочным шифром замены*.

### 6.1. Поточные шифры простой замены

*Шифрами простой замены* называются одноалфавитные однозначные замены*.* Наибольшее распространение получили *поточные шифры простой замены,* множества шифрвеличин и шифробозначений которых совпадают с алфавитом открытого теста. Рассмотрим самые известные из них.

#### Атбаш

Одним из первых шифров был шифр простой замены «АТБАШ». Некоторые фрагменты библейских текстов зашифрованы с помощью этого шифра. Правило зашифрования состояло в замене i-й буквы алфавита буквой с номером n – i + 1, где n – число букв в алфавите. Происхождение слова АТБАШ объясняется принципом замены букв. Это слово составлено из букв *Алеф, Тае, Бет, Шит*, то есть первой и последней, второй и предпоследней букв древнесемитского алфавита.

Алгоритм шифрования:

**Yi = X(n-i +1)**

где Y – исходный (открытый) текст; X(n-i +1) – зашифрованный текст.

Рассмотрим этот способ шифрования на примере пословицы:

#### ВЫПУЩЕННОЕ СЛОВО И КАМЕНЬ НЕ ИМЕЮТ ВОЗВРАТА

Таблица зашифрования первой половины алфавита:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Y** | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П |
| **X** | Я | Ю | Э | Ь | Ы | Ъ | Щ | Ш | Ч | Ц | Х | Ф | У | Т | С | Р |
| **i** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |

Таблица зашифрования второй половины алфавита:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Y** | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я |
| **X** | П | О | Н | М | Л | К | Й | И | З | Ж | Е | Д | Г | В | Б | А |
| **i** | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |

Зашифрованный текст:

**ЭДРМЖЪТТСЪОФСЭСЧ ХЯУЪТГТЪЧУЪБН СШЭПЯНЯ**

***Литорея*** (от лат. *littera* – буква) – тайнописание, род шифрованного письма, употреблявшегося в древнерусской рукописной литературе. Известна литорея двух родов: *простая* и *мудрая.* Простая, иначе называемая тарабарской грамотой, заключается в следующем: поставив согласные буквы в два ряда, в порядке:

б в г д ж з к л м н щ ш ч ц х ф т с р п

Употребляют в письме верхние буквы вместо нижних и наоборот, причем гласные остаются без перемены; так, например, *лсошамь* = *словарь* и т.п. *Мудрая литорея* предполагает более сложные правила подстановки. В разных дошедших до нас вариантах используются подстановки целых групп букв, а также числовые комбинации: каждой согласной букве ставится в соответствие число, а потом совершаются арифметические действия над получившейся последовательностью чисел (например, ко всем прибавляется некоторая константа – ключ).

По своей сути литорея является шифром простой замены, который легко дешифруется современными методами.

Наиболее простой разновидностью шифрования является прямая (простая) замена символов шифруемого сообщения другими символами того же самого или некоторого другого алфавита. Ключом такого преобразования является соответствующая таблица замены, устанавливающая однозначное соответствие между символами исходного и шифрующего алфавитов. Если оба алфавита состоят из одного и того же набора символов, например букв русского алфавита, то их взаимно однозначное соответствие устанавливается таблицей замены, имеющей следующий вид (табл. 6.1).

*Таблица 6.1*

**Таблица простой замены**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер символа | 1 | 2 | 3 | 4 | | 5 | | 6 | | 7 | | 8 | | 9 | | 10 | | 11 | | 12 | | 13 | 14 | 15 |
| Исходные символы | А | Б | В | Г | | Д | | Е | | Ж | | З | | И | | К | | Л | | М | | Н | О | П |
| Символы замены | Э | М | А | Ы | | С | | П | | Г | | Ъ | | У | | Р | | Я | | Ч | | В | Ф | Е |
| Номер символа | 29 | 12 | 1 | 27 | | 17 | | 15 | | 4 | | 26 | | 19 | | 16 | | 31 | | 23 | | 3 | 20 | 6 |
| Номер символа | 16 | 17 | 18 | 19 | 20 | | 21 | | 22 | | 23 | | 24 | | 25 | | 26 | | 27 | | 28 | 29 | 30 | 31 |
| Исходные символы | Р | С | Т | У | Ф | | Х | | Ц | | Ч | | Ш | | Щ | | Ъ | | Ы | | Ь | Э | Ю | Я |
| Символы замены | И | Н | Ш | Ю | Щ | | Т | | Ж | | Х | | Д | | К | | З | | Ц | | Л | Б | О | Ь |
| Номер символа | 9 | 13 | 24 | 30 | 25 | | 18 | | 7 | | 21 | | 5 | | 10 | | 8 | | 22 | | 11 | 2 | 14 | 28 |

Символы замены в данной таблице имеют произвольный порядок следования букв данного алфавита. Поэтому всегда стремятся установить определенные правила, пользуясь которыми можно достаточно просто восстановить вторую строку в таблице замены. В известном из истории шифре Цезаря эта строка образовывалась путем циклического сдвига символов исходного алфавита на определенное число позиций. Именно этот сдвиг устанавливал ключ шифрования.

Шифрование символов исходного алфавита обычно осуществляется путем преобразования числовых эквивалентов символов этого алфавита, в качестве которых используются их исходные порядковые номера в алфавите. Если, например, по табл. 6.1 производится шифрование методом замены слова <УНИВЕРСИТЕТ>, то получим шифрованную последовательность символов <ЮВУАПИНУШПШ>. Этому соответствует преобразование исходного числового кортежа <19,13,9,3,6,16,17,9,18,6,18> в числовой кортеж

<30,3,19,1,15,9,13,19,24,15,24>.

В шифре Цезаря номера шифрованной последовательности символов вычисляются путем суммирования по модулю R номеров исходной последовательности с величиной сдвига, где R – число символов используемого алфавита. Очевидно, что число различных ключей в шифре Цезаря равно числу R символов в алфавите, поэтому путем перебора различных вариантов сдвига можно однозначно восстановить по перехваченному шифру исходное сообщение (последовательность символов в используемом алфавите предполагается известной).

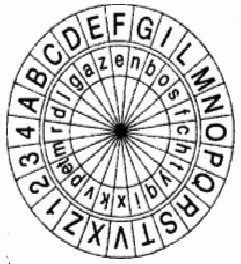
Существенно большее число вариантов ключа обеспечивает так называемый *лозунговый шифр*, также относящийся к шифрам замены. В этом шифре начальные позиции символов замены заполняются буквами легко запоминаемого слова-лозунга, в котором не должно быть повторяющихся букв (например, *футбол, учебник* и др.). В оставшиеся позиции строки замены вписываются в алфавитном порядке буквы используемого алфавита, не вошедшие в словолозунг.

Основным недостатком метода прямой одноалфавитной замены является наличие одних и тех же статистических характеристик появления букв как в исходном, так и закрытом тексте. Зная, на каком языке написан исходный текст и частотную характеристику использования символов этого языка, криптоаналитик путем статистической обработки перехваченных сообщений может установить соответствие между символами обоих алфавитов.

### 6.2. Шифры многоалфавитной замены

Существенно более стойкими являются шифры многоалфавитной замены, использующие несколько алфавитов для замены символов исходного текста. Формально многоалфавитную замену можно представить следующим образом. При N-алфавитной замене символ S01 из исходного алфавита А0 заменяется символом S11 из алфавита А1, S02 заменяется символом S22 из алфавита А2 и так далее. После замены S0N символом SNN из алфавита АN символ S0(N+1) замещается символом S1(N+1) из алфавита А1 и так далее.

#### Шифровальный диск Альберти



***Рис. 6.2.*** *Шифровальный диск Альберти*

Шифровальный диск представлял собой пару соосных дисков разного диаметра (см. рисунок). Больший из них – неподвижный, его окружность разделена на 24 равных сектора, в которые вписаны 20 букв латинского алфавита в их естественном порядке и 4 цифры (от 1 до 4). При этом из 24-буквенного алфавита были удалены 4 буквы, без которых можно было обойтись, подобно тому, как в русском языке обходятся без Ъ, Е, И. Меньший диск – подвижный, по его окружности, разбитой также на 24 сектора, были вписаны все буквы смешанного латинского алфавита.

Имея два таких прибора, корреспонденты договаривались о первой индексной букве на подвижном диске. При шифровании сообщения отправитель ставил индексную букву против любой буквы большего диска. Он информировал корреспондента о таком положении диска, записывая эту букву внешнего диска в качестве первой буквы шифртекста. Очередная буква открытого текста отыскивалась на неподвижном диске и стоящая против нее буква меньшего диска являлась результатом ее зашифрования. После того как были зашифрованы несколько букв текста, положение индексной буквы изменялось, о чем также каким-либо образом передавалось корреспонденту.

Такой шифр имел две особенности, которые делают изобретение Альберти событием в истории криптографии.

Во-первых, в отличие от шифров простой замены шифровальный диск использовал не один, а несколько алфавитов для зашифрования. Такие шифры получили название *многоалфавитных*.

Во-вторых, шифровальный диск позволял использовать так называемые коды с перешифрованием, которые получили широкое распространение лишь в конце XIX века, то есть спустя четыре столетия после изобретения Альберти. Для этой цели на внешнем диске имелись цифры. Альберти составил код, состоящий из 336 кодовых групп, занумерованных от 11 до 4444. Каждому кодовому обозначению соответствовала некоторая законченная фраза. Когда такая фраза встречалась в открытом сообщении, она заменялась соответствующим кодовым обозначением, а с помощью диска цифры зашифровывались как обычные знаки открытого текста, превращаясь в буквы.

#### Таблица Тритемия

Шифралфавиты записаны в строки таблицы один под другим, причем каждый из них сдвинут на одну позицию влево по сравнению с предыдущим. Тритемий предлагал использовать эту таблицу для многоалфавитного зашифрования самым простым из возможных способов: первая буква текста шифруется первым алфавитом, вторая буква – вторым и т.д. В этой таблице не было отдельного алфавита открытого текста, для этой цели служил алфавит первой строки. Таким образом, открытый текст, начинающийся со слова *«FIGHT...*» (борьба), приобретал вид «*FJIKH.*..».

Преимущество этого метода шифрования по сравнению с методом Альберти состоит в том, что с каждой буквой задействуется новый алфавит. Альберти менял алфавиты лишь после трех или четырех слов. Поэтому его шифртекст состоял из отрезков, каждый из которых обладал закономерностями открытого текста, которые помогали вскрыть криптограмму. Побуквенное зашифрование не дает такого преимущества.

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

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

1. F G H I K L M N O P Q R S T U X Y Z W A B C D
2. G H I K L M N O P Q R S T U X Y Z W A B C D E
3. H I K L M N O P Q R S T U X Y Z W A B C D E F
4. I K L M N O P Q R S T U X Y Z W A B C D E F G
5. K L M N O P Q R S T U X Y Z W A B C D E F G H
6. L M N O P Q R S T U X Y Z W A B C D E F G H I
7. M N O P Q R S T U X Y Z W A B C D E F G H I K
8. N O P Q R S T U X Y Z W A B C D E F G H I K L
9. O P Q R S T U X Y Z W A B C D E F G H I K L M
10. P Q R S T U X Y Z W A B C D E F G H I K L M N
11. Q R S T U X Y Z W A B C D E F G H I K L M N O
12. R S T U X Y Z W A B C D E F G H I K L M N O P
13. S T U X Y Z W A B C D E F G H I K L M N O P Q
14. T U X Y Z W A B C D E F G H I K L M N O P Q R
15. U X Y Z W A B C D E F G H I K L M N O P Q R S
16. X Y Z W A B C D E F G H I K L M N O P Q R S T
17. Y Z W A B C D E F G H I K L M N O P Q R S T U
18. Z W A B C D E F G H I K L M N O P Q R S T U X
19. W A B C D E F G H I K L M N O P Q R S T U X Y

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

Рис. 6.3. Таблица Тритемия

Шифр Тритемия является также первым нетривиальным примером *периодического шифра*. Так называется многоалфавитный шифр, правило зашифрования которого состоит в использовании периодически повторяющейся последовательности простых замен.

#### Шифр Белазо

Большое распространение получил алгоритм многоалфавитной замены с использованием пароля и таблицы Тритемия, которая представляет собой квадратную матрицу размерностью [R x R], где R – количество символов в используемом алфавите. В первой строке располагаются символы в алфавитном порядке. Начиная со второй строки символы записываются со сдвигом влево на одну позицию. Выталкиваемые символы заполняют освобождающиеся позиции справа (циклический сдвиг). Если русский алфавит, приведенный в табл. 6.1, дополнить символом пробела └─┘, то таблица Тритемия ТТ будет иметь размерность [32 x 32] (рис. 6.4).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | А | Б | В | Г | Д | Е | Ж | З | И | . | . | . | Ь | Э | Ю | Я | └─┘ |
|  | Б | В | Г | Д | Е | Ж | З | И | К | . | . | . | Э | Ю | Я | └─┘ | А |
| ТВ= | В | Г | Д | Е | Ж | З | И | К | Л | . | . | . | Ю | Я | └─┘ | А | Б |
|  | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
|  | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
|  | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
|  | └─┘ | А | Б | В | Г | Д | Е | Ж | З | . | . | . | Ы | Ь | Э | Ю | Я |

***Рис. 6.4.*** *Таблица Тритемия для русского алфавита*

Шифрование осуществляется с помощью пароля, состоящего из М неповторяющихся символов. Из полной матрицы ТТ выделяется матрица ТШ размерностью [(М+1) х R]. Она включает первую строку и строки, первые элементы которых совпадают с символами ключа. Если в качестве ключа выбрано слово <ЗОНД>, то матрица шифрования содержит пять строк (рис. 6.5).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | А | Б | В | Г | Д | Е | Ж | З | И | К | . | . | . | Э | Ю | Я | └─┘ |
|  | З | И | К | Л | М | Н | О | П | Р | С | . | . | . | Г | Д | Е | Ж |
| ТВ= | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | . | . | . | К | Л | М | Н |
|  | Н | О | П | Р | С | Т | У | Ф | Х | Ц | . | . | . | И | К | Л | М |
|  | Д | Е | Ж | З | И | К | Л | М | Н | О | . | . | . | А | Б | В | Г |

***Рис. 6.5.*** *Матрица шифрования для ключа <ЗОНД>*

Замена с использованием пароля эквивалентна простой замене с циклическим изменением алфавита. При этом в каждом цикле мы имеем многоалфавитную подстановку с числом используемых алфавитов, соответствующим числу букв в слове ключа. Это пример *периодического шифра многозначной замены*.

При шифровании необходимо вначале записать под буквами шифруемого текста буквы ключевого слова. Ключ при этом повторяется необходимое число раз. Символ шифруемого текста определяет столбец матрицы шифрования. Необходимый для его замены символ находится на пересечении этого столбца со строкой, соответствующей букве ключа, записанного под шифруемым текстом.

На рис. 6.6. приведен пример шифрования с паролем слова <КРИПТОГРАФИЯ>. Процесс шифрования осуществляется следующим образом:

* под шифруемым словом записываем нужное число раз ключевое слово <ЗОНД>;
* берем первую букву шифруемого слова (К) и соответствующую ей букву ключа (3);
* по букве К входим в соответствующий столбец матрицы шифрования

(рис. 6.6);

* выбираем в этом столбце букву, расположенную в строке, соответствующей букве ключа (3).

В нашем примере такой буквой является буква С, которая помещается в шифрованный текст в качестве символа замены исходной буквы К. Данная процедура циклически повторяется до завершения шифрования всего слова. В результате получает шифрованное слово <СЭХУЩЫРФЗАХВ>.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Исходное слово | К | Р | И | П | Т | О | Г | Р | А | Ф | И | Я |
| Ключ | З | О | Н | Д | З | О | Н | Д | З | О | Н | Д |
| Шифрованное слово методом замены | ***С*** | ***Э*** | ***Х*** | ***У*** | ***Щ*** | ***Ы*** | ***Р*** | ***Ф*** | ***З*** | ***А*** | ***Х*** | ***В*** |

***Рис. 6.6.*** *Пример шифрования с помощью пароля Белазо*

Криптостойкость методов многоалфавитной замены значительно выше методов простой замены, так как одни и те же символы исходной последовательности могут заменяться разными символами. Однако стойкость шифра к статистическим методам криптоанализа зависит от длины ключей.

#### Шифр Виженера

Наибольшее распространение получил алгоритм многоалфавитной замены Виженера. Шифрование осуществляется по таблицам Тритемия с паролем, в качестве которого используется сам текст сообщения (или шифрованный текст). Позже такая таблица стала называться *таблицей Виженера*. Первая строка служила открытым алфавитом, а первый столбец – алфавитом ключа. В качестве ключевой последовательности использовался открытый текст с добавлением к нему первой буквы, известной и отправителю, и получателю. Такой механизм шифрования явился прообразом *шифров гаммирования*, широко используемых в настоящее время (шифр подробно рассмотрен в п. 7.2).

С целью повышения криптостойкости можно использовать усовершенствованные варианты таблицы шифрования Виженера. Приведем некоторые из них:

* во всех (кроме первой) строках матрицы буквы располагаются в произвольном порядке;
* в качестве ключа используются случайные последовательности чисел.

### 6.3. Биграммные и n-граммные шифры замены

Более эффективны обобщения подстановки Цезаря – шифр Порта, рассмотренный в историческом обзоре, шифр Хилла и шифр Плэйфера. Они основаны на подстановке не отдельных символов, а *биграмм (шифр Плэйфера) или n-грамм (шифр Хилла)*. При более высокой криптостойкости они значительно сложнее для реализации и требуют достаточно большого количества ключевой

информации.

#### Шифр Плэйфера

Биграммный шифр Плэйфера (Playfair, Великобритания), применявшийся Великобританией во время Первой мировой воны, был основан на лозунговом способе заполнения шифртаблицы.

Способ шифрования состоял в разбивке входного текста на биграммы. Переход от биграмм входного текста к биграммам выходного текста осуществляется по следующим правилам: если буквы входной биграммы оказались в одном столбце таблицы, шифрование и расшифровка происходит в соответствии с алгоритмом Полибия; если же буквы входной биграммы оказались в одной строке таблицы, то шифрование осуществляется слева направо, а дешифрование – наоборот.

Буквы биграмм могут совпадать. Если буквы входной биграммы оказались в разных столбцах и строках таблицы, то рисуется воображаемый прямоугольник, а выходная биграмма берется как его альтернативные вершины.

Для приведенной на рис. 6.7 таблицы открытый текст *ПУСТЬ КОНСУЛЫ БУДУТ БДИТЕЛЬНЫ,* записанный без пробелов:

***ПУСТЬКОНСУ ЛЫБУ ДУТБ ДИТЕ ЛЬНЫ,***

имеет шифртекст:

#### УБ РХ ЫИ ДО ПБ КЩ РБ НР ШР ЖЛ ФР ИЩ ЗЮ



***Рис. 6. 7.*** *Пример использования шифра Плэйфера (движение некоторых букв показано стрелками)*

#### N-граммная подстановка

Вместо подстановки одной буквы можно использовать подстановку биграмм, триграмм и т.д. Для биграммной подстановки в общем виде требуется ключ, состоящий из перестановок 262 биграмм. Он может быть представлен с помощью таблицы, в которой ряд соответствует первой букве биграммы, а столбец – второй букве, причем клетки таблицы заполнены заменяющими символами (обычно также биграммами).

#### Шифр Виженера с перемешанным один раз алфавитом

Такой шифр представляет собой простую замену с последующим применением шифра Виженера:



«Обратным» к такому шифру является шифр Виженера с последующей простой заменой:



### 6.4. Матричные шифры

Имеется еще один метод подстановки n-грамм, который заключается в применении к последовательным n-граммам некоторой матрицы, имеющей обратную. Предполагается, что буквы занумерованы от 0 до 25 и рассматриваются как элементы некоторого алгебраического кольца. Если к n-грамме сообще-

ния применить матрицу , то получится n-грамма криптограммы:



Матрица является ключом, и расшифровка выполняется с помощью обратной матрицы. Обратная матрица будет существовать тогда и только тогда,

когда определитель имеет обратный элемент в нашем кольце.

#### Аналитические методы шифрования

качестве матричного шифрования информации могут использоваться аналитические преобразования, основанные на преобразованиях матричной алгебры.

Шифрование k-го блока исходной информации, представленного в виде вектора Bk = ║bj║, осуществляется путем перемножения этого вектора на матрицу А = ║aіј║, используемую в качестве ключа. В результате перемножения получается блок шифротекста в виде вектора Сk = ║сі║, где элементы вектора Сk определяются по формуле:

.

Сi j aijbj

Приведем пример, взяв в качестве ключа квадратную матрицу третьего по-

1 4 8

рядка: А 3 7 2.

6 9 5

Пусть требуется с помощью этого ключа зашифровать слово

Т0 *= <****ЗАБАВА****>.*

Дляэтого необходимо выполнить следующие шаги.

1. Определяется числовой эквивалент ТЭ исходного слова как последовательность соответствующих порядковых номеров букв этого слова в алфавите:

ТЭ = <8,1,2,1,3,1>.

1. Умножается ключевая матрица А на векторы В1 = {8,1,2} и В2 = {1,3,1}:

.

38

26

21

1

3

1

5

6

9

2

3

7

1

4

8

2

С

;

67

35

28

2

1

8

9

5

6

2

3

7

1

8

4

1

С













1. Зашифрованное слово записывается в виде последовательности чисел

Т1 = <***28,35,67,21,26,38***>.

Расшифрование осуществляется также с использованием правил матричной алгебры, только в качестве ключа берется матрица, обратная той, с помощью которой осуществлялось шифрование, а в качестве вектора-сомножителя – соответствующие фрагменты символов зашифрованного текста. Тогда компонентами вектора-результата будут цифровые эквиваленты букв открытого текста.

Для расшифрования полученной в предыдущем примере последовательности чисел Т1 необходимо выполнить следующие шаги.

1. Вычисляется определитель матрицы-ключа │А │ = -115.

17 -3 -15

1. Находится присоединенная матрица А52 -43 15, каждый элемент

-48 22 -5

аij которой является алгебраическим дополнением элемента аij матрицы-ключа

А, то есть аij   1 i jΔij, где Δij – определитель матрицы, получаемой вычеркиванием i-й строки и j-ого столбца исходной матрицы А.

17 52 -48

1. Получается транспонированная матрица А*m* -3 -43 22, элементы

-15 15 -5

которой аijm  аji.

|  |  |  |
| --- | --- | --- |
| -17 / 115  4. Вычисляется обратная матрица А1 3 / 115  15 / 115  менты которой аij-1  аij / A . 5. Определяются векторы В1 = А-1С1 и В2 = А-1С2: | -52 / 115  43 / 115  -15 / 115 | 48 / 115  -22 / 115, эле-  5 / 115 |

-17 / 115 -52 / 115 48 / 115

1

1

38

21

2

8

67

28

В 3 / 115 43 / 115 -22 / 115  35  1 ;

1

15 / 115 -15 / 115 5 / 115

-17 / 115 -52 / 115 48 / 115

В 3 / 115 43 / 115 -22 / 115  26  3 .

2

15 / 115 -15 / 115 5 / 115

6. Числовой эквивалент расшифрованного слова ТЭ = <8,1,2,1,3,1> заменяется соответствующими символами алфавита, в результате чего получается исходное слово Т0 = <ЗАБАВА>.

### 6.5. Модель шифров замены

Определим модель A = (X, K, Y, E, D) произвольного шифра замены. Будем считать, что открытые и шифрованные тексты являются словами в алфавитах A и B соответственно: X A\*, Y B\*, |A| = n, |B| = m. Здесь и далее C\* обозначает множество слов конечной длины в алфавите C.



Перед зашифрованием открытый текст предварительно представляется в виде последовательности подслов, называемых *шифрвеличинами*. При зашифровании шифрвеличины заменяются некоторыми их эквивалентами, которые назовем *шифробозначениями*. Как шифрвеличины, так и шифробозначения представляют собой слова из A\* и B\* соответственно.

Пусть U = {u1,...,uN} – множество возможных шифрвеличин, V =

{v1,...,vM} – множество возможных шифробозначений. Эти множества должны быть такими, чтобы любые тексты x X, y Y можно было представить словами из U\*, V\* соответственно. Требование однозначности расшифрования влечет неравенства N >= n, M >= m, M >= N.



Для определения правила зашифрования Ek(x) в общем случае нам понадобится ряд обозначений и понятие распределителя, который, по сути, и будет выбирать в каждом такте шифрования замену соответствующей шифрвеличине. Поскольку M >= N, множество V можно представить в виде объединения V = i=1NVi непересекающихся непустых подмножеств Vi. Рассмотрим произвольное семейство, состоящее из r таких разбиений множества V: V = i=1NV (i),



= 1,..,r, r N.



Заметим, что последнее обозначение представляется неточным, поскольку речь идет уже не о представлении множества V, а о множестве таких представлений. Кроме того, согласно тексту, N не отрезок натурального ряда, а число, что препятствует записи r N. Вследствие изложенного семейство разбиений предпочтительно записать в виде { i=1NV (i)}, = 1,..,r, r {1,2,..,N}, i=1NV (i) = V, V (i) V (j) = , V (i) != .



Семейство биекций, соответствующее этому семейству разбиений множества V, обозначим через { }: {V (1),..,V (N)} и будем предполагать, что



(ui) = V (i), i = 1,..,N.

Рассмотрим также произвольное отображение : K x N Nr\*,



где Nr = {1,2,..,r}, такое, что для любых k K, l N выполнено (k,l) = a1(k)...al(k), aj(k) Nr, j=1,..r.



Назовем последовательность (k,l) *распределителем*, отвечающим данным значениям k K, l N.



Теперь мы сможем определить правило зашифрования произвольного

шифра замены. Пусть x X, x = x1...xl, xi U, i = 1,..,l, k K и (k,l) = a1(k)...al(k). Тогда Ek(x) = y, где y = y1...yl, yi aj(k)(xj), j = 1..l.



В качестве yj можно выбрать любой элемент множества aj(k)(xj). Всякий раз при шифровании этот выбор можно производить случайно. Подчеркнем, что такая многозначность при зашифровании не препятствует расшифрованию, т.к. V (i) V (j) = при i != j.



### 6.6. Классификация шифров замены

В модели произвольного шифра замены, приведенной выше, правило зашифрования Ek(x) является *многозначной функцией*. Выбор ее значений представляет собой некоторую проблему, которая делает многозначные функции Ek(x) не слишком удобными для использования. Избавиться от этой проблемы позволяет использование однозначных функций, что приводит к естественному разделению всех шифров замены на *однозначные* и *многозначные замены* (называемые также *омофонами*).

Для однозначных шифров справедливо свойство: для любых , i: |V(i)| = 1, для многозначных шифров – существуют , i: |V (i)| > 1.



Наибольшее применение получили шифры однозначной замены. К ним, в частности, относится и рассмотренный ниже шифр *гаммирования*, играющий большую роль как в классической, так и в современной криптографии. Далее шифры замены будем считать однозначными. Тогда M = N и a(ui) = va,i, i = 1...M.

Заметим, что правило зашифрования Ek естественным образом индуцирует отображение k: U V, которое в свою очередь продолжается до отображения k: U\* V\*. Для упрощения записи будем использовать одно отображение Ek для



каждого из трех указанных отображений.

В силу инъективности (по k) отображения Ek и того, что |U| = |V|, введенные в общем случае отображения являются биекциями : U V, V U, определенными равенствами (ui) = v (i), i = 1..N, = 1..r. Число таких биекций не превосходит N!.



|  |
| --- |
| Функция  называется биекцией, если она: |
| 1. Переводит разные элементы множества *X* в разные элементы множества *Y* (инъективность). 2. Любой элемент из имеет свой прообраз (сюрьективность). Иными словами, . |
| Биекцию также называют взаимно однозначным отображением. |

Для шифра однозначной замены определение правила зашифрования можно уточнить: в формуле yj j(k)(xj), j=1..l включение следует заменить равенством yj = j(k)(xj), j=1..l.



Введем еще ряд определений.

Если для некоторого числа q N выполняются включения vi Bq, i = 1...N, то соответствующий шифр будем называть *шифром равнознач-*

*ной замены*.

Иначе говоря, шифр равнозначной замены имеет шифробозначения из одинакового количества символов, шифр разнозначной замены, вообще говоря, – из разного количества. Например, шифр Цезаря, сопоставляющий каждой букве ровно одну букву, – равнозначный. А шифр, сопоставляющий некоторым буквам однозначное число, а другим двузначное – разнозначный.

В подавляющем большинстве случаев используются шифры замены, для которых U Ap, для некоторого p N. При p = 1 говорят о *поточных шифрах замены*, при p > 1 – о *блочных шифрах замены*. Таким образом, поточные шифры замены предусматривают только однобуквенные шифрвеличины, блочные – многобуквенные.

Наконец, в случае r = 1 шифр замены называют *одноалфавитным шифром замены*, в противном случае – *многоалфавитным шифром замены*.

Для однозначных замен эти термины следует употреблять с осторожностью, т.к. в этом случае алфавит шифробозначений – единственный с точностью до его нумерации, т.е. до порядка следования шифробозначений. Если, например, шифрвеличины и шифробозначения – буквы русского алфавита, то распределитель  указывает, в каком порядке следует расположить буквы для замены каждой шифрвеличины сообщения. Если для двух одинаковых шифрвеличин, расположенных в разных местах открытого текста, распределитель дает разные порядки букв алфавита шифробозначений, то одно и то же правило замены j переведет эти шифрвеличины в разные шифробозначения.

### 6.7. Криптоанализ поточного шифра простой замены

Метод вскрытия шифра простой замены основан на том, что частотные характеристики m-грамм шифртекста и открытого текста одинаковы. При этом существенно используются априорные частотные характеристики предполагаемого открытого текста, получаемые с учетом «характера переписки».

Такие характеристики являются более «рельефными» для литературных текстов и менее для формализованных электронных текстов. Чем менее рельефно распределение знаков текста, тем сложнее задача вскрытия шифра простой замены, для открытых текстов с «почти равномерным» распределением знаков эта задача становится практически не решаемой.

Таким образом, вскрытие шифров простой замены может оказаться задачей весьма сложной. Методы «рандомизации» или «сжатия» открытых текстов, например с использованием компьютерных архиваторов, значительно усложняют задачу вскрытия шифра простой замены.

Рельефность диаграммы текста тесно связана с такой его важной теоретико-информационной характеристикой, как *избыточность*. Рассмотрим решение задачи вскрытия простой замены лишь при условии, что предполагаемые открытые тексты – это литературные тексты с избыточностью. Кроме того, будем считать, что при дешифровании мы располагаем достаточно большим числом знаков шифртекста, чтобы опираться на «статистику».

Алгоритм вскрытия простой замены по тексту криптограммы достаточно сложно формализовать. При любой попытке формализации может быть потерян какой-нибудь важный нюанс. Поэтому рассмотрим лишь основные этапы алгоритма вскрытия шифра простой замены.

#### Основные этапы алгоритма вскрытия шифра простой замены

1. Подсчет частот встречаемости шифробозначений, а также некоторых их сочетаний, например биграмм и триграмм подряд идущих знаков.
2. Выявление шифробозначений, заменяющих гласные и согласные буквы.
3. Выдвижение гипотез о значениях шифробозначений и их проверка.
4. Восстановление истинного значения шифробозначений.

Если длина текста достаточно велика, то найденные на этапе 1 частоты окажутся близкими к табулированным значениям частот знаков (соответственно – биграмм или триграмм). Проведенная на этом этапе работа служит основанием для выдвижения гипотез о значениях шифрвеличин, соответствующих данным шифробозначениям. При этом учитывается, что каждая буква имеет группу предпочтительных связей (см. рис. 6.7), которые составляют ее наиболее характерную особенность.

Как правило, такие гипотезы подтверждаются не полностью. Хорошим критерием при этом является «читаемость» восстанавливаемого открытого текста. Выделение шифробозначений, отвечающих гласным и согласным, основано на характерных свойствах этих букв, указанных на рис. 6.8. Кроме того, необходимо учитывать следующие особенности большинства европейских языков:

* + если шифробозначение часто встречается, равномерно располагается по шифртексту, в отдельных местах чередуется через 1, 2 или 3 знака, сочетается со средними и редкими (по частоте) шифробозначениями, то это дает основания полагать, что искомое шифробозначение скрывает гласную букву;
  + удвоение согласных в открытом тексте происходит реже;
  + если некоторое шифробозначение признано гласной, то буква, часто сочетающаяся с ней, скорее всего согласная;
  + в открытом тексте чрезвычайно редко встречаются три и более подряд идущие гласные;
  + четыре и более подряд идущие согласные также очень редки.

Важно учитывать также процентное соотношение чисел гласных и согласных в открытом тексте.

При проверке гипотез о значениях шифробозначений полезен поиск в шифртексте слов с характерной структурой, которые часто встречаются в открытом тексте. Для русского языка это, например, слова *сколько, которое, что* и т.п. Для английского языка – слова *every*, *that, look, the* и т.п. Такие слова выделяются в шифртексте посредством интервалов между повторяющимися частыми буквами, характерными сочетаниями гласных и согласных.

Если с помощью приведенных соображений произведено несколько идентификаций шифробозначений, то дальнейшая работа по вскрытию текста криптограммы не представляет особого труда.

Задача дешифрования еще более упрощается, если известно, что использовался сдвиговый или аффинный шифр. Так, для аффинного шифра бывает достаточно идентифицировать лишь пару шифробозначений с тем, чтобы полностью восстановить открытый текст.

Таким образом, становится понятным, что наиболее трудно формализуемым фрагментом алгоритма является проверка выдвигаемых гипотез о значениях шифробозначений. Трудность состоит в формулировке критерия, подтверждающего или отвергающего ту или иную гипотезу.

#### Вопросы и задания для самостоятельной работы

1. Какие шифры называются шифрами простой замены?
2. Приведите пример шифра перестановки, который может рассматриваться и как блочный шифр замены?
3. Что является ключом шифра простой замены? Каково максимально возможное число ключей шифра простой замены?
4. Что более целесообразно для надежной защиты информации: архивация открытого текста с последующим шифрованием или шифрование открытого текста с последующей архивацией?
5. Имеет ли шифр Плэйфера эквивалентные ключи, то есть такие ключи, на которых любые открытые тексты шифруются одинаково?
6. Какие шифры называются омофонами? Приведите пример.
7. К каким шифрам относится система шифрования Петра I «Цифирь»?
8. Какие шифры называются шифрами многозначной замены? Приведите примеры.
9. Является ли шифр пропорциональной замены омофоном?

## ТЕМА 7. ШИФРЫ ГАММИРОВАНИЯ

В основе *шифров гаммирования* лежит следующий алгоритм – символы исходного текста складываются с символами некой случайной последовательности – *гаммы* (по названию  – буквы греческого алфавита, которой в математических записях обозначаются случайные величины)*.*

При работе алгоритмов *поточного* шифрования имитируется концепция одноразового блокнота, при этом используется короткий ключ для генерации шифрпоследовательности, которая «похожа на случайную последовательность». Такую шифрующую последовательность часто именуют псевдослучайной, и изучение того, каким образом генерировать псевдослучайную последовательность, составляет основную часть работы в области поточных шифров. Принято говорить, что шифрующая последовательность порождается генератором гаммы, или генератором псевдослучайной последовательности.

Последовательность считают случайной, если по поводу нее нельзя сделать никаких предсказаний и для нее невозможно найти никаких простых описаний. Но если шифрующая гамма может в принципе генерироваться эффективно, то такое простое описание наверняка должно существовать. Тем не менее, генератор способен порождать «неотличимые от случайной последовательности», если к гамме не могут быть применены никакие вычисления, позволяющие обнаружить это простое описание. Исходный ключ должен быть преобразован таким сложным образом, чтобы вычислительно было невозможно его вос-становить. Уровень случайности последовательности можно определять в терминах

классов вычислений (тестов), которые не выявляют в ней статистических нерегулярностей.

Программы-генераторы случайных чисел на самом деле выдают детерминированные числовые ряды, которые только кажутся случайными по своим свойствам. От них требуется, чтобы, даже зная закон формирования, но не зная ключа в виде начальных условий, никто не смог бы отличить числовой ряд от случайного.

### 7.1. Основные требования к гамме

Можно сформулировать *3 основных требования* к криптографически стойкому генератору псевдослучайной последовательности или гаммы.

1. Период гаммы должен быть достаточно большим для шифрования сообщений различной длины.
2. Гамма должна быть трудно предсказуемой. Это значит, что если известны тип генератора и кусок гаммы, то невозможно предсказать следующий за этим куском бит гаммы с вероятностью выше заданной.
3. Генерирование гаммы не должно быть связано с большими техническими и организационными трудностями.

При однократном использовании случайной гаммы одинакового размера с зашифровываемыми данными взлом шифра невозможен (так называемые криптосистемы с одноразовым или бесконечным ключом). В данном случае «бесконечный» означает, что гамма не повторяется.

В некоторых поточных шифрах ключ короче сообщения. Так, в системе Вернама для телеграфа используется бумажное кольцо, содержащее гамму. Конечно, стойкость такого шифра не идеальна.

Обмен ключами размером с шифруемую информацию не всегда уместен. Поэтому чаще используют гамму, получаемую с помощью генератора псевдослучайных чисел (ПСЧ). В этом случае ключ – порождающее число (начальное значение, вектор инициализации, initializing value, IV) для запуска генератора ПСЧ. Каждый генератор ПСЧ имеет период, после которого генерируемая последовательность повторяется. Очевидно, что период псевдослучайной гаммы должен превышать длину шифруемой информации.

Генератор ПСЧ считается корректным, если наблюдение фрагментов его выхода не позволяет восстановить пропущенные части или всю последовательность при известном алгоритме, но неизвестном начальном значении.

При использовании генератора ПСЧ возможны несколько вариантов.

1. Побитовое шифрование потока данных. Цифровой ключ используется в качестве начального значения генератора ПСЧ, а выходной поток битов суммируется по модулю 2 с исходной информацией. В таких системах отсутствует свойство распространения ошибок.
2. Побитовое шифрование потока данных с обратной связью (ОС) по шифртексту. Такая система аналогична предыдущей, за исключением того, что шифртекст возвращается в качестве параметра в генератор ПСЧ. Характерно свойство распространения ошибок. Область распространения ошибки зависит от структуры генератора ПСЧ.
3. Побитовое шифрование потока данных с ОС по исходному тексту. Базой генератора ПСЧ является исходная информация. Характерно свойство неограниченного распространения ошибки.
4. Побитовое шифрование потока данных с ОС по шифртексту и по исходному тексту.

В любом случае, когда генерируемая последовательность основана только на состоянии ЭВМ, она называется псевдослучайной. Действительно, случайными являются только некоторые физические процессы.

При зашифровании открытого текста *равновероятной* гаммой получается шифртекст, вероятностные свойства которого не отличаются от самой равновероятностной гаммы. Это обстоятельство не дает возможности криптоаналитику использовать диаграмму повторяемости букв открытого текста, т.к. при наложении гаммы эта информация «стирается». Поэтому вероятностные свойства гаммы должны быть близки к случайной равновероятностной последовательности.

Использование при шифровании неравновероятной гаммы позволяет криптоаналитику выделить по шифртексту несколько знаков гаммы, имеющих достаточную суммарную вероятность, и получить возможность восстановить открытый текст. Такая ситуация может произойти, например, в результате неисправности гаммообразующего устройства, т.е. устройства, вырабатывающего гамму, при этом в гамме могут встречаться не все знаки.

Большой ошибкой при использовании шифров гаммирования является повторное использование гаммы, также дающее криптоаналитику возможность восстановления открытых текстов.

Таким образом, можно сформулировать основные требования к гамме.

1. Для каждого сообщения использовать новую шифрующую гамму (повторное использование гаммы недопустимо).
2. Для формирования гаммы использовать аппаратные генераторы случайных чисел на основе физических процессов.
3. Длина шифрующей гаммы должна быть не менее длины защищаемого сообщения.

### 7.2. Шифр Виженера

Еще одно важное усовершенствование многоалфавитных систем, состоящее в идее использования в качестве ключа текста самого сообщения или же шифрованного текста, принадлежит Блезу де Виженеру. Такой шифр был назван *самоключом*. В книге Виженера «Трактат о шифрах» самоключ представлен следующим образом.

В простейшем случае за основу бралась таблица Тритемия с добавленными к ней в качестве первой строки и первого столбца алфавитами в их естественном порядке (рис. 7.1). Позже такая таблица стала называться *таблицей Виженера*.

В общем случае таблица Виженера состоит из циклически сдвигаемых алфавитов, причем первая строка может быть произвольным смешанным алфавитом. Первая строка служит алфавитом открытого текста, а первый столбец – алфавитом ключа. Для зашифрования открытого сообщения (To = t1t2...) Виженер предлагал в качестве ключевой последовательности (Г) использовать само сообщение (To) с добавленной к нему в качестве первой буквы (to) известной отправителю и получателю. Последовательности букв подписывались друг под другом:

Г = tot1t2...ti-1...

To = t1t2t3...ti...

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tш = s1s2s3...si...

|  |  |  |
| --- | --- | --- |
|  | A B C D E F G H I K L M N O P Q R S T U X Y Z W | |
| A  B  C  D  E  F  G  H  I  K  L  M  N  O  P  Q  R  S  T  U  X  Y  Z  W |  |  |
|  | 1. B C D E F G H I K L M N O P Q R S T U X Y Z W 2. C D E F G H I K L M N O P Q R S T U X Y Z W A 3. D E F G H I K L M N O P Q R S T U X Y Z W A B 4. E F G H I K L M N O P Q R S T U X Y Z W A BC 5. F G H I K L M N O P Q R S T U X Y Z W A B C D 6. G H I K L M N O P Q R S T U X Y Z W A B C D E 7. H I K L M N O P Q R S T U X Y Z W A B C D E F 8. I K L M N O P Q R S T U X Y Z W A B C D E F G 9. K L M N O P Q R S T U X Y Z W A B C D E F G H 10. L M N O P Q R S T U X Y Z W A B C D E F G H I 11. M N O P Q R S T U X Y Z W A B C D E F G H I K 12. N O P Q R S T U X Y Z W A B C D E F G H I K L 13. O P Q R S T U X Y Z W A B C D E F G H I K L M 14. P Q R S T U X Y Z W A B C D E F G H I K L M N 15. Q R S T U X Y Z W A B C D E F G H I K L M N O 16. R S T U X Y Z W A B C D E F G H I K L M N O P 17. S T U X Y Z W A B C D E F G H I K L M N O P Q 18. T U X Y Z W A B C D E F G H I K L M N O P Q R 19. U X Y Z W A B C D E F G H I K L M N O P Q R S 20. X Y Z W A B C D E F G H I K L M N O P Q R S T 21. Y Z W A B C D E F G H I K L M N O P Q R S T U 22. Z W A B C D E F G H I K L M N O P Q R S T U X 23. W A B C D E F G H I K L M N O P Q R S T U X Y   W A B C D E F G H I K L M N O P Q R S T U X Y Z |

***Рис. 7.1.*** *Таблица Виженера*

При этом пара букв, стоящих друг под другом в Г и Тo, указывала, соответственно, номера строк и столбцов таблицы, на пресечении которых находится знак si шифрованного текста (Tш). Например, фраза HUNC CAVETO VIRUM..., и начальная буква Р дают шифртекст YCHP ECUWZH IDAMG. Во втором варианте Виженер предлагал в качестве ключевой последовательности использовать шифрованный текст:

Г = sos1s2...si-1...

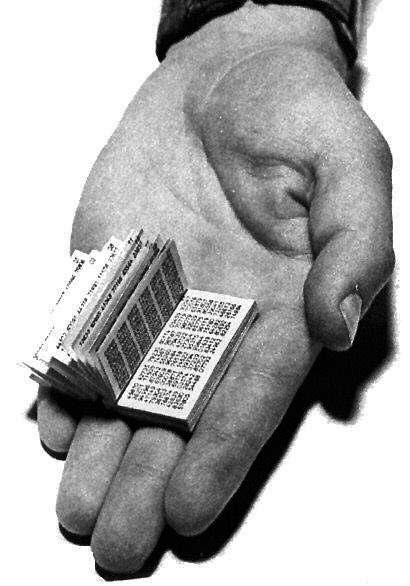
To = t1t2t3...ti...

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tш = s1s2s3...si...

Самоключ Виженера был незаслуженно забыт на долгое время, а под шифром Виженера часто понимают самый простой вариант с коротким ключевым словом и с таблицей, состоящей из обычных алфавитов.

### 7.3. Одноразовый блокнот К. Шеннона

Популярность поточных шифров можно связывать с работой Клода Шеннона, посвященной анализу *одноразовых гамма-блокнотов*. Название «одноразовый блокнот» стало общепринятым в годы Второй мировой войны, когда для шифрования широко использовались бумажные блокноты.

Одноразовый блокнот использует длинную шифрующую последовательность, которая состоит из случайно выбираемых бит или наборов бит (символов). Шифрующая последовательность побитно или посимвольно накладывается на открытый текст, имеет ту же самую длину, что и открытое сообщение, и может использоваться только один раз (о чем свидетельствует само название шифрсистемы); ясно, что при таком способе шифрования требуется огромное количество шифрующей гаммы.

Открытый текст сообщения m записывают как последовательность бит или символов m = m0m1...mn-1, а двоичную или символьную шифрующую последовательность k той же самой длины – как k = k0k1...kn-1.

Шифртекст c = c0c1...cn-1 определяется соотношением ci = mi ki при 0≤i≤n1, где  обозначает операцию «исключающее ИЛИ» (ассемблерная операция XOR) по модулю два или по любому другому модулю в случае символьной гаммы.

В своей исторической работе «Communication theory of secrecy systems» («Теория связи в секретных системах», 1949 г.) Шеннон доказал то, что до этого многие принимали на веру: одноразовый гамма-блокнот является «невскрываемой» шифрсистемой. Фактически Шеннон представил эту криптосистему как совершенную: даже противник, обладающий безграничным количеством вычислительной мощности, не в силах придумать ничего лучшего, чем предполагать значения бит или символов сообщения, поскольку шифртекст статистически не зависит от открытого текста.

### 7.4. Аддитивные методы шифрования

Далее рассмотрим аддитивные методы шифрования. Сущность этих методов заключается в последовательном суммировании цифровых кодов, соответствующих символам исходной информации, с некоторой специальной последовательностью кодов, называемой гаммой. Такое шифрование является по сути гаммированием, так как представляет собой наложение гаммы на исходный текст.

Процедуру наложения гаммы на исходный текст можно осуществить двумя способами. При первом способе символы исходного текста и символы гаммы заменяются цифровыми эквивалентами, которые затем складываются по модулю R, где R – число символов в алфавите. При втором способе символы исходного текста и гаммы представляются в виде двоичных кодов, а затем соответствующие разряды складываются по модулю 2.

Для аддитивных методов ключом шифрования является гамма. Основными свойствами гаммы, определяющими криптостойкость, являются длительность периода гаммы (длина ключа) и равномерность статистических характеристик. Последнее свойство обеспечивает отсутствие закономерностей в появлении различных символов гаммы в пределах периода.

Если ключ короче, чем шифруемая последовательность символов, то такой шифртекст может быть расшифрован криптоаналитиком статистическими методами исследования.

Если ключ представляет собой непериодическую последовательность случайных чисел, длина которой превышает длину шифруемой информации, то без знания ключа расшифровать такой шифртекст практически невозможно.

На практике наиболее эффективным стало использование для получения гаммы генераторов (датчиков) псевдослучайных чисел. Такие генераторы используют исходную информацию относительно малой длины для получения практически бесконечной последовательности псевдослучайных чисел.

Для этой цели могут использоваться конгруэнтные генераторы, вырабатывающие псевдослучайные последовательности чисел, для которых строго математически определяются такие основные характеристики, как периодичность и случайность выходных последовательностей.

Среди конгруэнтных генераторов выделяется своей простотой и эффективностью линейный генератор, вырабатывающий псевдослучайную последовательность чисел Т(i) в соответствии с соотношением:

Т(i+1) = (aT(i) + с) mod m,

где а и с – константы, Т(0) – исходная величина, выбранная в качестве по-

рождающего числа.

Период генерируемых чисел зависит от величины а и с и является максимальным тогда и только тогда, когда с – нечетное число и а(mod 4) = 1. Значение m обычно принимается равным 2S, где S – длина машинного слова в битах. Такой генератор может быть сравнительно легко создан как аппаратными средствами, так и программно.

### 7.5. Режим гаммирования ГОСТ 28147-89

Схема реализации режима гаммирования приведена на рисунке 7.2.

Открытые данные, разбитые на 64-разрядные блоки T0(i), зашифровываются в режиме гаммирования путем поразрядного суммирования по модулю 2 в сумматоре СМ5 с гаммой шифра Гш(i), которая вырабатывается блоками по 64 бита. Число двоичных разрядов в блоке Т0(M), где М определяется объемом шифруемых данных, может быть меньше 64, при этом неиспользованная для зашифрования часть гаммы шифра из блока Гш(M) отбрасывается.

В ключевое запоминающее устройство (КЗУ) вводятся 256 бит ключа. В накопители N1, N2 вводится 64-разрядная двоичная последовательность (*синхропосылка*) S=(S1,S2,...,S64), являющаяся исходным заполнением этих накопителей для последующей выработки М блоков гаммы шифра. Синхропосылка вводится в N1 и N2 так, что значение S1 вводится в 1-й разряд N1, значение S33 – в 1-й разряд N2, S64 – в 32-й разряд N2.

Исходное заполнение накопителей N1 и N2 (синхропосылка S) зашифровывается в режиме простой замены (нижняя часть рисунка). Результат зашифрования A(S)=(Y0,Z0) переписывается в 32-разрядные накопители N3 и N4 так, что заполнение N1 переписывается в N3, а N2 – в N4.

Заполнение накопителя N4 суммируется по модулю (232–1) в сумматоре

СМ4 с 32-разрядной константой С1 из накопителя N6, результат записывается в N4.

Заполнение накопителя N3 суммируется по модулю 232 в сумматоре СМ3 с 32-разрядной константой С2 из накопителя N5, результат записывается в N3.

Заполнение N3 переписывается в N1, а заполнение N4 – в N2. При этом заполнение N3, N4 сохраняется.

Заполнение N1 и N2 зашифровывается в режиме простой замены. Полученное в результате в N1, N2 зашифрование образует первый 64-разрядный блок гаммы шифра Гш(1), который суммируется в СМ5 с первым 64-разрядным блоком открытых данных Т0(1).

В результате получается 64-разрядный блок зашифрования данных Гш(1).

Для получения следующего 64-разрядного блока гаммы шифра Гш(2) заполнение N4 суммируется по модулю (232–1) в СМ4. С константой С1 из N6, заполнение N3 суммируется по модулю 232 в сумматоре СМ3 с С2 (в N5). Новое заполнение N3 переписывается в N1, а новое заполнение N4 переписывается в N2, при этом заполнение N3, N4 сохраняется.

Заполнение N1 и N2 зашифровывается в режиме простой замены. Полученное в результате зашифрования заполнение N1, N2 образует второй 64разрядный блок гаммы шифра Гш(2), который поразрядно суммируется по модулю 2 в СМ5 со вторым блоком открытых данных Т0(2).

Аналогично вырабатываются блоки гаммы шифра Гш(3), ..., Гш(М) и зашифровываются блоки открытых данных Т0(3), ..., Т0(М).

В канал связи (или память ЭВМ) передается синхропосылка S и блоки зашифрованных данных Тш(1), Тш(2), ..., Тш(М).

Уравнение зашифрования имеет вид:

Tш(i) = A(Yi-1 +C 2, Zi-1 ’C +1) T0(i) = Гш(i) T0(i),

Где + – суммирование по модулю 232,



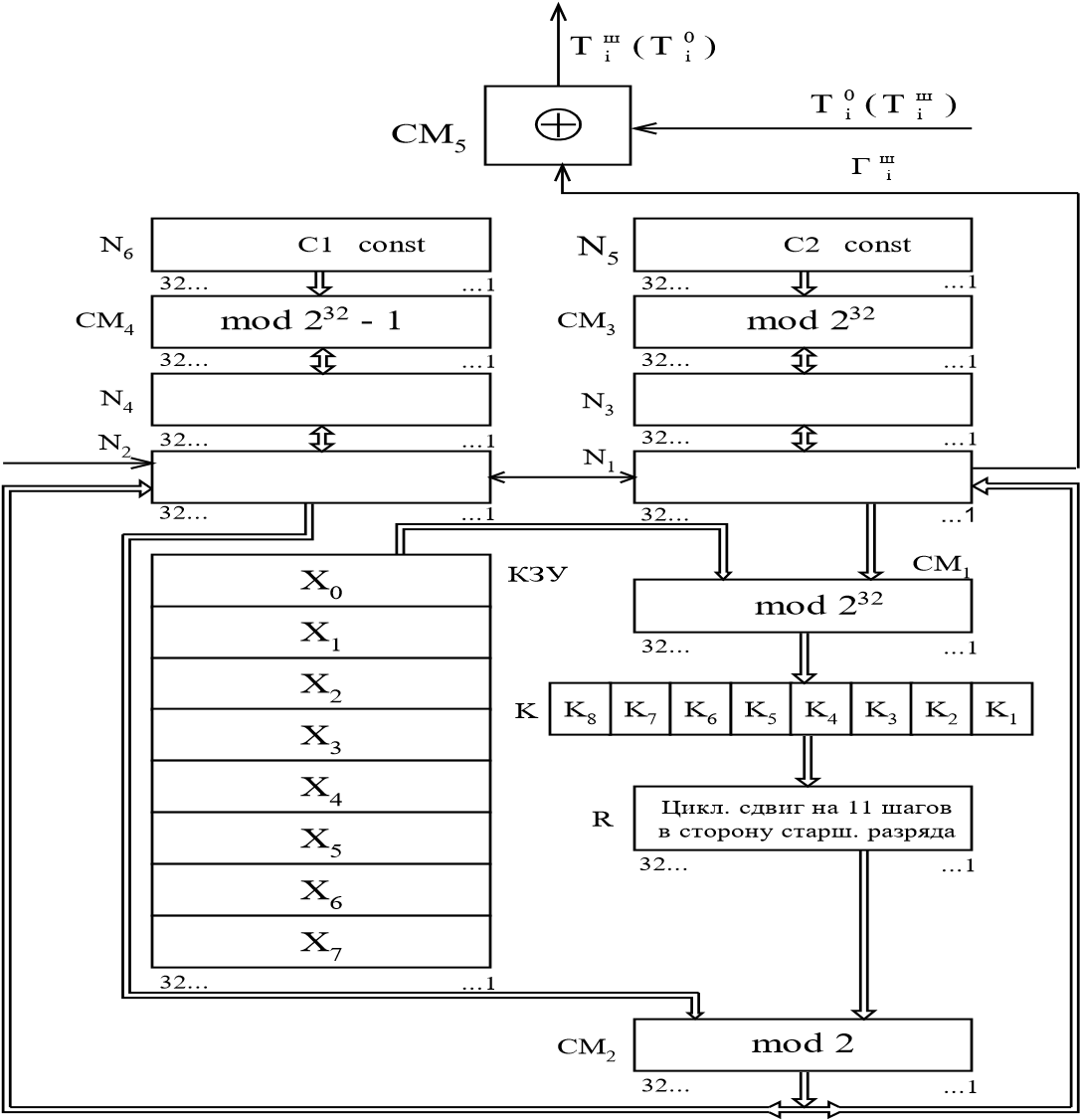
+’ – суммирование по модулю 232–1,

 – суммирование по модулю 2,

Yi – содержимое накопителя N3 после зашифрования i-го блока открытых данных T0(i),

Zi – содержимое накопителя N4 после зашифрования i-го блока открытых данных T0(i),

(Y0, Z0) = A(S).



***Рис. 7.2.*** *Режим гаммирования ГОСТ 28147-89*

#### Вопросы и задания для самостоятельной работы

1. Поясните, что вы понимаете под совершенным шифром. Приведите примеры.
2. Почему шифр Тритемия, лежащий в основе шифра Виженера, не является шифром гаммирования?
3. Почему наложение на открытый текст гаммы, представляющей собой периодическую последовательность небольшого периода, не дает надежной защиты?
4. Почему недопустимо использовать дважды одну и ту же гамму (даже случайную и равновероятностную!) для зашифрования разных открытых текстов?
5. Почему в качестве гаммы нецелесообразно использовать текст художественного произведения? Можете ли вы предложить метод вскрытия такого шифра?
6. Перечислите основные требования к гамме.

## ТЕМА 8. ПОТОЧНЫЕ СИСТЕМЫ ШИФРОВАНИЯ

Все недостатки непосредственного применения криптоалгоритмов устраняются в криптографических системах (криптосистемах). *Криптосистема* – это завершенная комплексная модель, способная производить двусторонние криптопреобразования над данными произвольного объема и подтверждать время отправки сообщения, обладающая механизмом преобразования паролей и ключей и системой транспортного кодирования. Таким образом, криптосистема выполняет три основные функции:

1. Усиление защищенности данных.
2. Облегчение работы с криптоалгоритмом со стороны человека.
3. Обеспечение совместимости потока данных с другим программным обеспечением.

Современные поточные криптосистемы ориентированы на программные, программно-аппаратные и аппаратные методы реализации. Поточные криптосистемы преобразуют информацию в процессе потока данных.

Поточные шифры, шифрующие и расшифровывающие данные по одному биту, хорошо подходят для аппаратной реализации.

### 8.1. Регистры сдвига с обратной связью

Большинство реальных поточных шифров основано на *регистрах сдвига с обратной связью*. Регистр сдвига применяют для генерации ключевой последовательности. Регистр сдвига с обратной связью состоит из двух частей: *регистра сдвига* и *функции обратной связи*.

Регистр сдвига представляет собой последовательность битов. (Количество битов определяется длиной сдвигового регистра. Если длина равна n битам, то регистр называется *n*-битовым регистром сдвига.) Всякий раз, когда нужно извлечь бит, все биты регистра сдвига сдвигаются вправо на 1 позицию. Новый крайний левый бит является функцией всех остальных битов регистра.

bn = F(b1,b2, …,bn).

На выходе регистра оказывается один, обычно младший значащий бит. Периодом регистра называется длина получаемой последовательности до начала ее повторения.



***Рис. 8.1.*** *Регистр сдвига с линейной обратной связью*

Простейшим видом регистра сдвига с обратной связью является *регистр сдвига с линейной обратной связью (РСЛОС)*. Обратная связь представляет собой XOR некоторых битов регистра; эти биты называются *отводной последовательностью*.

РСЛОС (*n*-битовый) может находиться в одном из 2*n* внутренних состояний. Это означает, что теоретически такой регистр может генерировать псевдослучайную последовательность с периодом 2n−1 битов (период равен 2*n*−1, потому что заполнение РСЛОС нулями приведет к тому, что сдвиговый регистр будет выдавать бесконечную последовательность нулей). Только при определенных последовательностях РСЛОС циклически пройдет через все 2*n*−1 внутренних состояний. Такие РЛСОС имеют максимальный период.

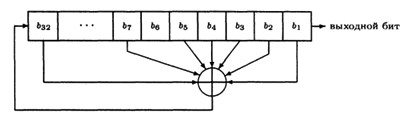
Примером таких последовательностей является конгруэнтный генератор.

Для того чтобы конкретный РСЛОС имел максимальный период, многочлен, ассоциированный с получаемой псевдослучайной последовательностью битов, должен быть примитивным по модулю 2, то есть не раскладываться на произведение двоичных многочленов меньшей степени.

Например, многочлен *x*32 + *x*7 + *x*5 + *x*3 + *x*2 + *x* + 1 примитивен по модулю 2. Рассмотрим этот многочлен.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 32 | … | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |

Степень многочлена задает длину РСЛОС. Свободный член многочлена всегда равен 1, и его можно опустить. Степени формальной переменной многочлена, за исключением 0-й, задают последовательность, отсчитываемую от левого края сдвигового регистра. То есть члены многочлена с меньшей степенью соответствуют позициям, расположенным ближе к правому краю регистра. Тогда для взятого 32-битового сдвигового регистра (рис. 8.2) новый бит генерируется с помощью операции «сложение по модулю 2» тридцать второго, седьмого, пятого, третьего, второго и первого битов. Данный РСЛОС будет иметь максимальную длину, циклически проходя до повторения через 232−1 различных значений.



***Рис. 8.2.*** *32-битовый сдвиговой регистр*

Сами по себе РСЛОС являются генераторами псевдослучайных последовательностей с определенными свойствами, но они обладают некоторыми нежелательными неслучайными свойствами. Для РСЛОС длины n внутреннее состояние представляет собой предыдущие n выходных битов генератора. Даже если схема обратной связи хранится в секрете, она может быть определена по 2*n* выходным битам генератора с помощью алгоритма Берлекэмпа-Мэсси (алгоритм построения многочлена над конечным полем GF(2), генерирующего кольцо с заданным, или не меньше заданного, количеством элементов).

Кроме того, большие случайные числа, генерируемые с использованием идущих подряд бит этой последовательности, сильно коррелированны и для некоторых типов приложений вовсе не являются случайными. Несмотря на это, РСЛОС часто используются при разработке алгоритмов шифрования.

### 8.2. Скремблеры

Представителями поточных шифров являются скремблеры (англ. scramble – перемешивать). Скремблерами называются программные или аппаратные реализации алгоритма, позволяющего шифровать побитно непрерывные потоки информации. Сам скремблер представляет собой набор бит, изменяющихся на каждом шаге по определенному алгоритму. После выполнения каждого очередного шага на его выходе появляется шифрующий бит – либо 0, либо 1, который накладывается на текущий бит информационного потока операцией XOR.

В последнее время сфера применения скремблирующих алгоритмов значительно сократилась. Это объясняется в первую очередь снижением объемов побитной последовательной передачи информации, для защиты которой были разработаны данные алгоритмы. Практически повсеместно в современных системах применяются сети с коммутацией пакетов, для поддержания конфиденциальности которой используются блочные шифры. А их криптостойкость превосходит, и порой довольно значительно, криптостойкость скремблеров.

Суть скремблирования заключается в побитном изменении проходящего через систему потока данных. Практически единственной операцией, используемой в скремблерах, является XOR – «побитное исключающее ИЛИ». Параллельно прохождению информационного потока в скремблере по определенному правилу генерируется поток бит – кодирующий поток. Как прямое, так и обратное шифрование осуществляется наложением по XOR кодирующей последовательности на исходную.

Генерация кодирующей последовательности бит производится циклически из небольшого начального объема информации – ключа по следующему алгоритму. Из текущего набора бит выбираются значения определенных разрядов и складываются по XOR между собой. Все разряды сдвигаются на 1 бит вправо, а только что полученное значение («0» или «1») помещается в освободившийся самый старший разряд. Значение, находившееся в самом младшем разряде до сдвига, добавляется в кодирующую последовательность, становясь очередным ее битом (см. рис. 8.3).

+

+

+

ИП

ЗП

***Рис. 8.3.*** *Схема скремблирования с комбинацией «1012»*

Скремблер записывается комбинацией «1012» – единицы соответствуют разрядам, с которых снимаются биты для формирования обратной связи.

Кодирующая последовательность зависит от исходного ключа и скремблера.

Пример шифрования информационной последовательности 01011110112 со скремблером 1012 с начальным ключом 1102:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Номер сдвига | Состояние  регистра | Бит код. посл. | Бит инф. посл. | Бит зашифр.  посл. | Следующий старший бит регистра |
| 1 | 110 | 0 | 0 | 0  0 = 0 | 1  0 = 1 |
| 2 | 111 | 1 | 1 | 1  1 = 0 | 1  1 = 0 |
| 3 | 011 | 1 | 0 | 1  0 = 1 | 0  1 = 1 |
| 4 | 101 | 1 | 1 | 1  1 = 0 | 1  1 = 0 |
| 5 | 010 | 0 | 1 | 0  1 = 1 | 0  0 = 0 |
| 6 | 001 | 1 | 1 | 1  1 = 0 | 0  1 = 1 |
| 7 | 100 | 0 | 1 | 0  1 = 1 | 1  0 = 1 |
| 8 | 110 | 0 | 0 | 0  0 = 0 | 1  0 = 1 |
| 9 | 111 | 1 | 1 | 1  1 = 0 | 1  1 = 0 |
| 10 | 011 | 1 | 1 | 1  1 = 0 | 0  1 = 1 |

Таким образом, устройство скремблера очень простое. Его реализация возможна как на электронной, так и на электрической базе, что и обеспечило его широкое применение в полевых условиях. Бит выходной последовательности зависит только от одного входного бита. Данный факт связан с неизбежно возникающими в канале передачи помехами, которые могут исказить биты, не связанную с ними группу байт, как это имеет место в блочных шифрах.

Расшифрование скремблированных последовательностей происходит по той же самой схеме, что и шифрование. Именно для этого в алгоритмах применяется результирующее шифрование по «исключающему ИЛИ» – схема, однозначно восстановимая при расшифровании без каких-либо дополнительных вычислительных затрат.

Главная проблема шифров на основе скремблеров – синхронизация передающего и принимающего устройств. При пропуске или ошибочном вставлении хотя бы одного бита вся передаваемая информация необратимо теряется. Поэтому в системах шифрования на основе скремблеров очень большое внимание уделяется методам синхронизации. На практике для этих целей обычно применяется комбинация двух методов:

а) добавление в поток информации синхронизирующих битов, заранее

известных приемной стороне, что позволяет ей при нахождении такого бита активно начать поиск синхронизации с отправителем;

б) использование высокоточных генераторов временных импульсов, что

позволяет в моменты потери синхронизации производить расшифрование принимаемых битов информации «по памяти» без синхронизации.

Число бит, охваченных обратной связью, то есть разрядность устройства памяти для порождающих кодирующую последовательность бит, называется разрядностью скремблера. Изображенный выше скремблер имеет разрядность 5. В отношении параметров криптостойкости данная величина полностью идентична длине ключа блочных шифров, который будет проанализирован далее. На данном же этапе важно отметить, что чем больше разрядность скремблера, тем выше криптостойкость системы, основанной на его использовании.

При достаточно долгой работе скремблера неизбежно возникает его зацикливание. По выполнении определенного числа тактов в ячейках скремблера создается комбинация бит, которая в нем уже однажды оказывалась, и с этого момента кодирующая последовательность начнет циклически повторяться с фиксированным периодом. В приведенном выше примере на восьмом сдвиге состояние регистра повторяется 110.

Данная проблема неустранима по своей природе, так как в N разрядах скремблера не может пребывать более 2N комбинаций бит, и, следовательно, максимум через 2N–1 циклов повтор комбинации обязательно произойдет. Комбинация «все нули» сразу же исключается из цепочки графа состояний скремблера – она приводит скремблер к такому же положению «все нули». Это указывает еще и на то, что ключ «все нули» неприменим для скремблера. Каждый генерируемый при сдвиге бит зависит только от нескольких бит хранимой в данный момент скремблером комбинации. Поэтому после повторения некоторой ситуации, однажды уже встречавшейся в скремблере, все следующие за ней будут в точности повторять цепочку, уже прошедшую ранее в скремблере.

Возможны различные типы графов состояния скремблера. Состояние скремблера определяется двоичным кодом, а переход – сдвигом от исходного кода к полученному коду с использованием функции скремблера.

Количество состояний 3-разрядного скремблера равно 8. На рисунке 8.4 приведены все варианты для 3-разрядного скремблера.

В случае *в* и *д* (соответствует скремблеру 011 и 101) все возможные состояния, кроме нулевого, объединены в один цикл. Очевидно, что именно в этом случае, когда все 2N–1 состояний системы образуют цикл, период повторения выходных комбинаций равен 8 (максимален), а корреляция между длиной цикла и начальным состоянием скремблера (ключом), которая привела бы к появлению более слабых ключей, отсутствует.

000

001

010

100

101

011

110

111

***Рис. 8.4 а.*** *Граф переходов скремблера 001*

000

0011

010

100

101

011

110

111

***Рис. 8.4 б.*** *Граф переходов скремблера 010*

000

001

010

100

101

011

110

111

***Рис. 8.4 в.*** *Граф переходов скремблера 011*

000

001

010

100

101

011

110

111

***Рис. 8.4 г.*** *Граф переходов скремблера 100*

000

001

010

100

101

011

110

111

***Рис. 8.4 д.*** *Граф переходов скремблера 101*

000

001

010

100

101

011

110

111

***Рис. 8.4 е.*** *Граф переходов скремблера 110*

000

0011

010

100

101

011

110

111

***Рис. 8.4 ж.*** *Граф переходов скремблера 111*

Для скремблера любой разрядности N всегда существует такой выбор охватываемых обратной связью разрядов, что генерируемая ими последовательность бит будет иметь период, равный 2N–1 битам. Так, например, в 8битном скремблере, при охвате 0-го, 1-го, 6-го и 7-го разрядов (скремблер 11000011) действительно за время генерации 255 бит последовательно проходят все числа от 1 до 255, не повторяясь ни разу (максимальный период).

Схемы с выбранными по данному закону обратными связями называются генераторами последовательностей наибольшей длины (ПНД), и именно они используются в скремблирующей аппаратуре. Из множества генераторов ПНД заданной разрядности во времена, когда они реализовывались на электрической или минимальной электронной базе, выбирались те, у которых число разрядов, участвующих в создании очередного бита, было минимальным. Обычно генератора ПНД удавалось достичь за 3 или 4 связи. Сама же разрядность скремблеров превышала 30 бит, что давало возможность передавать до 240 бит = 100 Мбайт информации без опасения начала повторения кодирующей последовательности.

ПНД неразрывно связаны с математической теорией неприводимых полиномов. Оказывается, достаточно, чтобы полином степени N не был представим по модулю 2 в виде произведения никаких других полиномов, для того чтобы скремблер, построенный на его основе, создавал ПНД. Например, единственным неприводимым полиномом степени 3 является x3+x+1, в двоичном виде он записывается как 10112 (единицы соответствуют присутствующим разрядам). Скремблеры на основе неприводимых полиномов образуются отбрасыванием самого старшего разряда (он всегда присутствует, а следовательно, несет информацию только о степени полинома), так, на основе указанного полинома мы можем создать скремблер 0112 с периодом зацикливания 7(=23–1). Естественно, что на практике применяются полиномы значительно более высоких порядков. А таблицы неприводимых полиномов любых порядков можно всегда найти в специализированных математических справочниках.

Существенным недостатком скремблирующих алгоритмов является их нестойкость к фальсификации.

В настоящее время скремблирование применяется в аналоговой телефонии для шифрования речевого сигнала, при этом *скремблер*представляет собойустройство, осуществляющее преобразование сигнала путем изменений соотношений между временем, амплитудой и частотой, не выходящих за пределы используемого диапазона.

### 8.3. Методы рандомизации сообщений

Два основных метода внесения случайности (рандомизации) в процесс шифрования представляют из себя:

а) внесение случайных бит в шифруемый файл с игнорированием их на

дешифрующей стороне;

б) шифрование исходного файла случайным ключом.

Усовершенствованием, направленным на повышение стойкости всей системы в целом, является создание ключей сеанса. Эта операция необходима в тех случаях, когда производится частое шифрование сходных блоков данных одним и тем же ключом. Например, это имеет место при передаче информации или команд в автоматизированных системах управления, в банковских операциях и многих других случаях передачи информации, имеющей определенный заранее известный формат.

В этом случае необходимо введение какой-либо случайной величины в процесс шифрования. Это можно сделать несколькими способами:

1. Запись в начало файла данных псевдослучайной последовательности байт заранее оговоренной длины с отбрасыванием ее при расшифровании – этот метод будет работать только при применении алгоритмов создания цепочек с памятью.
2. Применение модифицированных алгоритмов создания цепочек, которые при шифровании каждого блока смешивают с ним:

а) фиксированную случайную величину, прикрепленную к началу за-

шифрованного файла;

б) значения (значение), вычисляемые с помощью того же шифра и ключа

от заранее оговоренной величины.

1. Создание специально для каждого файла совершенно случайного ключа, так называемого ключа сеанса, которым и шифруется весь файл (сам же ключ сеанса шифруется первоначальным ключом, называемым в этом случае мастер-ключом).

#### Генераторы случайных и псевдослучайных последовательностей

Самая большая проблема всех методов рандомизации сообщений – это порождение действительно случайной последовательности. Дело в том, что генераторы случайных последовательностей, используемые для общих целей, например, в языках программирования, являются на самом деле псевдослучайными генераторами. В принципе существует конечное, а не бесконечное множество состояний ЭВМ, и, как бы сложно ни формировалось в алгоритме число, оно все равно имеет относительно немного случайных бит информации.

Наиболее часто в прикладных задачах создания случайных и псевдослучайных чисел результат формируют из счетчика тиков – системных часов. В этом случае данные о текущем часе несут примерно 16 бит информации, значение счетчика тиков – еще 16 бит. Итого – 32 бита информации. На сегодняшний день границей стойкой криптографии является значение в 40 бит, при реальных длинах ключей в 128 бит.

К 32 битам можно добавить еще 16 бит из сверхбыстрого таймера, работающего на частоте 1,2 МГц в компьютерах архитектуры IBM PC, и этого еще недостаточно. Кроме того, даже если мы сможем набрать длину ключа в 128 бит, она будет нести псевдослучайный характер, поскольку основана на состоянии только лишь данной ЭВМ на момент начала шифрования. Источниками понастоящему случайных величин могут быть только внешние объекты, например человек или внешняя среда.

Два наиболее часто применяемых метода создания случайных последовательностей с помощью человека основаны на вводе с клавиатуры. В обоих случаях пользователя просят, не задумываясь, понабирать на клавиатуре бессмысленные сочетания букв.

По первому методу над самими введенными значениями производятся действия, повышающие случайность выходного потока. Так, например, удаляются несколько бит (до трех) введенного ASCII символа, часто удаляются еще один старший и младший биты. Затем объем полученной последовательности уменьшается еще в три раза наложением первого и второго бита на третий операцией XOR.

По второму методу на введенные символы алгоритм не обращает никакого внимания, зато запоминают интервалы времени, через которые произошли нажатия. Запись моментов производится по отсчетам быстрого системного таймера или внутреннему счетчику процессора, появившемуся в процессорах, начиная с Intel Pentium (частота соответствует частоте процессора). Так как верхние и младшие биты имеют определенную корреляцию между символами (первые из-за физических характеристик человека, вторые из-за особенностей операционной системы), то они отбрасываются (обычно удаляются 0–8 старших битов и 4–10 младших).

Как более редко встречающиеся варианты можно встретить:

1. комбинацию обоих клавиатурных методов;
2. метод, основанный на манипуляторе «мышь» – он выделяет случайную информацию из смещений пользователем указателя мыши.

В мощных криптосистемах военного применения используются действительно случайные генераторы чисел, основанные на физических процессах. Они представляют собой платы либо внешние устройства, подключаемые к ЭВМ через порт ввода-вывода. Два основных источника белого Гауссовского шума – высокоточное измерение тепловых флуктуаций и запись радиоэфира на частоте, свободной от радиовещания.

Таким образом, генераторы случайных последовательностей играют большую роль в современной криптографии. В том случае, когда генерируемая последовательность основана только на состоянии ЭВМ, она называется псевдослучайной. Действительно случайными являются только некоторые физические процессы и человеческий фактор.

### 8.4. Поточные шифрсистемы

#### Шифрсистема А5

А5 – это поточный шифр, используемый для шифрования GSM (Group Special Mobile), – европейского стандарта для цифровых сотовых мобильных телефонов.

Данная шифрсистема состоит из трех PCЛОС длиной 19, 22 и 23. Выходом является XOR трех PCЛОС. В А5 используется изменяемое управление тактированием. Каждый регистр тактируется в зависимости от своего среднего бита, затем выполняется XOR с обратной пороговой функцией средних битов всех трех регистров. Обычно на каждом этапе тактируется два РСЛОС.

Существует атака на открытом тексте, основанная на предположении о содержании первых двух РСЛОС и попытке определения третьего РСЛОС по ключевой последовательности. Тем не менее идеи, лежащие в основе А5, позволяют проектировать надежные поточные шифры.

Алгоритм эффективен и удовлетворяет всем известным статистическим тестам, единственная его слабость – короткие регистры. Варианты А5 с более длинными сдвиговыми регистрами и более плотными многочленами обратной связи позволяют противостоять силовой атаке.

#### Шифрсистема RC4

RC4 – это поточный шифр с переменным размером ключа, разработанный в 1987 г. Ривестом (R. Rivest) для RSA Data Security, Inc.

Алгоритм RC4 строится, как и любой потоковый шифр, на основе параметризованного ключом генератора псевдослучайных битов с равномерным распределением. Основные преимущества шифра – высокая скорость работы и переменный размер ключа. Типичная реализация выполняет 19 машинных команд на каждый байт текста.

Алгоритм работает в режиме OFB: поток ключей не зависит от открытого текста. Используется S-блок размером 8×8 : *S*0, *S*1, *S*2, …, *S*255. Размер блока является размером слова для алгоритма. Элементы слова представляют собой перестановку чисел от 0 до 255, а перестановка является функцией ключа переменной длины. В алгоритме применяются два счетчика, *i* и *j*, с нулевыми начальными значениями. Для генерации случайного байта выполняются следующие вычисления:

* *i* = (*i* + 1) mod 256;
* *j* = (*j* + *Si*) mod 256.  Поменять местами *Si* и *Sj*.  *t* = (*Si* + *Sj*) mod 256;  *K* = *St*.

*К* используется в операции XOR с открытым текстом для получения шифртекста или в операции XOR с шифртекстом для получения открытого текста. Шифрование выполняется примерно в 10 раз быстрее, чем в DES. Также несложна и инициализация S-блока. Сначала S-блок заполняется по правилу: *S*0 = 0, *S*1 = 1, …, S255 = 255. После этого ключ записывается в массив: *K*0, *K*1, …, *K*255. Затем при начальном значении *j* = 0 в цикле выполняются следующие вычисления:

* for *i* = 0 to 255 do *j* = (*j* + *Si* + *Ki*) mod 256
* Поменять местами *Si* и *Sj*.

Компания RSA Data Security, Inc. утверждает, что алгоритм устойчив к дифференциальному и линейному криптоанализу и что он в высокой степени нелинеен. S-блок медленно изменяется при использовании: *i* и *j* обеспечивают случайное изменение каждого элемента. RC4 входит в десятки коммерческих продуктов, включая Lotus Notes, AOCE компании Apple Computer и Oracle Secure SQL. Этот алгоритм также является частью спецификации стандарта Сотовой цифровой пакетной передачи данных CDPD (Cellular Digital Packet Data).

#### Шифрсистема SEAL

SEAL – это эффективный поточный шифр, разработанный в IBM Рогэвэем (P. Rogaway) и Копперсмитом (D. Coppersmith).

Алгоритм оптимизирован для 32-битовых процессоров. Для нормальной работы ему нужно восемь 32-битовых регистров и память на несколько килобайт. В SEAL предусмотрен ряд предварительных действий с ключом с сохранением результатов в нескольких таблицах. Таблицы используются для ускорения процедур шифрования и расшифрования. Особенностью SEAL является то, что он в действительности является не традиционным поточным шифром, а представляет собой семейство псевдослучайных функций. При 160-битовом ключе *k* и 32-битовом регистре *n*, SEAL растягивает *n* в *L*-битовую строку *k*(*n*). *L* может принимать любое значение, меньшее 64 Кбайт. SEAL использует следующее правило: если *k* выбирается случайным образом, то *k*(*n*) должно быть неотличимо от случайной *L*-битовой функции *n*.

Практический эффект того, что SEAL является семейством псевдослучайных функций, состоит в том, что он удобен в ряде приложений, где не применимы традиционные поточные шифры. При использовании большинства поточных шифров создается однонаправленная последовательность бит: единственным способом определить *i*-й бит (зная ключ и позицию *i*) является генерирование всех битов вплоть до *i*-го. Отличие семейства псевдослучайных функций состоит в том, что можно легко получить доступ к любой позиции клюючевой последовательности. Например, для шифрования жесткого диска, состоящего из множества 512-байтовых секторов, можно воспользоваться семейством псевдослучайных функций, подобных SEAL, и выполнить XOR каждого сектора с *k*(*n*). Это то же самое, как если бы была выполнена операция XOR всего диска с длинной псевдослучайной функцией, причем любая часть этой длинной последовательности бит может быть вычислена независимо. Семейство псевдослучайных функций также упрощает проблему синхронизации, встречающуюся в стандартных поточных шифрах, – можно зашифровать на ключе *k* *n*-е передаваемое сообщение *хn*, выполнив XOR *хn* и *k*(*n*). Получателю не нужно хранить состояние шифра для восстановления *хn*, ему не приходится беспокоиться и о потерянных сообщениях, влияющих на процесс расшифрования.

#### Вопросы и задания для самостоятельной работы

1. Каковы с точки зрения криптографии преимущества и недостатки перехода к поточному шифрованию?
2. Как реализуется предложенный К. Шенноном принцип «перемешивания» при практической реализации алгоритмов поточного шифрования?
3. Каковы основные недостатки алгоритма DES и каковы пути их устранения?
4. Для каких криптосистем можно рекомендовать использовать поточные шифры?
5. В чем заключаются достоинства и недостатки систем поточного шифрования по сравнению с блочными шифрами?
6. Почему возникает проблема синхронизации поточных шифров?
7. Какие причины обусловили широкое использование линейных регистров сдвига в качестве управляющих блоков поточных шифрсистем?

## ТЕМА 9. БЛОЧНЫЕ СИСТЕМЫ ШИФРОВАНИЯ

Современные блочные криптосистемы ориентированы на программные и программно-аппаратные методы реализации. Блочные криптосистемы представляют собой блочные (групповые) шифрпреобразования.

Блочные шифры легче реализуются программно, чем поточные, так как они позволяют избежать трудоемких манипуляций с битами и оперируют удобными для компьютера блоками данных.

### 9.1. Блочные системы шифрования

Блочные шифры шифруют целые блоки информации (от 4 до 32 байт) как единое целое – это значительно увеличивает стойкость преобразований к атаке полным перебором и позволяет использовать различные математические и алгоритмические преобразования.

Блочная криптосистема разбивает открытый текст М на последовательные блоки M1, M2,... и зашифровывает каждый блок с помощью одного и того же обратимого преобразования Ek, выполненного с помощью ключа *К*. Ek(М)=Ek(M1), Ek(M2). Любое из них можно рассматривать как последовательность операций, проводимых с элементами ключа и открытого текста, а также производными от них величинами.

Выбор элементов алгоритма шифрования достаточно велик, однако «элементарные» операции должны обладать хорошими криптографическими свойствами и допускать удобную техническую или программную реализацию. Обычно используются операции:

* побитового сложения по модулю 2 (обозначение операции ) двоичных векторов (XOR):

00=0

01=1

11=0

* сложение целых чисел по определенному модулю: например, по модулю 232, обозначение операции – +

a + b= a+b, если a+b<232, а + b= a+b-232, если a+b232,

где + – сложение целых чисел;

* умножение целых чисел по определенному модулю:

ab(mod n) = res(ab/n) – остаток от деления произведения целых чисел ab

на n;

* перестановка битов двоичных векторов;
* табличная замена элементов двоичных векторов.

Практическая стойкость алгоритмов шифрования зависит и от особенностей соединения операций в последовательности. Примерами блочных систем являются алгоритмы блочного шифрования, принятые в качестве стандартов шифрования данных в США и России – DES–алгоритм и ГОСТ-28147-89 соответственно.

Прямое криптографическое преобразование (шифрование) переводит блок открытого текста в блок шифртекста той же длины. Обратное криптографическое преобразование (расшифрование) переводит блок шифртекста в исходный блок открытого текста. Необходимое условие выполнения как прямого, так и обратного криптографического преобразования – наличие секретного ключа.

Для многих блочных шифров разрядность блока составляет 64 бита.

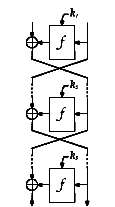
Прямое криптографическое преобразование обладает следующим свойством: различные блоки открытого текста отображаются в различные блоки шифртекста. При обратном преобразовании соответствие сохраняется. Прямое преобразование можно рассматривать как перестановку на множестве сообщений с фиксированным размером блока. Результат перестановки носит секретный характер, что обеспечивается секретным компонентом – ключом.

### 9.2. Конструкция Фейстеля

Конструкцией (сетью) Фейстеля называется метод обратимых преобразований текста, при котором значение, вычисленное от одной из частей текста, накладывается на другие части. Часто структура сети выполняется таким образом, что для шифрования и расшифрования используется один и тот же алгоритм – различие состоит только в порядке использования материала ключа.

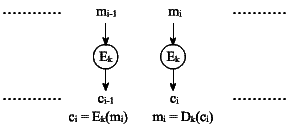
Сеть Фейстеля является модификацией метода смешивания текущей части шифруемого блока с результатом некоторой функции, вычисленной от другой независимой части того же блока (рис. 9.1). Эта методика получила широкое распространение, поскольку обеспечивает выполнение требования итерационного использования ключа и материала исходного блока информации.

Принцип итерирования является основным при разработке криптографических преобразований и заключается в многократной, состоящей из нескольких циклов обработке одного блока открытого текста.



***Рис. 9.1.*** *Схема конструкции Фейстеля*

На каждом цикле данные подвергаются специальному преобразованию при участии вспомогательного ключа, полученного из заданного секретного ключа. Выбор числа циклов определяется требованиями криптостойкости и эффективности реализации шифра. Как правило, чем больше циклов, тем выше криптостойкость и ниже эффективность реализации (больше задержка при шифровании/расшифровании), и наоборот. Так, например, в случае DES (федеральный криптостандарт США) для того, чтобы все биты шифртекста зависели от всех битов ключа и всех битов открытого текста, необходимо 5 циклов криптографического преобразования. DES с 16 циклами обладает высокой криптостойкостью по отношению к ряду криптоаналитических атак.



***Рис. 9.2.*** *Режим Электронной кодовой книги (ЕСВ)*

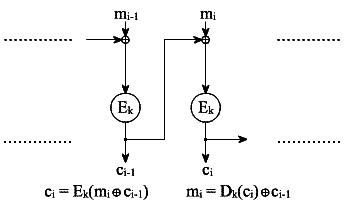
Конструкция Фейстеля (Н. Feistel), или сеть Фейстеля, представляет собой разновидность итерированного блочного шифра. При шифровании блок открытого текста разбивается на две равные части: правую и левую. Очевидно, что длина блока при этом должна быть четной. На каждом цикле одна из частей подвергается преобразованию при помощи функции f и вспомогательного ключа ki, полученного из исходного секретного ключа. Результат операции суммируется по модулю 2 (операция XOR) с другой частью. Затем левая и правая части меняются местами. Схема конструкции Фейстеля представлена на рисунке 9.1. Преобразования на каждом цикле идентичны, но на последнем не выполняется перестановка. Процедура расшифрования аналогична процедуре шифрования, однако ki выбираются в обратном порядке. Конструкция Фейстеля хороша тем, что прямое и обратное криптографические преобразования для такого блочного шифра имеют идентичную структуру.

Конструкция Фейстеля применяется в криптоалгоритмах DES, ГОСТ 28147-89, Lucifer, FEAL, Khufu, Khaire, JOKI, COST, CAST, Blowfish и др.

Блочный шифр, использующий такую конструкцию, является обратимым и гарантирует возможность восстановления входных данных функции на каждом цикле. Сама функция не обязательно должна быть обратимой. При задании произвольной функции не потребуется реализовывать две различные процедуры – одну для шифрования, а другую для расшифрования. Структура сети Фейстеля автоматически позаботится об этом.

### 9.3. Режимы шифрования блочных шифров

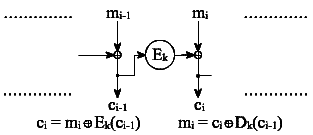
При использовании блочных шифров применяются различные режимы шифрования. Очевидно, что применение того или иного режима шифрования не должно отрицательно сказываться на эффективности и тем более криптостойкости блочного шифра. Режимы шифрования позволяют реализовать дополнительные, отсутствующие в исходной конструкции блочного шифра функции.



***Рис. 9.3.*** *Режим Сцепления блоков шифра (СВС)*

#### Шифрование в режимах ECB и CBC

Стандарт режимов шифрования для блочных шифров (применительно к криптоалгоритму DES) опубликован в материалах Национального института стандартов США. В более общем виде, применительно к произвольному блочному шифру с переменной длиной блока, стандарт опубликован в материалах и включает шифрование в следующих режимах: Электронной кодовой книги (Electronic Code Book, ЕСВ), Сцепления блоков шифра (Cipher Block Chaining, СВС), Обратной связи по шифртексту (Cipher Feedback, СРВ) и Обратной связи по выходу (Output Feedback, OFB). В режиме ЕСВ (рис. 9.2) шифрование/расшифрование j-ro блока открытого текста/шифртекста выполняется независимо: mi = Dk(ci), cj = EK(mi), где через Еk и Dk обозначены процедуры шифрования/расшифрования на секретном ключе k.



***Рис. 9.4.*** *Режим Обратной связи по шифртексту (СFВ)*

Криптостойкость режима ЕСВ не ниже, чем криптостойкость используемого блочного шифра. Недостаток заключается в том, что фиксированные блоки открытого текста (например, последовательность нулей длины l = nb бит, где b – длина блока) будут соответствовать фиксированным блокам шифртекста. Следовательно, открытый текст может быть легко изменен путем удаления, реплицирования и перестановки блоков шифртекста. Скорость обработки блоков в режиме ЕСВ фиксирована и определяется эффективностью реализации блочного шифра. Режим ЕСВ допускает эффективное распараллеливание вычислений. Однако конвейерная обработка блоков в данном режиме невозможна.

#### Режим CBC

В режиме СВС (рис. 9.3) каждый i-й блок открытого текста суммируется по модулю 2 (операция XOR) с (i-1)-м блоком шифртекста и затем шифруется. Начальное значение (в наших обозначениях c0) задается вектором инициализации.

Криптостойкость режима СВС определяется криптостойкостью используемого блочного шифра. Применение режима СВС позволяет устранить недостаток режима ЕСВ: каждый блок открытого текста маскируется блоком шифртекста, полученным на предыдущем этапе. Таким образом, возможность изменения открытого текста при использовании режима СВС весьма ограничена – любые манипуляции с блоками шифртекста, за исключением удаления первого и последнего блоков, будут обнаружены. Скорость обработки в данном режиме не ниже производительности блочного шифра – задержка при выполнении операции XOR пренебрежимо мала. Процедура шифрования в режиме СВС с трудом поддается распараллеливанию, процедуру расшифрования распараллелить значительно проще.

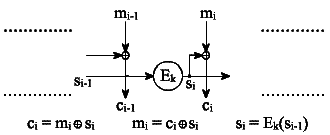
#### Шифрование в режимах CFB и OFB

В режиме СFВ i-й блок шифртекста формируется путем шифрования (i-1)го блока шифртекста и его суммированием (операция XOR) с i-м блоком открытого текста.

#### Режим СFВ

Режим СFВ (рис. 9.4) можно задать таким образом, что обратная связь будет захватывать не целый n-битный блок, а только k бит предыдущего блока, k <= n. Начальное значение c0 так же, как в режиме СВС, задается при помощи вектора инициализации.

Криптостойкость СFВ определяется криптостойкостью используемого шифра. Фиксированные блоки открытого текста «маскируются» блоками шифртекста. Возможности изменения открытого текста те же, что и в режиме СВС. Если в режиме СРВ с полноблочной обратной связью имеется два идентичных блока шифртекста, результат, например, DES-шифрования на следующем шаге будет тем же. Скорость шифрования СFВ-режима с полноблочной обратной связью та же, что и у блочного шифра, причем возможности распараллеливания процедуры шифрования ограничены.



***Рис. 9.5.*** *Режим Обратной связи по выходу (OFB)*

#### Режим OFB

Режим OFB (рис. 9.5) аналогичен СFВ, за исключением того, что суммируемые с открытым текстом биты генерируются независимо от открытого текста и шифртекста. Вектор инициализации s0 задает начальное значение последовательности блоков si, и каждый блок si получается путем шифрования предыдущего блока si–1. Открытый текст шифруется суммированием (операция XOR) iго блока открытого текста с si из независимой последовательности блоков.

Обратная связь по выходу на k разрядов не рекомендуется из соображений криптостойкости. Режим OFB имеет следующее преимущество по сравнению с режимом CFB: ошибки, возникающие в результате передачи по каналу с шумом, при расшифровании не «размазываются» по всему шифртексту, а локализуются в пределах одного блока. Однако открытый текст может быть изменен путем определенных манипуляций с блоками шифртекста. Скорость шифрования в режиме OFB та же, что и у блочного шифра. Несмотря на то, что OFBшифрование не поддается распараллеливанию, эффективность процедуры может быть повышена за счет предварительной генерации независимой последовательности блоков.

#### Шифрование в режимах усовершенствованного OFB и PCBC

Известные недостатки привели к появлению усовершенствованного варианта шифрования в режиме OFB. Основные изменения касаются метода генерации независимой последовательности блоков: для получения очередного блока предлагается шифровать не si, a si + IV(mod 264), где IV – некоторый вектор инициализации.

#### Режим шифрования РСВС

Режим шифрования РСВС (Propagating Cipher Block Chaining) применяется в протоколе Кеrbеrоs и позволяет обнаруживать ошибки. Данный режим шифрования не является федеральным или международным стандартом. Режим РСВС – вариант режима СВС, обладающий специфическим свойством – в результате расшифрования единичная ошибка распространяется на весь шифртекст (решается обратная задача с точки зрения режима OFB). Данное свойство позволяет с высокой надежностью обнаруживать ошибки, возникающие при передаче сообщений по каналам с шумом. Шифрование в режиме РСВС выполняется по правилу: ci = Ek(mi mi-1 ci-1), расшифрование:



mi = Dk(ci) ci-1 mi-1, где m0 m0 – вектор инициализации.



### 9.4. Алгоритмы блочного шифрования

#### Алгоритм DES

В 1973 году Национальное Бюро Стандартов США начало работы по созданию стандарта шифрования данных на ЭВМ. Был объявлен конкурс, который выиграла фирма IBM, представившая алгоритм шифрования, сейчас известный как DES-алгоритм (Data Encryption Standard).

Рассмотрим работу DES-алгоритма в простейшем (базовом) режиме ЕСВ – электронной кодовой книги. Алгоритм работы показан на рисунке 9.6.

Входные 64-битовые векторы, называемые блоками открытого текста, преобразуются в выходные 64-битовые векторы, называемые блоками шифртекста, с помощью 56-битового ключа К (число различных ключей равно 256=7\*106).

Алгоритм реализуется в 16 аналогичных циклах шифрования, где в i-м цикле используется цикловой ключ Ki, представляющий собой выборку 48 битов из 56 битов ключа К. Реализация алгоритма функции f показана на рисунке

9.7. Здесь операция Е – расширение 32-битового вектора до 48-битового, операция Sj(S-боксы) – замена 6-битовых векторов на 4-битовые.

Основным недостатком алгоритма считается 56-битовый ключ, слишком короткий для противостояния полному перебору ключей на специализированном компьютере. Недавние результаты показали, что современное устройство стоимостью 1 млн. долл. способно вскрыть секретный ключ с помощью полного перебора в среднем за 3,5 часа. Поэтому было принято решение использовать DES-алгоритм для закрытия коммерческой (несекретной) информации. В этих случаях практическая реализация перебора всех ключей экономически нецелесообразна, так как затраты не соответствуют ценности зашифрованной информации.

В ходе открытого обсуждения алгоритма в прессе рассматривались пути усиления его криптографических свойств. Наиболее простой вариант предполагал использовать независимые 48-битовые векторы в качестве цифровых ключей, что позволит увеличить общее число ключей до 2768.

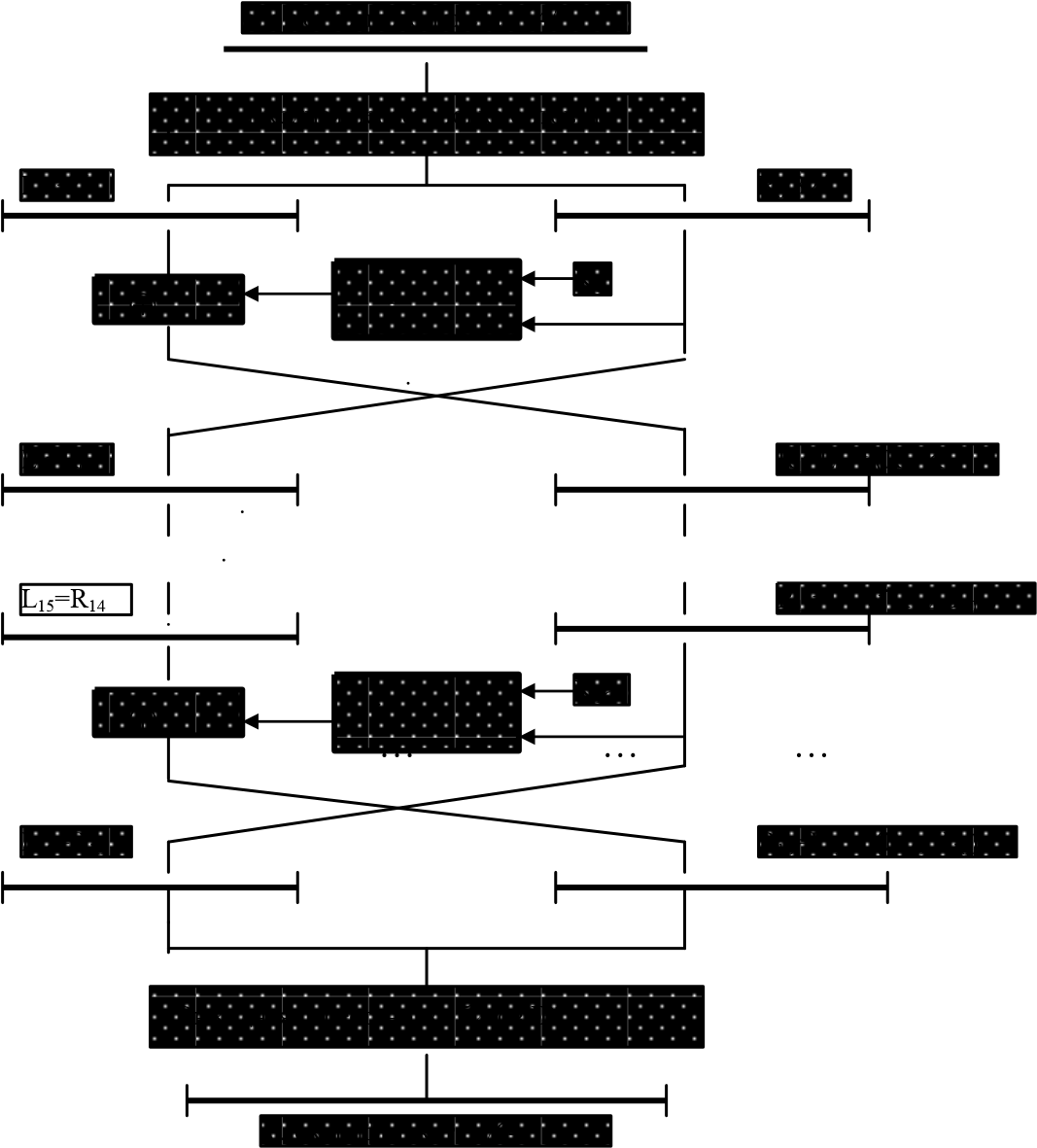
Режим электронной кодовой книги (ЕСВ) используется в основном для шифрования коротких сообщений служебного содержания – паролей, ключей и т.п. Наиболее общий режим – режим сцепления блоков (СВС, Cifer Block Chaining), схема которого показана на рисунке 9.3. Здесь каждый входной блок зависит от всех предыдущих. Начальный вектор bo (случайный начальный вектор) вырабатывается для каждого сообщения и может передаваться в линию связи как в открытом, так и в шифрованном виде, что препятствует атакам на шифртекст, основанным на наличии стандартов в начале сообщения.

DES – это алгоритм с эффективной длиной ключа в 56 бит (хотя часто говорят о 8 байтах, но старший бит в байте не используется).

Автор: National Institute of Standards and Technology (NIST). Параметры:

- размер блока 64 бита - размер ключа 56 бит - число раундов 16.

DES-алгоритм является первым примером широкого производства и внедрения технических средств в область защиты информации. К настоящему времени выпускается несколько десятков устройств аппаратно-программной реализации DES-алгоритма. Для выпуска такого рода устройства необходимо получить сертификат Национального Бюро Стандартов на право реализации продукта, который выдается только после всесторонней проверки по специальным тестирующим процедурам.



***Рис. 9.6.*** *Блок-схема DES-алгоритма*



R

i

-

1



E

E

=

(

R

i

-

1

)



/48/







K

i



S

1



S

8



S

2



S

3











Перестановка

**Р**

/32/

**f**

(

R

i

-

1

, K

i

)



/32/

***Рис. 9.7.*** *Блок-схема функции f DES-алгоритма*

Достигнута высокая скорость шифрования. По некоторым сообщениям, в одном из устройств на основе специализированной микросхемы она составляет около 45 Мбит/сек.

a0 a1 . . . an

DES

(

реж.

ECB)

DES

(

реж. Е

CB

)

. . .

. . .







DES

(

реж.Е

CB

)

b

0

b

1

b

2

b

n+1

***Рис. 9.8.*** *Реализация DES-алгоритма в режиме сцепления блоков*

Основные области применения DES-алгоритма:

* хранение данных в ЭВМ (шифрование файлов, паролей);
* электронная система платежей (между клиентом и банком);
* электронный обмен коммерческой информацией (между покупателем и продавцом).

#### Алгоритм Triple DES

Triple DES – улучшенный вариант стандарта DES. Его суть заключается в последовательном трехкратном применении шифрования к исходному тексту.

#### Алгоритм ГОСТ 28147

В 1989 году был разработан блочный шифр для использования в качестве государственного стандарта шифрования (зарегистрирован как ГОСТ 2814789)/7/. В основном алгоритм применяется в банковской системе, судя по публикациям – несколько медлителен, но обладает весьма высокой стойкостью.

Его общая схема близка к схеме DES-алгоритма, лишь отсутствует начальная перестановка и число циклов шифрования равно 32 (вместо 16 в DES-алгоритме). Ключом считается набор из 8 элементарных 32-битовых ключей X1, X2,...,X8 (общее число ключей 2256). В циклах шифрования трижды используется прямая последовательность элементарных ключей и один раз – обратная: X8,

X7,...,X1.

Основное отличие – в реализации функции f стандарта шифрования (приведена на рис. 9.9). Элементы S1, S2, ..., S8 представляют собой таблицы замены 4-битовых векторов и могут рассматриваться как долговременные ключи.

ГОСТ 28147-89, как DES-алгоритм, предусматривает различные режимы использования, и только базовый (режим простой замены) совпадает, по сути, с базовым режимом DES-алгоритма, остальные в той или иной мере отличаются.

Известна специальная реализация алгоритма шифрования ГОСТ 28147-89 – аппаратная плата шифрования данных «Криптон-3» для IBM PC, ее производительность – 50 Кбит/сек.

...

Сложение по модулю 2

32

г на 11 бит влево

Циклический сдви

R

i

-

1

/32/

K

i

f(R

i

-

1

,

K

i

)

/32/

/32/

S

1

S

2

S

3

S

8

***Рис. 9.9.*** *Блок-схема функции f**алгоритма ГОСТ 28147-89*

#### Алгоритм Lucifer

Это первый (опубликованный в открытой печати) блочный алгоритм.

Авторы: Horst Feisstel, Walter Tuchman (IBM). Тип – сеть Фейстеля.

Параметры:

* размер блока 128 бит - размер ключа 128 бит - число раундов 16.

В каждом используется подключ в 72 бита, порождаемый из главного ключа. Поскольку имеет больший размер ключа и блока по отношению к DES, поэтому более устойчив к криптоанализу.

|  |
| --- |
| ***SP-сеть***  SP-сеть (Substitution-Permutation network, подстановочно-перестановочная сеть) – разновидность блочного шифра, предложенная в 1971 году Хорстом Фейстелем. В простейшем варианте представляет собой «сэндвич» из слоев двух типов, используемых многократно по очереди (рис. 9.10). Первый тип слоя – P-слой, состоящий из P-блока большой разрядности, за ним идет второй тип слоя – S-слой, представляющий собой большое количество S-блоков малой разрядности, потом опять P-слой и т.д.  Первым криптографическим алгоритмом на основе SP-сети была первая версия алгоритма Lucifer (1971). В следующих версиях алгоритма вместо SPсети использовалась сеть Фейстеля, которая является альтернативой SP-сетям. В настоящее время из алгоритмов на основе SP-сетей широко используется AES.        ***Рис. 9.10.*** *Упрощенная схема S- и P-слоев в алгоритме Lucifer (1971)* |

***Алгоритм IDEA*** *(International Decryption-Encryption Algorithm)*

Время разработки – 1990–1991 годы, место разработки – Цюрих, Швейцария. Архитектура – общая сбалансированная шифрующая SP-сеть, инвариант раунда – побитовая сумма по модулю 2 старшей и младшей половин блока. Авторы: Xuejia Lai, James Massey.

Параметры:

* размер блока 64 бита - размер ключа 128 бит
* число раундов 8.

#### Алгоритм NewDES

Создан в 1985 г. как творческая переработка DES. Это самостоятельный алгоритм, а не вариант DES. NewDES несколько проще, чем DES, поскольку у него нет начальной и конечной перестановки. Операции производятся над байтами, а не над битами, как в DES. Brute-force атака на NewDES требует 2119 операций, против 2111 для TripleDES.

Автор: Robert Scott.

Параметры:

* размер блока 64 бита - размер ключа 120 бит
* число раундов 17.

#### Алгоритм RC2

Блочный алгоритм шифрования. Длина ключа переменная – от 8 до 1024 бит. Разрабатывался под 16-битное слово. Реализует 16 pаyндов «перемешивающих» (mixing) и 2 pаyнда «размазывающих» (mashing) пpеобpазований. Описан в RFC2268. Разработал Ron Rivest (RSA Laboratories).

Режимы: ECB, CBC, CFB 8 бит, OFB, OFB counter 8 бит.

ECB, CBC, OFB: шифруют данные блоками по 64 бита (8 байт) CFB, OFBC:

шифруют данне блоками по 8 бит (1 байту).

Автор: RSA Data Security (Ron Rivest)/RC – Ron's Code.

Параметры:

* размер блока 64 бита - размер ключа до 1024 бит
* число раундов 16.

#### Алгоритм RC5

Блочный шифр с переменными параметрами. Режимы: ECB, CBC, CFB 8бит, OFB, OFB counter 8бит. Шифр RC5 «словоориентированный»; все простейшие вычислительные операции производятся над w-битными словами. RC5 – блочный шифр с размерностью входного и выходного блоков 2 слова. Номинальный выбор для w – 32 бита, при котором входной и выходной блоки RC5 имеют размер 64 бита. В принципе, RC5 допускает любое значение w>0, однако для простоты принимают допустимые значения w – 16, 32 и 64 бита.

Число раундов r является вторым параметром RC5. Выбор большего числа раундов увеличивает степень защиты. Возможные значения для r: 0,1,...,255. Заметим также, что RC5 имеет расширенную ключевую таблицу S, получаемую из предоставляемого пользователем секретного ключа. Размер t таблицы S также зависит от числа раундов r и составляет t=2(r+1) слов. Выбор большего числа раундов, таким образом, увеличивает требования к памяти.

Для записи параметров RC5 применяют следующую нотацию: RC5-w/r/b. Например, запись RC5-32/16/10 означает, что используются 32-битные слова, 16 раундов и 10-байтовый (80-битный) секретный ключ, а также расширенная ключевая таблица размером 2(16+1)=34 слов. «Номинальным» набором параметров считается RC5-32/12/16 (размер слова 32 бита, число раундов – 12 и 16байтовый ключ). ECB, CBC, OFB: шифруют данные блоками по 64 бита (8 байт) CFB, OFBC: шифруют данные блоками по 8 бит (1 байту).

Автор: RSA Data Security (Ron Rivest) /RC – Ron's Code.

Параметры:

* размер блока 32/64/128 бит
* размер ключа до 2048 бит.

#### Алгоритм Rijndael

Данный алгоритм разработан двумя специалистами по криптографии из Бельгии (V.Rijmen & J.Daemen). Он является нетрадиционным блочным шифром, поскольку не использует сеть Фейстеля для криптографических преобразований. Алгоритм представляет каждый блок кодируемых данных в виде двумерного массива байт размером 4х4, 4х6 или 4х8 в зависимости от установленной длины блока. Далее на соответствующих этапах преобразования производятся либо над независимыми столбцами, либо над независимыми строками, либо вообще над отдельными байтами в таблице.

Все преобразования в шифре имеют строгое математическое обоснование. Сама структура и последовательность операций позволяют выполнять данный алгоритм эффективно как на 8-битных, так и на 32-битных процессорах. В структуре алгоритма заложена возможность параллельного исполнения некоторых операций, что на многопроцессорных рабочих станциях может еще поднять скорость шифрования в 4 раза.

Алгоритм состоит из некоторого количества раундов (от 10 до 14 – это зависит от размера блока и длины ключа), в которых последовательно выполняются следующие операции:

ByteSub – табличная подстановка 8х8 бит (рис. 9.11);

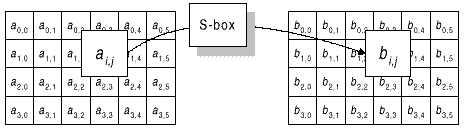


Рис. 9.11

ShiftRow – сдвиг строк в двумерном массиве на различные смещения (рис.

9.12);

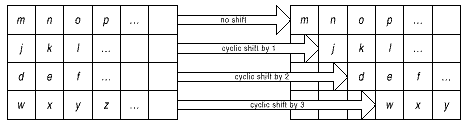


Рис. 9.12

MixColumn – математическое преобразование, перемешивающее данные внутри столбца (рис. 9.13);

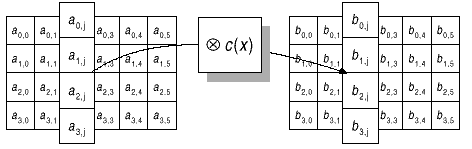


Рис. 9.13

AddRoundKey – добавление материала ключа операцией XOR (рис. 9.14).

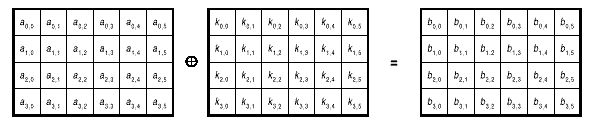


Рис. 9.14

В последнем раунде операция перемешивания столбцов отсутствует, что делает всю последовательность операций симметричной.

#### Алгоритм AES

Advanced Encryption Standard (AES), также известный как Rijndael (описан выше) – симметричный алгоритм блочного шифрования (размер блока 128 бит, ключ 128/192/256 бит). Принят в качестве стандарта шифрования правительством США по результатам конкурса AES. Национальный институт стандартов и технологий США (англ. National Institute of Standards and Technology,

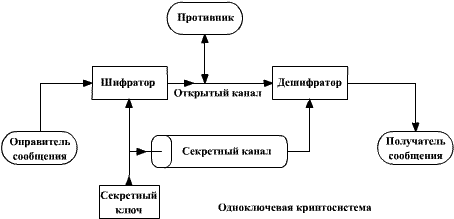
NIST) опубликовал спецификацию AES 26 ноября 2001 года после пятилетнего периода, в ходе которого были созданы и оценены 15 вариантов стандарта. 26 мая 2002 года AES был объявлен стандартом шифрования. AES является одним из самых распространенных алгоритмов симметричного шифрования.

#### Вопросы и задания для самостоятельной работы

1. Каковы с точки зрения криптографии преимущества и недостатки перехода к блочному шифрованию?
2. Как реализуется предложенный К. Шенноном принцип «перемешивания» при практической реализации алгоритмов блочного шифрования?
3. Каковы основные недостатки алгоритма DES и каковы пути их устранения?
4. В каких случаях можно рекомендовать использовать блочный шифр в режиме простой замены?
5. От каких потенциальных слабостей позволяет избавиться использование блочных шифров в режимах шифрования с обратной связью?
6. Какой режим блочного шифрования представляет собой, по сути, поточный шифр и какой?
7. В чем заключаются достоинства и недостатки систем блочного шифрования по сравнению с поточными криптосистемами?

## ТЕМА 10. СИСТЕМЫ ШИФРОВАНИЯ С ОТКРЫТЫМИ КЛЮЧАМИ

Способы противодействия противнику основаны на применении различных криптографических методов. Наиболее классическим из них является *симметричная* (одноключевая) схема передачи сообщений. Прежде чем передавать сообщение в открытый канал, отправитель шифрует сообщение с помощью *секретного ключа*. Получатель расшифровывает сообщение с помощью того же ключа. Противник не знает секретный ключ и не может так же просто дешифровать сообщение. Таким образом, его задача сводится к раскрытию ключа.

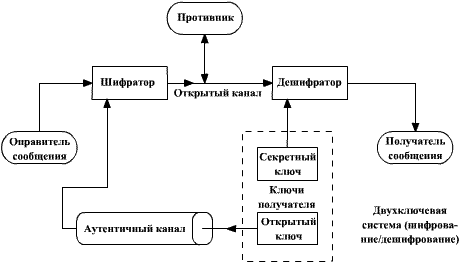


***Рис. 10.1.*** *Симметричная (одноключевая) криптосистема*

Для решения в первую очередь задачи распределения ключей была выдвинута концепция двухключевой (*асимметричной*) криптографии или криптографии *с открытыми ключами.*

### 10.1. Асимметричные системы

В такой системе (рис. 10.2) для шифрования и расшифрования используются различные ключи. Информация предназначена конкретному получателю. Получатель знает закрытый ключ. Отправитель располагает открытым ключом, парным закрытому ключу получателя. Отправитель зашифровывает сообщение с помощью открытого ключа. Расшифровать сообщение можно только с помощью закрытого ключа получателя.



***Рис. 10.2.*** *Асимметричная (двухключевая) криптосистема*

Для передачи открытого ключа от получателя к отправителю секретный канал не нужен. Вместо него используется аутентичный канал, гарантирующий подлинность источника информации. Он открыт и доступен криптоаналитику противника. Однако механизм аутентификации позволяет обнаружить попытки нарушения целостности и подлинности передаваемой информации. Отсутствие аутентификации позволило бы противнику заменить открытый ключ получателя на свой собственный и получать в последующем всю адресованную получателю информацию.

Уникальным свойством двухключевых систем является возможность доказательства принадлежности в случае отказа отправителя/получателя от ранее переданного/принятого сообщения. Эта возможность достигается применением

*цифровой подписи*. Цифровая подпись (ЦП) обеспечивает также аутентификацию и контроль целостности.

Современная технология шифрования основана на применении двух различных ключей. Одноразовый псевдослучайный сеансовый ключ используется для шифрования/расшифрования сообщений в течение одного сеанса. В свою очередь, долговременный ключ используется для шифрования/расшифрования сеансовых ключей. Данный метод имеет ряд преимуществ по сравнению с использованием долговременного ключа в качестве сеансового.

Применение асимметричных систем приводит к существенным задержкам при шифровании (по сравнению с симметричными). В целях минимизации задержки используют гибридную схему, где асимметричная система используется для шифрования короткого сеансового ключа, а информационные потоки в течение сеанса проходят по симметричной системе. То есть долговременный ключ принадлежит асимметричной системе, а сеансовые ключи – симметричной. Данный способ шифрования известен как *метод цифрового конверта* и широко используется на практике.

Аутентификация отправителя и контроль целостности сообщений обеспечиваются цифровой подписью. Отправитель шлет получателю сообщение, полученное конкатенацией некоего сообщения и результата шифрования этого сообщения на секретном ключе. При этом размер результата шифрования равен размеру исходного сообщения (в символах сообщения). Получатель, зная открытый ключ, может выполнить расшифрование и тем самым осуществить аутентификацию источника и контроль целостности.

Использование совместно с цифровой подписью специальной хешфункции позволяет избежать удвоения размера передаваемого сообщения. Алгоритм ее вычисления известен отправителю и получателю.

В специальной литературе для обозначения результата вычисления хешфункции нередко используется термин *дайджест* сообщения (message digest), или цифровой отпечаток пальца (digital fingerprint).

Таким образом, процедура вычисления цифровой подписи сводится к последовательному вычислению хеш-функции от исходного сообщения и шифрованию полученного значения на секретном ключе отправителя (или расшифрованию на открытом ключе при проверке подписи).

В случае когда отправитель и получатель знают один и тот же сеансовый ключ, целостность и аутентичность сообщений можно обеспечить, вычислив значение хеш-функции от объединения (конкатенации) передаваемого сообщения и сеансового ключа. Результат вычисления называется кодом аутентификации сообщения (КАС) (message authentication code, МАС). КАС предназначен для защиты от навязывания ложных сообщений со стороны третьих лиц.

Другой способ, позволяющий контролировать целостность передаваемых сообщений, основан на использовании кода целостности сообщений (КЦС) (message integrity check, MIC), в другой терминологии – имитовставки. Как и в случае КАС, предполагается, что отправитель и получатель знают общий сеансовый ключ. Для симметричной криптосистемы вычисление КЦС состоит из вычисления хеш-функции от передаваемого сообщения с последующим шифрованием полученного значения на сеансовом ключе. Для асимметричной системы контроль целостности обеспечивается за счет цифровой подписи.

Отметим, что в случае симметричной криптосистемы КЦС не обеспечивает доказательства принадлежности при отказе отправителя от ранее переданного сообщения и не позволяет доказать факт подделки сообщения в случае компрометации отправителя со стороны получателя (что всегда возможно, т.к. получатель знает ключ отправителя).

В асимметричной системе никто, кроме отправителя подписанного сообщения, не знает секретного ключа, на котором сообщение было подписано. Данное обстоятельство позволяет однозначно доказывать принадлежность при отказе отправителя/получателя от ранее переданного/принятого сообщения. Кроме того, получатель, не зная секретного ключа, не может подписать сообщение от лица отправителя и скомпрометировать его.

### 10.2. Открытое распределение ключей. Схема Диффи–Хеллмана

Важной составной частью шифрсистемы является ключевая система шифра. Под ней понимается описание всех видов ключей (долговременных, суточных, сеансовых и других) и алгоритмов их использования.

Одной из основных характеристик ключа является его размер, определяющий число всевозможных ключевых установок шифра. Если размер ключа чрезмерно велик, то это приводит к удорожанию изготовления ключей, усложнению процедуры установки ключа, понижению надежности работы шифрующего устройства.

Важнейшей частью практической работы с ключами является обеспечение секретности ключа. К основным мерам по защите ключей относятся следующие:

* ограничение круга лиц, допущенных к работе с ключами;
* регламентация рассылки, хранения и уничтожения ключей;
* регламентация порядка смены ключей;
* применение технических мер защиты ключевой информации от несанкционированного доступа.

Рассмотрим один из принципов распределения ключей (на основе односторонней функции), проработка которого имела весьма неожиданные последствия – была изобретена система шифрования с открытым ключом. Сначала небольшое отступление.

Понятие односторонней функции было введено в теоретическом исследовании о защите входа в вычислительные системы. Функция f(x) называется односторонней (one-way function), если для всех x из ее области определения легко вычислить y=f(x), но нахождение по заданному y0 такого x0, для которого f(x0)=y0, вычислительно неосуществимо, то есть требуется настолько огромный объем вычислений, что за них просто и не стоит браться.

Однако существование односторонних функций не доказано. В качестве приближения была предложена Дж. Гиллом целочисленная показательная функция f(x)=ax(mod n), где основание a и показатель степени x принадлежат интервалу (1, n-1), а умножение ведется по модулю n (3\*4 mod 10=2; 7\*8mod 9=2). Функция вычисляется достаточно эффективно по схеме Горнера. Если представление числа x в двоичной форме имеет вид:

xk-12k-1 + xk-22k-2 + ...+ x121 + x020,

то

y = f(x) = ax mod n = ((...(axk-1)2\*axk-2)2\*...\*ax1)2\*ax0 mod n.

Операция, обратная к этой, известна как операция *вычисления дискретного логарифма*: по заданным y, a и n найти такое целое x, что aх(mod n) = y. До настоящего времени не найдено достаточно эффективных алгоритмов решения этой задачи.

Американские криптологи Диффи и Хеллман (Diffi W., Hellman M.E. New direction in criptography. IEEE Trans. Inf. Theory, v. IT-22, 1976) предложили схему распространения (рассылки) ключей для секретной связи на основе односторонней показательной функции. Ее суть состоит в следующем.

В протоколе обмена секретными ключами предполагается, что все пользователи знают некоторые числа **n** и **a** (1< a < n). Для выработки общего секретного ключа пользователи A и B должны проделать следующую процедуру:

1. Определить секретные ключи пользователей КА и КВ.

Для этого каждый пользователь независимо выбирает случайные числа из интервала (1,..., n-1).

1. Вычислить открытые ключи пользователей YA и YB.

Для этого каждый использует одностороннюю показательную функцию Y=aK mod n со своим секретным ключом.

1. Обменяться ключами YA и YB по открытому каналу связи.
2. Независимо определить общий секретный ключ К.

Для этого пользователи выполняют вычисления с помощью той же односторонней функции.

**A:** YBKA(mod n) = [**a**KB ]KA mod n = **a**KA\*KB mod n = K.

**B:** YAKB(mod n) = [**a**KA ]KB mod n = **a**KB\*KA mod n = K.

Здесь каждый имеет показатель степени, а основание получает от партнера.

Безопасность (секретность) изложенной схемы зависит от сложности вычисления секретных ключей пользователей (КА и КВ). Пока не найдено удовлетворительных быстрых алгоритмов нахождения К из а, YA и YB без явного определения КА или КВ.

### 10.3. Криптосистема RSA

Название криптосистемы образовано из первых букв фамилий предложивших ее авторов (Rivest R., Shamir A., Adleman L. A method for obtaining digital signatures and pablic-key cryptosystems. Commun. ACM, v.21, N2, 1978).

Система относится к блочным экспоненциальным системам, так как каждый блок М открытого текста рассматривается как целое число в интервале

(0,..., n-1) и преобразуется в блок С следующим открытым преобразованием: C = E (e,n) (M) = Me (mod n), где E (e,n) – преобразование, а (e,n) – ключ зашифрования.

При расшифровании блок открытого текста М восстанавливается таким же преобразованием, но с другим показателем степени. M = D (d,n) (C) = Cd (mod n), где D (d,n) – преобразование, а (d,n) – ключ расшифрования.

В основе этого метода лежит довольно сложное теоретическое обоснование. Числа e и d связаны с n определенной зависимостью, и существуют рекомендации по выбору ключевых элементов на основе простых чисел. Если взять два простых числа p и q, определить n = p х q, то можно определить пару чисел e и d, удовлетворяющих заданным условиям. Если сделать открытыми числа e и n, а ключ d (и обязательно p и q) держать в секрете, то предложенная система является RSA-криптосистемой открытого шифрования. Очевидно, ее стойкость определяется сложностью операции извлечения из С корня степени е по модулю n.

***Принцип работы шифрсистемы*** Рассмотрим основные этапы реализации алгоритма RSA.

*Шифрование:*

1. Отправитель вычисляет *n = p ∙ q* и функцию Эйлера от *n φ(n)*:

2.

φ(n)=(p-1)(q-1).

1. Затем он выбирает случайное целое число e, взаимно простое с φ(n) и вычисляет d, мультикативное обратное e по модулю φ(n), т.е. удовлетворяющее условию

*ed ≡ 1 (mod φ(n)).*

Напомним, что два числа являются взаимно простыми, если их HOD =1.

1. После этого он публикует *е* и *n* как свой открытый ключ шифрования, сохраняя d как закрытый (секретный) ключ.
2. Рассмотрим теперь числа *e* и *d*. Предположим, что мы знаем одно из них и знаем соотношение, которым они связаны. Мы могли бы легко вычислить второе число, однако мы не знаем чисел *p* и *q*. Следовательно, можно одно из чисел подарить кому-нибудь вместе с *n* и попросить его посылать нам сообщения следующим образом.
3. Сообщение представить как векторы (блоки) длины *l:*

*X= (x1,x2...,xl); 0<xi< n.* 6. Каждое xi возвести в степень *e* по mod *n*.

7. Прислать нам *Y=(x1e (mod n), x2e (mod n),...,xle (mod n)).* *Расшифрование:*

Обозначим *t=yi ≡хie (mod n)* и рассмотрим расшифрование полученной информации. Для этого возведем полученное число *t* в степень второго числа пары – *d*: *R=td(mod n) ≡ xe (mod n)d(mod n) ≡ xed (mod n).*

В соответствие с п.2 соотношение *ed≡1(mod φ(n))*, а это означает, что *ed–1* делится на *(p–1)(q–1)* без остатка, т.е. *ed=1+a(p–1)(q–1)*, где *а –* целое число. Утверждается, что *хed mod n =x.*

*Доказательство:*

Действительно, *xed(mod n) = x1+a(p-1)(q-1) (mod n)* Учитывая,что

*xp-1 ≡ 1(mod p), xq-1 ≡ 1(mod q)* (эти соотношения доказываются как малая

теорема Ферма) получим *x(p-1)(q-1) ≡ 1(mod pq) x1+a(p-1) (q-1)(mod n) = x,* т.к.

*xa(p-1) (q-1) ≡ 1(mod pq),* из-за того, что *x(p-1) (q-1) ≡ 1(mod pq),* *x mod n = x*, так как *x < n*.

Что и требовалось доказать.

#### Размер ключа

Размер ключа в алгоритме RSA связан с размером модуля *n*. Два числа *p* и *q*, произведением которых является модуль, должны иметь приблизительно одинаковую длину, поскольку в этом случае найти сомножители (факторы) сложнее, чем в случае, когда длина чисел значительно различается. Например, если предполагается использовать 768-битный модуль, то каждое число должно иметь длину приблизительно 384 бита.

Если два числа чрезвычайно близки друг к другу или их разность близка к некоторому предопределенному значению, то возникает потенциальная угроза безопасности, однако такая вероятность – близость двух случайно выбранных чисел – незначительна.

1. Возьмем *x = (p + q) /2* и *y = (p – q)/*2, тогда *x2 – y2 = p ∙ q = n*.
2. При *p > q*, имеем *x* ≈ *n* .

Поскольку *y =*  x - n2 *),* то значения *p* и *q* можно легко найти, если раз-

ность (*p – q)* достаточно мала.

Оптимальный размер модуля определяется требованиями безопасности: модуль большего размера обеспечивает большую безопасность, но и замедляет работу алгоритма RSA. Длина модуля выбирается в первую очередь на основе значимости защищаемых данных и необходимой стойкости защищенных данных и во вторую очередь – на основе оценки возможных угроз.

#### Область применения

Криптосистема RSA используется в самых различных продуктах, на различных платформах и во многих отраслях. В настоящее время криптосистема RSA встраивается во многие коммерческие продукты, число которых постоянно увеличивается. Также ее используют операционные системы Microsoft, Apple, Sun и Novell. В аппаратном исполнении RSA алгоритм применяется в защищенных телефонах, на сетевых платах Ethernet, на смарт-картах, широко используется в криптографическом оборудовании THALES (Racal). Кроме того, алгоритм входит в состав всех основных протоколов для защищенных коммуникаций Internet, в том числе S/MIME, SSL и S/WAN, а также используется во многих учреждениях, например, в правительственных службах, в большинстве корпораций, в государственных лабораториях и университетах.

### 10.4. Схема шифрования Эль-Гамаля

Шифросистема Эль-Гамаля (Elgamal) была предложена в 1984 году. Стандарты электронной цифровой подписи в США (DSA – Digital Signature Algorithm) и России (ГОСТ Р34.10-2012) базируются именно на ней.

***Принцип работы шифрсистемы*** Будем понимать под М – исходное сообщение.

*Шифрование:*

1. Выбирается большое простое число p, p > M. Для проверки р на простоту чаще всего используется алгоритм Миллера–Рабина.

Перед проверкой числа алгоритмом Миллера–Рабина рекомендуется выполнить следующие действия:

* 1. В сгенерированном числе установить младший бит равным 1, это обеспечит нечетность числа.
  2. Убедиться, что р не делится на небольшие простые числа 3, 5, 7, 11 и т.д. В некоторых реализациях проверяется делимость р на все простые числа меньше 256. Проверка делимости p на простые числа 3, 5 и 7 отсекает 54% нечетных чисел, проверка делимости р на простые числа меньше 100 убирает 76% нечетных чисел, а проверка делимости р на простые числа, меньшие 256, убирает 80% нечетных чисел.
  3. Выполнить тест Миллера–Рабина несколько раз. Если p не проходит хотя бы одной проверки – число не простое.

1. Выбираются числа x и g так, что 1 < x < p, 1 < g < p.
2. Вычисляется

*y g* *x*mod*p*.

Таким образом, открытый ключ: p, g, y. Закрытый ключ: x.

1. Выбирается случайное секретное число k (рандомизатор), взаимно простое с функцией Эйлера от числа р:

φ(p) = p − 1,

то есть НОД(k, φ(p)) = 1. 5. Вычисляется

* + 1.  *gk* mod *p*
    2. *y* *kM*mod*p*.

Пара чисел a и b является шифртекстом. Причем длина шифртекста получается длинней открытого текста вдвое.

*Расшифрование:*

Для расшифрования a и b вычисляем:

M = b / ax mod p.

*Доказательство:*



(

)

Стойкость данной схемы основана на сложности проблемы дискретного логарифмирования (по известным p, g и y приходится искать показатель степени х):

*y g* *x*mod*p*.

Введение в правило зашифрования рандомизатора k делает шифр ЭльГамаля *шифром многозначной замены*. В связи со случайным характером выбора параметра k подобная схема шифрования называется еще *схемой вероятностного шифрования*. Для нее текст и ключ не определяют шифртекст однозначно. Схемы вероятностного шифрования обладают, как правило, большей стойкостью по сравнению с детерминированными процессами шифрования.

### 10.5. Криптография на эллиптических кривых

Под *эллиптической криптографией* подразумевается все множество криптографических алгоритмов, использующих свойства эллиптических кривых над конечными полями. Основное преимущество эллиптической криптографии заключается в том, что на сегодняшний день не известно эффективных алгоритмов для решения задачи дискретного логарифмирования в полях точек эллиптических кривых.

Прогресс математики и вычислительной техники вызывает рост размеров блоков данных и ключей. Так, если в момент создания криптосистемы RSA считался достаточным размер чисел в 512 бит, то на момент создания настоящей книги рекомендуется не менее 4 Кбит. Иными словами, «безопасный» размер чисел в RSA вырос практически на порядок. Необходимость постоянно увеличивать размеры блоков данных и ключей является проблемой в современной криптографии.

Самый очевидный путь решения вышеупомянутой проблемы – представление блоков информации в криптографических алгоритмах не только в виде чисел (или элементов конечных полей), но и в виде иных алгебраических объектов большей сложности. Одним из весьма подходящих типов таких объектов являются точки эллиптических кривых.

#### Принцип работы шифрсистемы

Рассмотрим самый простой подход к шифрованию с использованием *эллиптических кривых*. Задача состоит в том, чтобы зашифровать сообщение М, которое может быть представлено в виде точки на эллиптической кривой Pm

(x,y).

В системе шифрования в качестве параметров рассматривается *эллиптическая кривая* Ep (a,b) и точка G на ней.

Участник B выбирает закрытый ключ nB и вычисляет открытый ключ PB = nB × G.

Чтобы зашифровать сообщение Pm, используется открытый ключ получателя B PB.

Участник А выбирает случайное целое положительное число k и вычисляет зашифрованное сообщение Cm, являющееся точкой на *эллиптической кривой*. Cm = {k × G, Pm + k × PB}.

Чтобы расшифровать сообщение, участник В умножает первую координату точки на свой закрытый ключ и вычитает результат из второй координаты:

Pm + k × PB – nB × (k × G) = = Pm + k × (nB × G) – nB × (k × G) = Pm.

Участник А зашифровал сообщение Pm добавлением к нему kxPB. Никто не знает значения k, поэтому, хотя PB и является открытым ключом, никто не знает k × PB. Противнику для восстановления сообщения придется вычислить k, зная G и k × G. Сделать это будет нелегко.

Получатель также не знает k, но ему в качестве подсказки посылается k ×

G. Умножив k × G на свой закрытый ключ, получатель получит значение, которое было добавлено отправителем к незашифрованному сообщению. Тем самым получатель, не зная k, но имея свой закрытый ключ, может восстановить незашифрованное сообщение.

Практически любая асимметричная криптосистема может быть реализована на эллиптических кривых, однако не для всех схем это дает выигрыш в стойкости. Например, для системы RSA и родственных ей систем, основанных на сложности задачи факторизации, это не усиливает схему. В то же время для схем, основанных на сложности задачи логарифмирования в дискретных полях, переход на эллиптические кривые позволяет существенно увеличить стойкость. Объясняется это тем, что при надлежащем выборе параметров кривой задача логарифмирования в группе точек кривой существенно сложнее задачи логарифмирования в мультипликативной группе исходного поля. Этот факт в сочетании с необходимостью постоянно увеличивать размеры блоков данных и ключей в современной криптографии привел к переходу на эллиптические кривые во многих областях применения. Так, старые стандарты ЭЦП РФ и США, просуществовав около 7 лет, с 1994 по 2001 гг., практически одновременно были заменены новыми, реализующими прежние криптографические схемы на эллиптических кривых, что позволило существенно увеличить стойкость и сократить размер блоков данных. Старый российский стандарт оперировал 1024битовыми блоками данных, новый оперирует 256-битовыми. По оценкам специалистов, трудоемкость взлома старого и нового стандартов ЭЦП России составляет величину порядка 1026 и 1038 операций умножения в базовом поле GF(p) соответственно. По указанной причине перевод асимметричных криптосистем, основанных на сложности задачи логарифмирования в дискретных полях, на эллиптические кривые может повысить их криптостойкость.

### 10.6. Криптоанализ шифра RSA

Существует несколько способов взлома шифра RSA. Наиболее эффективная атака: найти закрытый ключ, соответствующий необходимому открытому ключу. Это позволит нападающему читать все сообщения, зашифрованные открытым ключом, и подделывать подписи. Такую атаку можно провести, найдя главные сомножители (факторы) общего модуля n – p и q. На основании p, q и e (общий показатель) нападающий может легко вычислить частный показатель d. Основная сложность – поиск главных сомножителей (факторизация) n. Безопасность RSA зависит от разложения на сомножители, что является трудно разрешимой задачей, не имеющей эффективных способов решения.

Фактически задача восстановления секретного ключа эквивалентна задаче разложения на множители (факторизации) модуля: можно использовать d для поиска сомножителей n, и наоборот, можно использовать n для поиска d. Надо отметить, что усовершенствование вычислительного оборудования само по себе не уменьшит стойкость криптосистемы RSA, если ключи будут иметь достаточную длину. Фактически же совершенствование оборудования увеличивает стойкость криптосистемы.

Другой способ взломать RSA состоит в том, чтобы найти метод вычисле-

*e* (mod n), то корнем степени e ния корня степени e из (mod n). Поскольку С = M

из (mod n) является сообщение M. Вычислив корень, можно вскрыть зашифрованные сообщения и подделывать подписи, даже не зная закрытый ключ. Такая атака не эквивалентна факторингу, но в настоящее время неизвестны методы, которые позволяют взломать RSA таким образом. Однако в особых случаях, когда на основе одного и того же показателя относительно небольшой величины шифруется достаточно много связанных сообщений, есть возможность вскрыть сообщения. Упомянутые атаки – единственные способы расшифровать все сообщения, зашифрованные данным ключом RSA.

Существуют и другие типы атак, позволяющие, однако, вскрыть только одно сообщение и не позволяющие нападающему вскрыть прочие сообщения, зашифрованные тем же ключом.

Самое простое нападение на единственное сообщение – атака по предполагаемому открытому тексту. Нападающий, имея зашифрованный текст, предполагает, что сообщение содержит какой-то определенный текст, например «Нападение на рассвете», затем шифрует предполагаемый текст открытым ключом получателя и сравнивает полученный текст с имеющимся зашифрованным текстом. Такую атаку можно предотвратить, добавив в конец сообщения несколько случайных битов.

Другая атака единственного сообщения применяется в том случае, если кто-то посылает одно и то же сообщение M трем корреспондентам, каждый из которых использует общий показатель e = 3. Зная это, нападающий может перехватить эти сообщения и расшифровать сообщение M. Такую атаку можно предотвратить, вводя в сообщение перед каждым шифрованием несколько случайных бит.

Также существуют несколько атак по зашифрованному тексту (или атаки отдельных сообщений с целью подделки подписи), при которых нападающий создает некоторый зашифрованный текст и получает соответствующий открытый текст, например, заставляя обманным путем зарегистрированного пользователя расшифровать поддельное сообщение.

Разумеется, существуют и атаки, нацеленные не на криптосистему непосредственно, а на уязвимые места всей системы коммуникаций в целом; такие атаки не могут рассматриваться как взлом RSA, так как говорят не о слабости алгоритма RSA, а скорее об уязвимости его конкретной реализации. Например, нападающий может завладеть закрытым ключом, если тот хранится без должных предосторожностей. Необходимо подчеркнуть, что для полной защиты недостаточно защитить выполнение алгоритма RSA и принять меры вычислительной безопасности, то есть использовать ключ достаточной длины. На практике же наибольший успех имеют атаки на незащищенные этапы управления ключами системы RSA.

#### Вопросы и задания для самостоятельной работы

1. В чем состоят преимущества систем с открытыми ключами перед симметричными шифрсистемами?
2. Сложностью какой математической задачи определяется стойкость системы RSA?
3. Какие требования предъявляются к ключам в шифре RSA?
4. К какому типу шифров принадлежит схема шифрования, используемая в системе Эль-Гамаля? В чем ее преимущества?
5. Сложностью какой математической задачи определяется стойкость шифрсистемы Эль-Гамаля?
6. Назовите недостатки схемы Эль-Гамаля.
7. Какие проблемы информационной безопасности можно решить с помощью асимметричных шифров?
8. Изложите принципиальную схему организации секретной связи с использованием шифрсистемы с открытым ключом.
9. Каким образом с помощью криптосистемы RSA можно организовать передачу сообщений, подлинность которых мог бы проверить любой получатель?
10. Каким образом с помощью криптосистемы RSA можно организовать передачу сообщений, подлинность авторства которых можно при необходимости доказать?

## ТЕМА 11. ЭЛЕКТРОННЫЕ ЦИФРОВЫЕ ПОДПИСИ

Одним из основных применений криптосистем с открытым ключом является их использование при создании так называемой цифровой или электронной подписи (digital signature). Впервые идея цифровой подписи была высказана в работе Диффи и Хеллмана.

Большое значение аппарата электронных цифровых подписей в современном мире отражается принятием национальных (ГОСТ, СТБ, ECDSA, ESIGN) и коммерческих (RSA) стандартов цифровой подписи. Рассмотренные выше стандарты определяют технические аспекты применения цифровых подписей.

Однако возникают вопросы юридического характера.

* Как придать документам, существующим только в электронном виде, тот же правовой статус, который имеется у бумажных документов?
* Как обеспечить защищенный, надежный и юридически законный метод подписи электронных документов, который исключит необходимость создавать и подписывать бумажные документы, тем самым поощряя и облегчая электронную коммерцию?

Эти два вопроса носят скорее юридический характер, поэтому в настоящее время в большинстве развитых стран приняты нормативные акты, регулирующие взаимоотношения субъектов, использующих ЭЦП. Но ЭЦП – это не только инструмент для ведения электронного документооборота, но и новый интересный криптографический примитив, поэтому много математиков занимаются исследованиями в области схем цифровой подписи и идентификации.

### 11.1. Механизм действия электронной цифровой подписи

#### Основные определения

*Цифровая подпись* – строка данных, связывающая сообщение (в цифровом виде) с некоей сущностью, его породившей.

Алгоритм генерации цифровой подписи – метод получения цифровой подписи.

Алгоритм проверки цифровой подписи – метод установления подлинности цифровой подписи (то есть, что подпись действительно была создана указанной сущностью).

Схема (или механизм) цифровой подписи состоит из алгоритма генерации подписи и соответствующего алгоритма проверки.

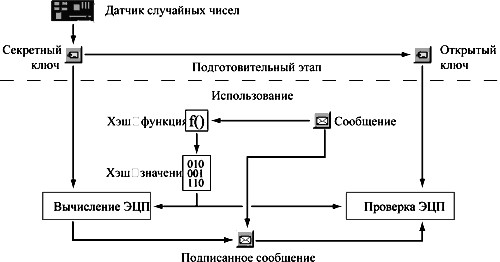
Процесс подписывания состоит из математического алгоритма генерации цифровой подписи и метода форматирования данных в сообщения, которые могут быть подписаны.

Процесс проверки цифровой подписи состоит из алгоритма проверки и метода восстановления данных из сообщения.

Рассмотрим на простом примере, как устроена исполнительная процедура электронной цифровой подписи. Предположим, есть два пользователя i и j, и один из них (пусть это будет пользователь i) решил послать другому важные данные, используя ЭЦП. Для этого пользователи должны выполнить следующие действия (рис. 11.1), состоящие, по сути, из двух этапов: подготовительного и исполнения передачи.

На подготовительном этапе пользователь i создает пару своих личных ключей ЭЦП: секретный ключ Di и открытый ключ Ei. Открытый ключ Ei вычисляется из соответствующего ему секретного ключа; обратное же вычисление невозможно (аналогично тому, как это происходит с алгоритмами асимметричного шифрования). После этого пользователь i передает открытый ключ Ei пользователю j.

На этапе исполнения процедуры пользователь i перед отправкой сообщения подписывает его с помощью своего секретного ключа Di. Заметим, что пользователь j может проверить подпись пользователя i в сообщении с помощью Kpi – открытого ключа абонента i.



***Рис. 11.1.*** *Схема использования ЭЦП*

Если же пользователь j тоже захочет ответить пользователю i подписанным сообщением, то он должен создать собственные ключи и передать пользователю i свой открытый ключ для проверки ЭЦП.

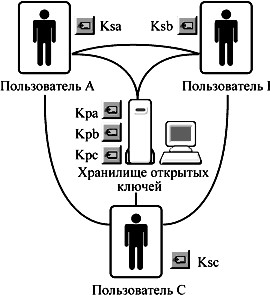
С теоретической точки зрения электронная подпись представляет собой последовательность фиксированной длины (длина эта зависит от конкретного алгоритма ЭЦП), которая вычисляется определенным образом с помощью содержимого подписываемой информации и секретного ключа.

Секретный ключ ЭЦП также есть не что иное, как последовательность определенной длины. Это тот самый уникальный элемент, без знания которого невозможно подделать ЭЦП его владельца. Соответственно, пользователь должен хранить свой секретный ключ таким образом, чтобы никто другой не смог «выведать» его значение. Если же у владельца ключа есть основания полагать, что ключ известен кому-либо еще, то такой секретный ключ ЭЦП считается скомпрометированным, и «потерпевший», допустивший компрометацию своего ключа, должен оповестить всех остальных корреспондентов, что его открытый ключ следует считать недействительным. После оповещения легко создать новую пару ключей ЭЦП, гарантирующую авторство писем. Механизмы такого оповещения определены, например, международным стандартом X.509.

Если же этого не сделать, подпись «потерпевшего», утерявшего свой ключ, может появиться под любым электронным документом. И тогда, по аналогии с обычной подделкой документов, можно сказать, что злоумышленник, завладевший чужим ключом, научился удачно подделывать подпись его владельца, да вдобавок еще и получил возможность ставить его печать.

Открытый ключ, наоборот, должен быть доступен всем, с кем данный корреспондент намерен обмениваться информацией, заверенной личной подписью. Например, он может храниться на каком-нибудь разделяемом ресурсе.

Естественно, количество пользователей такой системы может быть сколь угодно большим; каждый из них (рис. 11.2) должен, во-первых, иметь и надежно хранить собственный секретный ключ, а во-вторых – получить доступ к открытым ключам остальных корреспондентов.



***Рис. 11.2.*** *Распределение ключей ЭЦП*

Физическое представление ключей ЭЦП зависит от конкретной системы, поддерживающей использование ЭЦП. Чаще всего ключ записывается в файл, который, в дополнение к самому ключу, может содержать, например, информацию о пользователе – владельце ключа, о сроке действия ключа, а также некий набор данных, необходимых для работы конкретной системы.

Данные о владельце ключа позволяют реализовать «побочную», но важную функцию ЭЦП – установление авторства, поскольку при проверке подписи сразу же становится ясно, кто подписал то или иное сообщение. Обычно программы, осуществляющие проверку ЭЦП, настраиваются так, чтобы результат исполнения появлялся на экране в удобном для восприятия виде и с указанием поставившего ЭЦП пользователя, например, так:

*«Подпись файла message.doc верна (Автор: Иванов Иван*

*Иванович)»*

### 11.2. Функции хэширования

Как и для всякой последовательности, существует формула вычисления ЭЦП, которую математически можно представить как:

S = f (h(M), D),

где M – текст сообщения, D – секретный ключ, h(M) – функция хэширования.

Согласно приведенной зависимости, для формирования ЭЦП в качестве исходного значения берется не само сообщение, а его хэш (результат обработки сообщения хэш-функцией). При этом используются *бесключевые односторонние хэш-функции*.

Бесключевые односторонние хэш-функции формируют выход фиксированной длины и построены на односторонней функции, которая производит значение длины *n* по двум входным блокам длиной *n* каждый. Обычно входом функции служат блок текста и хэш предыдущего блока. Хэш последнего блока становится хэшем всего сообщения. Таким образом удается произвести хэш фиксированной длины независимо от длины входного сообщения.

Дело в том, что заверяемый подписью текст может быть абсолютно произвольного размера: от пустого сообщения до многомегабайтного файла, содержащего, например, графическую информацию. Но практически все применяемые алгоритмы вычисления ЭЦП используют для расчета сообщения заранее заданной стандартной длины (например, в отечественном алгоритме ЭЦП ГОСТ Р 34.10-94 этот размер определен равным 32 байтам). Задача хэшфункции – из сообщения произвольной длины вычислить цифровую последовательность нужного размера (скажем, 32 байта).

Хэш-функция должна удовлетворять определенным требованиям. Прежде всего необходимо, чтобы результат (хэш сообщения) однозначно соответствовал исходному сообщению и изменялся при любой модификации последнего, даже самой незначительной. Кроме того, хэш сообщения должен вычисляться таким образом, чтобы для любого сообщения M было бы невозможно подобрать такое сообщение M', для которого h(M) = h(M').

Другими словами, трудоемкость успешного вычисления сообщения M' по известному сообщению M и его хэшу h(M), удовлетворяющего условию h(M') = h(M), должна быть эквивалентна трудоемкости прямого перебора сообщений. Невыполнение этого условия создало бы возможность для злоумышленника подменять сообщения, оставляя их подпись верной.

С другой стороны, очевидно, что хэш будет одинаков для многих сообщений, поскольку множество возможных сообщений существенно больше множества возможных хэш-значений (действительно, количество сообщений безгранично, а количество хэш-значений ограничено числом 2N, где N – длина хэшзначения в битах).

К числу наиболее известных функций хэширования принадлежат следующие:

Отечественный стандарт ГОСТ Р 34.11-94. Вычисляет хэш-значение размером 32 байта.

MDx (Message Digest) – семейство алгоритмов хэширования, которые наиболее распространены за рубежом. Например, алгоритм MD5 применяется в последних версиях Microsoft Windows для преобразования пароля пользователя в 16-байтное число.

SHA-1 (Secure Hash Algorithm) – алгоритм вычисления 20-байтного хэшзначения входных данных. Он также очень широко распространен в мире, преимущественно в сетевых протоколах защиты информации.

Помимо средства для создания ЭЦП, хэш-функции успешно используются для аутентификации пользователей. Существует немало криптографических протоколов аутентификации, основанных на применении хэш-функций.

#### Хэширование DSA

Secure Hash Algorithm специфицирован NIST и NSA для использования в DSA. По сообщению длиной меньше, чем 264 битов строится 160-битное значение (message digest). Затем digest подают на вход DSA, который считает подпись сообщения. На проверяющей стороне по сообщению должны получить тот же digest. SHA надежен, так как вычислительно невозможно найти сообщение по его дайджесту, либо найти два различных сообщения с одинаковым дайджестом. Любое изменение сообщения с очень высокой вероятностью приведет к другому дайджесту (тогда подпись не будет принята).

#### Хэширование по ГОСТ Р 34.11-94

Чтобы обеспечить невозможность подбора сообщений с одинаковым хэшзначением, отечественный стандарт хэширования ГОСТ Р 34.11-94 шифрует исходные данные, используя другой отечественный криптостандарт ГОСТ 28147-89. Процедура вычисления хэш-значения содержит несколько шаговэтапов. На начальном этапе (шаг 1) инициализируется регистр хэш-значения. Если длина сообщения не превышает 256 бит, осуществляется переход к шагу 3, в противном случае – к шагу 2.

На шаге 2 выполняется итеративное вычисление хэш-значения блоков хэшируемых данных по 256 бит с использованием хранящегося в регистре хэшзначения предыдущего блока. Набор операций включает генерацию ключей шифрования на основе блока хэшируемых данных, зашифрование хранящегося в регистре хэш-значения (в виде четырех блоков по 64 бит) по алгоритму ГОСТ 28147-89 в режиме простой замены и перемешивание результата. Процедура шага 2 повторяется до тех пор, пока длина необработанных входных данных не станет меньше или равной 256 бит, после чего происходит переход к шагу 3.

Шаг 3 включает несколько операций. Прежде всего при необходимости необработанная часть сообщения дополняется до 256 бит битовыми нулями. Затем вычисляется хэш-значение (см. шаг 2), которое и есть результат – хэш сообщения.

### 11.3. Алгоритмы цифровой подписи

Существует много различных алгоритмов электронно-цифровых подписей.

Например: ГОСТ Р 34.10 RSA

DSA (ECDSA)

EGDA

ECDSA ESIGN Схема Шнорра

Схема Диффи – Лампорта

Вероятностная схема подписи Рабина

СТБ 1176.2-99

Pohlig – Hellman McEliece

LUC и другие…

Рассмотрим основные из них: ГОСТ, RSA, DSA (ECDSA), EGDA.

Остальные алгоритмы используются гораздо реже или являются модификациями первых четырех вариантов.

#### Алгоритм цифровой подписи RSA

RSA – первый алгоритм цифровой подписи, который был разработан в 1977 году в Массачусетском технологическом институте и назван по первым буквам фамилий ее разработчиков (Ronald Rivest, Adi Shamir и Leonard

Adleman). RSA основывается на сложности *разложения большого числа n на простые множители*.

Алгоритм RSA представляет собой следующую последовательность действий:

1. Берутся два очень больших простых числа P и Q и находятся произведение простых чисел N=P×Q и функция Эйлера от этого произведения φ(N)=(P– 1)×(Q–1).
2. Выбирается случайное целое число E, взаимно простое с φ(N), и вычисляется

D=(1 MOD φ(N))/E.

1. Потом E и N публикуются как открытый ключ, D сохраняется в тайне.
2. Если S – сообщение, длина которого, определяемая по значению выражаемого им целого числа, должна быть в интервале (1,N), то оно превращается в шифровку возведением в степень D по модулю N и отправляется получателю S’=SD MOD N.
3. Получатель сообщения расшифровывает его, возведя в степень E (число E ему уже известно) по модулю N, так как

S=(S’E MOD N)=S(DE) MOD N.

По современным оценкам сложность задачи разложения на простые множители при целых числах n из 64 байт составляет порядка 1017–1018 операций, то есть находится на грани досягаемости для технических возможностей современного взломщика. Поэтому обычно в системах цифровой подписи на основе алгоритма RSA применяют более длинные целые числа n (обычно от 75 до 128 байт). Это приводит к увеличению длины цифровой подписи относительно 64байтного варианта примерно на 20–100% (в данном случае ее длина совпадает с длиной записи числа n), а также от 70% до 800% увеличивает время вычислений при подписании ЭЦП и проверке.

Кроме того, при генерации и вычислении ключей в системе RSA необходимо проверять большое количество довольно сложных дополнительных условий на простые числа p и q (что сделать достаточно трудно и чего обычно не делают, пренебрегая вероятностью неблагоприятного исхода – возможной подделки цифровых подписей). Невыполнение любого из условий может сделать возможным фальсификацию подписи со стороны того, кто обнаружит невыполнение хотя бы одного из них (при подписании важных документов допускать, даже теоретически, такую возможность нежелательно).

В дополнение ко всем этим алгоритмическим слабостям метода RSA следует также иметь в виду, что он защищен патентом США, и поэтому любое его использование на территории США или стран, где законодательство допускает защиту алгоритма патентом, и где признаются американские патенты, требует приобретения лицензии.

Обобщенная схема формирования и проверки цифровой подписи RSА показана на рис. 11.3.

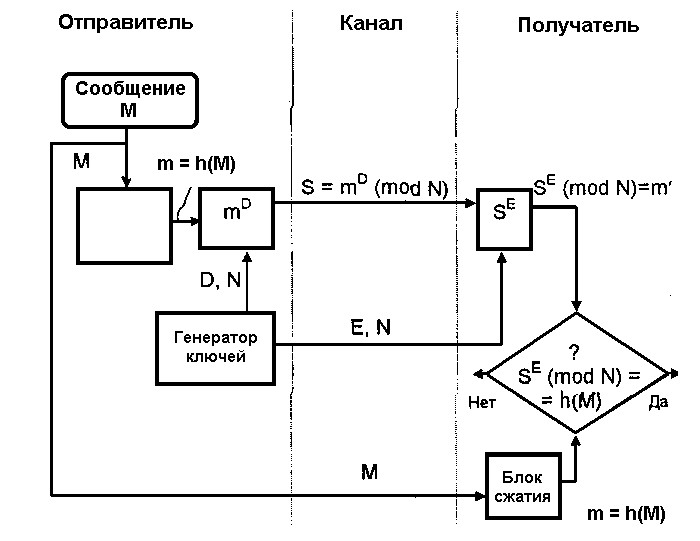
Допустим, что отправитель хочет подписать сообщение М перед его отправкой. Сначала сообщение М (блок информации, файл, таблица) сжимают с помощью хэш-функции h(М) в целое число m:

m = h(М).

Затем вычисляют цифровую подпись S под электронным документом М, используя хэш-значение m и секретный ключ D:

S = mD (mod N).

Пара (М, S) передается партнеру-получателю как электронный документ М, подписанный цифровой подписью S, причем подпись S сформирована обладателем секретного ключа D.



***Рис. 11.3.*** *Обобщенная схема цифровой подписи RSA*

После приема пары (М, S) получатель вычисляет хэш-значение сообщения М двумя разными способами. Прежде всего он восстанавливает хэш-значение m', применяя криптографическое преобразование подписи S с использованием открытого ключа Е:

m' = SE (mod N).

Кроме того, он находит результат хэширования принятого сообщения М с помощью такой же хэш-функции h(М):

m = h(М).

Если соблюдается равенство вычисленных значений, т.е.

SE (mod N) = h(М),

то получатель признает пару (М, S) подлинной. Доказано, что только обладатель секретного ключа D может сформировать цифровую подпись S по документу М, а определить секретное число D по открытому числу Е не легче, чем разложить модуль N на множители.

Кроме того, можно строго математически доказать, что результат проверки цифровой подписи S будет положительным только в том случае, если при вычислении S был использован секретный ключ D, соответствующий открытому ключу Е. Поэтому открытый ключ Е иногда называют «идентификатором» подписавшего.

***Недостатки алгоритма цифровой подписи RSA***

1. При вычислении модуля N, ключей Е и D для системы цифровой подписи RSА необходимо проверять большое количество дополнительных условий, что сделать практически трудно. Невыполнение любого из этих условий делает возможным фальсификацию цифровой подписи со стороны того, кто обнаружит такое невыполнение. При подписании важных документов нельзя допускать такую возможность даже теоретически.
2. Для обеспечения криптостойкости цифровой подписи RSA по отношению к попыткам фальсификации на уровне, например, национального стандарта США на шифрование информации (алгоритм DES), т.е. 1018, необходимо использовать при вычислениях N, D и Е целые числа не менее 2512 (или около 10154) каждое, что требует больших вычислительных затрат, превышающих на 20...30% вычислительные затраты других алгоритмов цифровой подписи при сохранении того же уровня криптостойкости.
3. Цифровая подпись RSA уязвима к так называемой мультипликативной атаке. Иначе говоря, алгоритм цифровой подписи RSA позволяет злоумышленнику без знания секретного ключа D сформировать подписи под теми документами, у которых результат хэширования можно вычислить как произведение результатов хэширования уже подписанных документов.

Рассмотрим три сообщения M1, M2 и M3, обладающих следующими свойствами:

m1 = h(M1), m2 = h(M2), m3 = h(M3) m3 = m1 \* m2 mod N

При этом допустим, что злоумышленник имеет реальные подписи (т.е. вычисленные законным владельцем ключа D) сообщений M1 и M2:

S1 = m1D mod N

S2 = m2D mod N

Тогда определить подпись сообщения M3 не составит труда: S3 = S1 \* S2 mod N, поскольку

S1 \* S2 mod N = mD \* m2D mod N = (m1 \* m2)D mod N = m3D mod N = S3.

Несмотря на кажущуюся простоту этой операции, имея всего лишь два подписанных сообщения, атаку осуществить достаточно сложно, поскольку необходимо еще и подобрать сообщение с нужным правильным хэшем. Однако с увеличением количества сообщений, подписанных владельцем ключа D, сложность подбора существенно снижается, причем нелинейно.

4*.* Существует и еще одна, не менее важная проблема, которую можно отнести к недостаткам применения данного метода формирования ЭЦП, – проблема нецелевого использования секретного ключа.

Предположим, среди знакомых некоего пользователя – владельца секретного ключа D есть злоумышленник. Под каким-либо предлогом этот мошенник просит пользователя расшифровать своим ключом D по алгоритму асимметричного шифрования RSA некое зашифрованное сообщение С. Доверчивый пользователь расшифровывает это сообщение с помощью формулы M = CD mod N и отсылает результат (расшифрованное сообщение) злоумышленнику.

На первый взгляд, опасности никакой нет. Однако при внимательном рассмотрении видно, что M – это по сути значение ЭЦП сообщения, хэш которого равен C. Следовательно, легальный пользователь поставил свою подпись под неким сообщением и отправил ее злоумышленнику.

В качестве меры противодействия такой атаке пользователям алгоритма RSA рекомендуется применять для собственно асимметричного шифрования и для ЭЦП разные пары ключей, и использовать все ключи только по назначению.

Как упоминалось выше, в файле, где хранится ключ, существуют дополнительные поля, содержимое которых несет в себе различную функциональность. Открытый ключ наиболее употребляемого в мире формата (X.509) обязательно сопровождается кодом назначения ключа, т.е. указывает, служит ли данный конкретный ключ ключом для асимметричного шифрования или для формирования ЭЦП.

5. Еще одним недостатком RSA является то, что RSA – очень медленный алгоритм. Для сравнения: программа, реализующая DES, по меньшей мере в 100 раз быстрее RSA, на аппаратном уровне – в 1000–10 000 раз, в зависимости от реализации.

#### Алгоритм цифровой подписи Эль-Гамаля (ЕGSА)

Более надежный и удобный для реализации на персональных компьютерах алгоритм цифровой подписи был разработан в 1984 г. американцем арабского происхождения Тахером Эль-Гамалем. В 1991 г. НИСТ США обосновал перед комиссией Конгресса США выбор алгоритма цифровой подписи Эль-Гамаля в качестве основы для национального стандарта.

Название ЕGSА происходит от слов ЕІ GаmаІ Signaturе Аlgorithm (алгоритм цифровой подписи Эль-Гамаля). Идея ЕGSА основана на том, что для обоснования практической невозможности фальсификации цифровой подписи может быть использована более сложная вычислительная задача, чем разложение на множители большого целого числа, – *задача дискретного логарифмирования*. Кроме того, Эль-Гамалю удалось избежать явной слабости алгоритма цифровой подписи RSA, связанной с возможностью подделки цифровой подписи под некоторыми сообщениями без определения секретного ключа.

В этом алгоритме целое число n полагается равным специально выбранному большому простому числу p, по модулю которого и производятся все вычисления. Такой выбор позволяет повысить стойкость подписи при ключах длиной 64 байта примерно в 1000 раз, то есть при такой длине ключей обеспечивается уровень стойкости порядка 1021. Правда, при этом длина самой цифровой подписи увеличивается в два раза и составляет 128 байт.

Рассмотрим подробнее алгоритм цифровой подписи Эль-Гамаля. Для того чтобы генерировать пару ключей (открытый ключ – секретный ключ), сначала выбирают некоторое большое простое целое число Р и большое целое число G, причем G < Р. Отправитель и получатель подписанного документа используют при вычислениях близкие большие целые числа Р (~10308 или ~21024) и G (~10154 или ~2512), которые не являются секретными.

1. Отправитель выбирает случайное целое число X, 1 < Х ≤ (Р-1), и вычисляет

Y =GX mod Р.

Число Y является открытым ключом, используемым для проверки подписи отправителя. Число Y открыто передается всем потенциальным получателям документов.

Число Х является секретным ключом отправителя для подписывания документов и должно храниться в секрете.

1. Для того чтобы подписать сообщение М, сначала отправитель хэширует его с помощью хэш-функции h(М) в целое число m:

m = h(М), 1<m<(Р–1),

и генерирует случайное целое число К, 1 < К < (Р–1), такое, что К и (Р–1) являются взаимно простыми. Затем отправитель вычисляет целое число а:

а = GK mod Р

и, применяя расширенный алгоритм Евклида, вычисляет с помощью секретного ключа Х целое число b из уравнения

m = Х \* а + К \* b (mod (Р-1)).

Пара чисел (а, b) образует цифровую подпись S:

S = (а, b),

проставляемую под документом М.

1. Тройка чисел (М, а, b) передается получателю, в то время как пара чисел (Х, К) держится в секрете.
2. После приема подписанного сообщения (М, а, b) получатель должен проверить, соответствует ли подпись S = (а, b) сообщению М.

Для этого получатель сначала вычисляет по принятому сообщению М число

m = h(М),

т.е. хэширует принятое сообщение М. Затем получатель вычисляет значение

А = Ya ab (mod Р)

и признает сообщение М подлинным, если, и только если

А = Gm (mod Р).

Иначе говоря, получатель проверяет справедливость соотношения

Ya ab (mod Р) = Gm (mod Р).

Можно строго математически доказать, что последнее равенство будет выполняться тогда, и только тогда, когда подпись S=(а, b) под документом М получена с помощью именно того секретного ключа X, из которого был получен открытый ключ Y. Таким образом, можно надежно удостовериться, что отправителем сообщения М был обладатель именно данного секретного ключа X, не раскрывая при этом сам ключ, и что отправитель подписал именно этот конкретный документ М.

Следует отметить, что выполнение каждой подписи по методу Эль-Гамаля требует нового значения К, причем это значение должно выбираться случайным образом. Если нарушитель раскроет когда-либо значение К, повторно используемое отправителем, то он сможет раскрыть секретный ключ Х отправителя. П р и м е р:

Выберем: числа Р=11, G=2 и секретный ключ Х = 8. Вычисляем значение открытого ключа:

Y = GX mod Р = Y = 28 mod 11=3.

Предположим, что исходное сообщение М характеризуется хэш-значением m=5.

Для того чтобы вычислить цифровую подпись для сообщения М, имеющего хэш-значение m = 5, сначала выберем случайное целое число К = 9. Убедимся, что числа К и (Р–1) являются взаимно простыми. Действительно, НОД (9,

10)=1. Далее вычисляем элементы а и b подписи:

а = GK mod Р = 29 mod 11 = 6, элемент b определяем, используя расширенный алгоритм Евклида: m = Х \* а + К \* b (mod(Р–1)).

При m = 5, а = 6, Х = 8, К = 9, Р = 11 получаем

5 = (6\* 8+9\* b)(mod 10) или

9\* b=-43(mod 10). Решение: b = 3. Цифровая подпись представляет собой пару: а = 6, b = 3.

Далее отправитель передает подписанное сообщение. Приняв подписанное сообщение и открытый ключ Y = 3, получатель вычисляет хэш-значение для сообщения М :

m = 5, а затем вычисляет два числа:

1) Yaab (mod Р) = 36 \* 63 (mod 11) =10 (mod 11); 2) Gm (mod Р) = 25 (mod 11) =10 (mod 11).

Так как эти два целых числа равны, принятое получателем сообщение признается подлинным.

Следует отметить, что схема Эль-Гамаля является характерным примером подхода, который допускает пересылку сообщения М в открытой форме вместе с присоединенным идентификатором (а, b). В таких случаях процедура установления подлинности принятого сообщения состоит в проверке соответствия идентификатора сообщению (аутентификации).

Схема цифровой подписи Эль-Гамаля имеет ряд преимуществ по сравнению со схемой цифровой подписи RSА:

1. При заданном уровне стойкости алгоритма цифровой подписи целые числа, участвующие в вычислениях, имеют запись на 25% короче, что уменьшает сложность вычислений почти в два раза и позволяет заметно сократить объем используемой памяти.
2. При выборе модуля Р достаточно проверить, что это число является простым и что у числа (Р–1) имеется большой простой множитель (т.е. всего два достаточно просто проверяемых условия).
3. Процедура формирования подписи по схеме Эль-Гамаля не позволяет вычислять цифровые подписи под новыми сообщениями без знания секретного ключа (как в RSА).

Однако алгоритм цифровой подписи Эль-Гамаля имеет и некоторые недостатки по сравнению со схемой подписи RSА. В частности, длина цифровой подписи получается в 1,5 раза больше, что, в свою очередь, увеличивает время ее вычисления.

Главная «заслуга» алгоритма Эль-Гамаля в том, что он послужил основой для принятия нескольких стандартов цифровой подписи, в том числе национального стандарта США DSS, введенного в действие 1 декабря 1994 года, и государственного стандарта РФ ГОСТ Р 34.10, введенного с 1 января 1995 года.

#### Алгоритм цифровой подписи DSA

Алгоритм DSA (Digital Signature Algorithm) был разработан в 1981 г. и с тех пор используется как стандарт США для электронной цифровой подписи – Digital Signature Standard (DSS). Алгоритм DSА является развитием алгоритмов цифровой подписи Эль-Гамаля и К.Шнорра. Согласно определению стандарта DSS, алгоритм DSA предусматривает применение в качестве хэш-функции алгоритма SHA. Заметим, что параметры алгоритма не засекречены.

Отправитель и получатель электронного документа используют при вычислении большие целые числа: G и Р – простые числа, L бит каждое (512 ≤ L ≤ 1024); q – простое число длиной 160 бит (делитель числа (Р–1)). Числа G, Р, q являются открытыми и могут быть общими для всех пользователей сети.

1. Отправитель выбирает случайное целое число X, 1 < Х < q. Число Х является секретным ключом отправителя для формирования электронной цифровой подписи.
2. Затем отправитель вычисляет значение

Y = G mod Р.

Число Y является открытым ключом для проверки подписи отправителя. Число Y передается всем получателям документов.

Этот алгоритм также предусматривает использование односторонней функции хэширования h(М). В стандарте DSS определен алгоритм безопасного хэширования SНА (Secure Hash Algorithm).

1. Для того чтобы подписать документ М, отправитель хэширует его в целое хэш-значение m:

m = h(М), 1<m<q,

затем генерирует случайное целое число К, 1< К< q, и вычисляет число r

* 1. = (GK mod Р) mod q.

1. Затем отправитель вычисляет с помощью секретного ключа Х целое число s:
   1. = ((m + r \* X)/K) mod q.

Пара чисел r и s образует цифровую подпись

S = (r,s)

под документом М.

Таким образом, подписанное сообщение представляет собой тройку чисел [М, r, s]. Получатель подписанного сообщения [М, r, s] проверяет выполнение условий

0 < r < q, 0 < s < q и отвергает подпись, если хотя бы одно из этих условий не выполнено.  **5.** Затем получатель вычисляет значение

w =(1/s) mod q,

хэш-значение m = h(М)

и числа u1 = (m \* w) mod q,

u2 = (r \* w) mod q.

**6.** Далее получатель с помощью открытого ключа Y вычисляет значение

v = ((Gu1 \* Yu2) mod Р) mod q

и проверяет выполнение условия

v = r.

Если условие v = r выполняется, тогда подпись

S = (r,s)

под документом М признается получателем подлинной.

Можно строго математически доказать, что последнее равенство будет выполняться тогда, и только тогда, когда подпись S = (r,s) под документом М получена с помощью именно того секретного ключа X, из которого был получен открытый ключ Y. Таким образом, можно надежно удостовериться, что отправитель сообщения владеет именно данным секретным ключом Х (не раскрывая при этом значения ключа X) и что отправитель подписал именно данный документ М.

Вычисление ЭЦП согласно алгоритму DSA и ее проверка приведены в табл. 11.1.

Существует строгое математическое доказательство, что условие v =r выполняется тогда и только тогда, если подпись (r, s) вычислена для сообщения M именно с помощью ключа D, на основании которого был определен использованный при проверке ключ E. Если же сообщение было изменено в процессе передачи, то значения m и u1 (которое зависит от m) будут неверными, и в итоге значения v и r не совпадут.

По сравнению с алгоритмом цифровой подписи Эль-Гамаля алгоритм DSА имеет следующие основные преимущества.

1. При любом допустимом уровне стойкости, т.е. при любой паре чисел G и Р (от 512 до 1024 бит), числа q, X, r, s имеют длину по 160 бит, сокращая длину подписи до 320 бит.
2. Большинство операций с числами К, r, s, Х при вычислении подписи производится по модулю числа q длиной 160 бит, что сокращает время вычисления подписи.
3. При проверке подписи большинство операций с числами u1, u2, v, w также производится по модулю числа q длиной 160 бит, что сокращает объем памяти и время вычисления.

Недостатком алгоритма DSА является то, что при подписывании и при проверке подписи приходится выполнять сложные операции деления по модулю q:

s = ((m + rX)/K) (mod q), w = (1/s) (mod q), что не позволяет получать максимальное быстродействие.

Следует отметить, что реальное исполнение алгоритма DSА может быть ускорено с помощью выполнения предварительных вычислений. Заметим, что значение r не зависит от сообщения М и его хэш-значения m. Можно заранее создать строку случайных значений К и затем для каждого из этих значений вычислить значения r. Можно также заранее вычислить обратные значения К-1 для каждого из значений К. Затем, при поступлении сообщения М, можно вычислить значение s для данных значений r и К-1. Эти предварительные вычисления значительно ускоряют работу алгоритма DSА.

По сравнению с описанным выше алгоритмом RSA у алгоритма DSA есть ряд достоинств. Во-первых, при любом достижимом уровне криптостойкости, который определяется размерностью чисел P и Q, числа q, D, r, s имеют фиксированную размерность (каждое по 160 бит), поэтому и ЭЦП присуща фиксированная размерность – 320 бит. Отсюда следует, что время вычислений для DSA существенно меньше, чем при использовании, например, RSA с тем же уровнем криптостойкости.

*Таблица 11.1*

**Процедуры вычисления ЭЦП как (r, s) и ее проверки**

**согласно алгоритму DSA**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Последовательность вычислений | | Последовательность проверок | | |
| Операция | Формла/ условие | Операция | Формула/ усло-  вие | Результат |
| Вычисление  хэша сообщения | m = h(M) | Проверка условий | 0 < r > q 0 < s < q> | Нет – любое =  ЭЦП неверн  а |
| Выбор случайного числа x | 1 < x < q | Вычисление промежуточных величин | m = h(M) w = 1/s \* mod q u1 = (m (w) mod q u2 = (r (w) mod q | Н/п |
| Определение  первой ча-  сти, r | r = (Gx mod P) mod q | Определение первой части ЭЦП | v = ((Gu1 \* Eu2) mod P) mod q | Н/п |
| Определение  второй ча-  сти, s\* | s = [(m + r \*  D)/x] \* mod q | Проверка ра-  венства | v = r | Нет –  ЭЦП неверн  а |
| Примечания: D – секретный ключ; E – открытый ключ. Н/п – неприменимо. \* Поскольку все вычисления выполняются в конечном поле, определяемом значением q, то s – целое число и вычисляется как s = (m + r \* Ks) \* (1/x) mod q, где (1/x) – целое, элемент конечного поля, обратный x, которое вычисляется по обобщенному алгоритму Евклида. | | | | |

Во-вторых, при проверке ЭЦП большинство операций с числами w, u1, u2, s также выполняются по модулю 160 бит, что, в свою очередь, сокращает объем требуемых ресурсов.

Но не бывает объекта без недостатков. Для DSA это наличие в нем ресурсоемких операций определения обратных величин по модулю q. Однако данный «минус» вполне компенсируется применением так называемых предвычислений (т.е. выполнением части вычислений заранее). В частности, можно предварительно сгенерировать массив случайных значений x и вычислить для всех x этого массива обратные величины 1/x. Кроме того, поскольку значение r не зависит от хэша подписываемого сообщения (т.е. от самого сообщения), удобно заранее вычислить и набор значений r как r = (Gx mod P) mod q.

Главной причиной для принятия в 1994 году в США национального стандарта цифровой подписи (DSS) на основе алгоритма Эль-Гамаля послужили эти преимущества, а также соображения, связанные с возможностью его реализации любым разработчиком свободно, не заключая коммерческих лицензионных соглашений с держателем патента компанией RSA Data Security. Свою роль сыграла и возможность свободного безлицензионного экспорта такой технологии из США: технологию цифровой подписи RSA можно «обернуть», поменяв местами процедуры подписывания и проверки, и получить готовый метод шифрования информации с открытыми ключами.

Технологии шифрования информации при экспорте из США требуют получения специального разрешения Госдепартамента, который согласовывает решения по вопросам экспорта криптографии с Агентством национальной безопасности (АНБ) – главной криптографической службой США, выступающей против экспорта криптографических технологий.

Несмотря на перечисленные обстоятельств, RSA является наиболее известным алгоритмом цифровой подписи и шифрования с открытым ключом. Он широко распространен, практически опробован во многих странах и признан в качестве стандарта *de facto* большинством разработчиков операционных систем, сетевых технологий и прикладного программного обеспечения. Его популярность объясняется, во-первых, тем, что он появился на восемь лет раньше, во-вторых, значительно более широкой известностью в научных кругах как самого алгоритма, так и его авторов, и, наконец, успешным бизнесом держателя патента – компании RSA Data Security (Эль-Гамаль был в 1994–1995 гг. ее сотрудником).

Технические преимущества алгоритма DSA в настоящее время уже не является лучшими, но вероятность его принятия в качестве международного стандарта остается довольно большой.

#### Алгоритм цифровой подписи на основе эллиптических кривых ECDSA

ECDSA (Elliptic Curve Digital Signature Algorithm) – алгоритм с открытым ключом для создания цифровой подписи, аналогичный по своему строению DSA, но определенный в отличие от него не над полем целых чисел, а над полем точек эллиптической кривой.

Создание ключей:

1. Выбирается эллиптическая кривая Ep (a,b). Число точек на ней должно делиться на большое целое n.
2. Выбирается точка Р Ep (a,b).
3. Выбирается случайное число d [1, n-1].
4. Вычисляется Q = d × P.
5. Закрытым ключом является d, открытым ключом – (E, P, n, Q). Создание подписи:
6. Выбирается случайное число k [1, n-1].
7. Вычисляется
8. k × P = (x1, y1) и
9. r = x1 (mod n).

Проверяется, чтобы r не было равно нулю, так как в этом случае подпись не будет зависеть от закрытого ключа. Если r = 0, то выбирается другое случайное число k.

1. Вычисляется k-1 mod n.
2. Вычисляется

s = k-1 (Н(M) + dr) (mod n).

Проверяется, чтобы s не было равно нулю, так как в этом случае необходимого для проверки подписи числа s-1 mod n не существует. Если s = 0, то выбирается другое случайное число k.

Подписью для сообщения М является пара чисел (r,s). Проверка подписи:

1. Проверить, что целые числа r и s принадлежат диапазону чисел [0, n-1]. В противном случае результат проверки отрицательный, и подпись отвергается.
2. Вычислить w = s-1 (mod n) и H(M).
3. Вычислить u1 = H(M) w (mod n); u2 = rw (mod n).
4. Вычислить

u1P + u2Q = (x0, y0); v = x0 (mod n).

1. Подпись верна в том и только том случае, когда v = r.

#### ГОСТ Р 34.10-94

Отечественный стандарт цифровой подписи обозначается как ГОСТ Р 34.11. Стандарт на электронную подпись ГОСТ Р 34.10 был опубликован Госстандартом РФ в мае 1994 года и вступил в действие с 1 января 1995 года. В предварительном варианте он был введен как ведомственный стандарт на цифровую подпись ЦБ РФ и использовался в этом качестве с сентября 1993 года по декабрь 1994 года.

193

Алгоритмы вычисления и проверки подписи в ГОСТ Р 34.10 устроены аналогично алгоритму DSA.

В нем используются следующие параметры: р – большое простое число длиной от 509 до 512 бит либо от 1020 до 1024

бит; q – простой сомножитель числа (р-1), имеющий длину 254...256 бит; а – любое число, меньшее (р-1), причем такое, что аq mod p=1; х – некоторое число, меньшее q; у = аx mod р.

Кроме того, этот алгоритм использует однонаправленную хэш-функцию Н(х). Стандарт ГОСТ Р 34.11-94 определяет хэш-функцию, основанную на использовании стандартного симметричного алгоритма ГОСТ 28147-89.

Первые три параметра р, q и а являются открытыми и могут быть общими для всех пользователей сети. Число х является секретным ключом. Число у является открытым ключом. Чтобы подписать некоторое сообщение m, а затем проверить подпись, выполняются следующие шаги.

1. Пользователь А генерирует случайное число k, причем k<q.
2. Пользователь А вычисляет значения r = (аk mod p) mod q, s = (х \* r + k (Н(m))) mod p.

Если Н(m) mod q=0, то значение Н(m) mod q принимают равным единице.

Если r=0, то выбирают другое значение k и начинают снова. Цифровая подпись представляет собой два числа: r mod 2256 и s mod 2256.

Пользователь А отправляет эти числа пользователю В.

1. Пользователь В проверяет полученную подпись, вычисляя

v = Н(m)q-2 mod q, z1 = (s \* v) mod q,

z2 = ((q-r) \* v) mod q, u = ((аz1 \* уz2) mod р) mod q.

Если u = r, то подпись считается верной.

Различие между этим алгоритмом и алгоритмом DSА заключается в том, что в DSА

s = (k-1 (х \* r + (Н(m)))) mod q,

что приводит к другому уравнению верификации.

Следует также отметить, что в отечественном стандарте ЭЦП параметр q имеет длину 256 бит. Западных криптографов вполне устраивает q длиной примерно 160 бит. Различие в значениях параметра q является отражением стремления разработчиков отечественного стандарта к получению более безопасной подписи.

*Таблица 11.2*

**Процедуры вычисления ЭЦП как (r, s) и ee проверки по алгоритму ГОСТ Р 34.10– 94**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Последовательность вычислений | | Последовательность проверок | | |
| Операция | Формула/условие | Операция | Формула/условие | Результат |
| Вычисление хэша сообще-  ния | m = h(M) | Проверка условий | 0 < r < q 0 < s < q | Нет –  любое = ЭЦП нев  ерна |
| Проверка условия\* | m [не равно] 0 mod q |  |  |  |
| Выбор случайного числа x | 0 < x < q | Вычисление промежуточных величин | v = mq-2 mod q z1 = (s (v) mod q z2 = ((q – r) (v) mod q | Н/п |
| Определение первой части,  r\*\* | r = (ax mod p) mod q | Определение первой части ЭЦП | u = (az1 \* yz2) mod p) mod q | Н/п |
| Определение второй части, s | s = (Ks (r + x (m) mod q | Проверка равенства | u = r | Нет – ЭЦП нев  ерна |
| Примечания: Ks – секретный ключ. Н/п – неприменимо. \* При m = 0 mod q значение m принимается равным 1; \*\* при r = 0 – возврат к выбору другого случайного числа x. | | | | |

Принципы вычислений по данному алгоритму аналогичны применяемым в предыдущем стандарте ГОСТ Р 34.10–94. Сначала генерируется случайное число x, с его помощью вычисляется r-часть ЭЦП, затем s-часть ЭЦП вычисляется из r-части, значения x, значения секретного ключа и хэш-значения подписываемых данных. При проверке же подписи аналогичным образом проверяется соответствие определенным соотношениям r, s, открытого ключа и хэш-значения информации, подпись которой проверяется. Подпись считается неверной, если соотношения неверны. Существует ограничение: при реализации ЭЦП по стандарту ГОСТ Р 34.10-94 разрешено использовать только 1024-битные значения параметра p.

#### ГОСТ Р 34.10-2001

ГОСТ Р 34.10-2001 (полное название: «*ГОСТ Р 34.10-2001. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи*») – российский стандарт, описывающий алгоритмы формирования и проверки электронной цифровой подписи. Принят и введен в действие Постановлением Госстандарта России от 12 сентября 2001 года вместо ГОСТ Р 34.10-94.

ГОСТ Р 34.10-2001 основан на эллиптических кривых. Его стойкость определяется сложностью взятия дискретного логарифма в группе точек эллиптической кривой, а также на стойкости хэш-функции по ГОСТ Р 34.11.

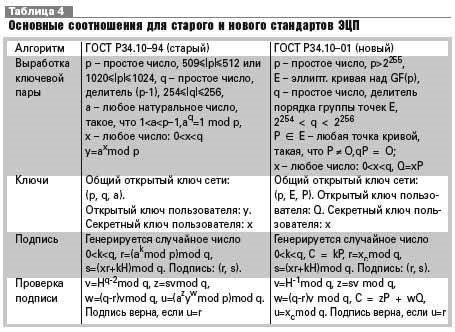
К сообщению *М* дописывается цифровая подпись размером 512 бит и текстовое поле. В текстовом поле могут содержаться, например, дата и время отправки или различные данные об отправителе:

|  |  |  |  |
| --- | --- | --- | --- |
| Сообщение М | + | Цифровая подпись | Текст |

Дополнение

|  |  |
| --- | --- |
| Данный алгоритм не описывает механизм генерации параметров, необходимых для формирования подписи, а только определяет, каким образом на основании таких параметров получить цифровую подпись. Механизм генерации параметров определяется на месте в зависимости от разработанной системы.    ***Параметры схемы цифровой подписи:*** | |
| ентами ми *a*  Также  раз). А 0  **1.**  **2.** | простое число *p* – модуль эллиптической кривой такой, что *p* > 2255; |
| эллиптическая кривая *E* задается своим инвариантом *J*(*E*) или коэффици-  , где *Fp* – конечное простое поле. *J*(*E*) связан с коэффициента-  и *b* следующим образом:    ,    при этом ;  целое число *m* – порядок группы точек эллиптической кривой (то есть  число элементов группы) – *m* не должно совпадать с *p*; простое число *q*, порядок одной из циклических подгрупп группы точек эллиптической кривой, то есть выполняется *m* = *nq* для некоторого . *q* лежит в пределах 2254 < *q* < 2256;  точка  эллиптической кривой *E*, являющаяся генератором  подгруппы порядка *q*, или, другими словами, *qP* = 0. Здесь *qP* = *P* + *P* + ...(*q*  – нулевой элемент группы точек эллиптической кривой; *h*(*M*) – хэш-функция (ГОСТ Р 34.11-94), которая конечные сообщения *M*  отображает в двоичные вектора длины 256 бит.  Каждый пользователь цифровой подписи имеет личные ключи:  ключ шифрования *d* – целое число, лежащее в пределах 0 < *d* < *q*. ключ расшифрования – точка эллиптической кривой, *dP* = *Q*. |
|

Алгоритм ГОСТ Р 34.10-2001 аналогичен алгоритму ГОСТ Р 34.10-94 (табл. 11.3). Различия старого и нового стандартов ЭЦП РФ заключаются в том, что в старом стандарте некоторые операции выполнялись в конечном поле GF(p), а в новом – в группе точек эллиптической кривой над полем GF(p).



*Таблица 11.3.*

**Сравнение**

**старого и нового стандартов ЭЦП**

**РФ**

### 11.4. Криптоанализ односторонних хэш-функций

#### Лобовая атака

Имея *h*(*M*), злоумышленник должен найти *M′* такое, что *h*(*M*)=*h*(*M′*). Тогда можно заявить, что подписано *M′*. Если хэш-функция дает *m*-битную строку, то этот метод требует 2*m* случайных сообщений.

#### Парадокс дней рождения

Парадокс дней рождения – известная статистическая задача. Сколько человек должно находиться в комнате, чтобы с вероятностью, большей ½, у кого-то был одинаковый с вами день рождения? Ответ: 183. Теперь, сколько человек должно находиться в комнате, чтобы с вероятностью, большей ½, хоть у когото из них совпали дни рождения? Ответ удивительно мал: 23, так как 23 человека ~ 253 пары.

Нахождение кого-то с определенным днем рождения аналогично лобовой атаке. Нахождение двух людей с одинаковым случайным днем рождения аналогично данной атаке.

Предположим, что наилучший способ атаки односторонней хэш-функции – грубая сила. Односторонняя хэш-функция производит значение длиной *m*, тогда для данного метода потребуется 2*m/2* значений. Если машина хэширует 1 млн. сообщений в секунду, то потребуется 600,000 лет для нахождения второго сообщения с таким же 64-битным хэшем. Та же машина может найти пару сообщений (случайных) с одинаковым хэшем за час.

#### Алгоритм Юваля

Вход: исходное сообщение *x*1, подделываемое сообщение *x*2, хэш-функция *h* с *m*-битным выходом.

Выход: *x*1′, *x*2′, являющиеся небольшими изменениями *x*1, *x*2, причем *h*(*x*1′)=*h*(*x*2′).

Сгенерировать *t=*2*m*/2 небольших изменений *x*1.

Захэшировать каждое изменение и сохранить хэш-значения с соответствующими сообщениями.

Генерировать небольшие изменения *x*2, вычислять для них хэш-значения и сравнивать с хэш-значениями из п.2, продолжать, пока не найдется совпадение. Совпадение ожидается после *t* просмотренных изменений *x*2.

Алиса готовит две версии контракта – один выгодный для Боба, другой – разоряющий его. Алиса делает несколько малозначительных изменений к каждому документу и считает хэш-значения каждый раз (например, заменяет SPACE на SPACE-BACKSPACE-SPACE, добавляет один или два пробела перед переводом строки и т.д.) Только путем одного «изменения/оставления как было» на каждой строке Алиса может сгенерировать 232 документов. Алиса сравнивает набор хэшей для обоих документов, подыскивая одинаковые пары (если хэш-функция 64-битная, то обычно хватит 232 пар). Выбирается та пара, что дает одинаковые хэши. Алиса подсовывает Бобу выгодный контракт. Боб подписывает его хэш. Алиса теперь может доказать, что Боб подписал невыгодный для него контракт.

#### Вопросы и задания для самостоятельной работы

1. Что общего между обычной и цифровой подписью? Чем они различаются?
2. Какие задачи позволяет решить цифровая подпись?
3. В чем заключается принципиальная сложность в практическом применении систем цифровой подписи?
4. Почему в криптографических системах, основанных на открытых ключах, нельзя использовать одинаковые ключи для шифрования и цифровой подписи?

## ТЕМА 12. КВАНТОВАЯ КРИПТОГРАФИЯ

*Квантовая криптография* (quantum cryptography) – раздел криптографии*,* посвященный применению методов квантовой физики для синтеза и анализа криптографических систем.

Методы защиты информации в квантовой криптографии основаны на определенных явлениях квантовой физики. В отличие от традиционной криптографии, которая использует математические методы, чтобы обеспечить секретность информации, квантовая криптография сосредоточена на физике информации, так как рассматривает случаи, когда информация переносится с помощью объектов квантовой механики. Процесс отправки и приема информации всегда выполняется физическими средствами, например при помощи электронов в электрическом токе или фотонов в линиях волоконно-оптической связи. А подслушивание может рассматриваться как измерение физических объектов

– переносчиков информации.

### 12.1. Природа секретности квантового канала связи

Состояние квантового объекта (то есть объекта очень малой массы и размеров, например электрона или фотона) может быть определено измерением. Однако сразу после выполнения этого измерения квантовый объект неизбежно переходит в другое состояние, причем предсказать это состояние невозможно. Следовательно, если в качестве носителей информации использовать квантовые частицы, то попытка перехватить сообщение приведет к изменению состояния частиц, что позволит обнаружить нарушение секретности передачи. Кроме того, невозможно получить полную информацию о квантовом объекте и, следовательно, невозможно его скопировать. Эти свойства квантовых объектов лежат в основе квантовой криптографии.

Носителями информации в квантовых протоколах являются фотоны, поляризованные под углами 0, 45, 90, 135 градусов. В соответствии с законами квантовой физики, с помощью измерения можно различить лишь два ортогональных состояния: если известно, что фотон поляризован либо вертикально, либо горизонтально, то путем измерения можно установить, как именно; то же самое можно утверждать относительно поляризации под углами 45 и 135 градусов. Однако с достоверностью отличить вертикально поляризованный фотон от фотона, поляризованного под углом 45 градусов, невозможно.

Квантовая криптография использует этот факт для обеспечения возможности двум сторонам, которые ранее не встречались и не обменивались никакой предварительной секретной информацией, осуществлять между собой связь в обстановке полной секретности без боязни быть подслушанными злоумышленником.

Таким образом, технология квантовой криптографии опирается на принципиальную неопределенность поведения квантовой системы – невозможно одновременно получить координаты и импульс частицы, невозможно измерить один параметр фотона, не исказив другой. Это фундаментальное свойство природы в физике известно как *принцип неопределенности Гейзенберга*, сформулированный в 1927 г.

Используя квантовые явления, можно спроектировать и создать такую систему связи, которая всегда может обнаруживать подслушивание. Это обеспечивается тем, что попытка измерения взаимосвязанных параметров в квантовой системе вносит в нее нарушения, разрушая исходные сигналы, а значит, по уровню шума в канале легитимные пользователи могут распознать степень активности перехватчика.

### 12.2. Основные направления развития квантовой криптографии

В квантовой криптографии выделились два основных направления развития систем распределения ключей.

Первое направление основано на кодировании квантового состояния одиночной частицы и базируется на принципе *невозможности различить абсолютно надежно два неортогональных квантовых состояния*.

Защищенность этого направления основывается на теореме о запрете клонирования неизвестного квантового состояния. Невозможно создать точную копию неизвестного квантового состояния без воздействия на исходное состояние. Пусть, например, отправитель и получатель используют для передачи информации двухуровневые квантовые системы, кодируя состояния этих систем. Если злоумышленник перехватывает носитель информации, посланный отправителем, измеряет его состояние и пересылает далее получателю, то состояние этого носителя будет иным, чем до измерения. Таким образом, подслушивание квантового канала приводит к ошибкам передачи, которые могут быть обнаружены легальными пользователями.

Основным протоколом квантовой криптографии на одночастичных состояниях является протокол BB84.

Второе направление развития основано на эффекте *квантового перепутывания (принцип парадокса Эйнштейна–Подольского–Розена)*.Две квантовомеханические системы (в том числе и разделенные пространственно) могут находиться в состоянии корреляции, так что измерение выбранной величины, осуществляемое над одной из систем, определит результат измерения этой величины на другой. Ни одна из запутанных систем не находится в определенном состоянии. Поэтому запутанное состояние не может быть записано как прямое произведение состояний систем. Состояние двух частиц со спином 1/2 может служить примером запутанного состояния.

Измерение, проведенное на одной из двух подсистем, дает с равной вероятностью состояния «0» или «1». Состояние же другой подсистемы будет противоположным, т.е. «0», если результат измерения на первой системе был «1», и наоборот.

Основным протоколом квантового распределения ключей, построенным на эффекте квантового запутывания, является протокол, предложенный А. Экертом в 1991 году и названный в его честь E91 (Ekert). Второе его название – EPR (Einstein–Podolsky–Rosen).

Базовые принципы этих двух направлений легли в основу разработки всех протоколов квантового распределения ключей.

### 12.3. Протоколы квантового обмена информацией

Существует множество протоколов квантовой криптографии основанных на передаче информации посредством кодирования в состояниях одиночных фотонов, например: BB84, B92, ВВ84(4+2), с шестью состояниями, Гольденберга–Вайдмана, Коаши–Имото и их модификации. Единственным протоколом, разработанным для кодирования информации в спутанных состояниях, на момент написания книги является протокол E91.

Проведя сравнительный анализ приведенных выше протоколов, из расчета количества принятых фотонов, можно судить о том, что наиболее эффективным является ВВ84. Более поздние его модификации направлены на уменьшение процента ошибок и количества полезной информации, которую теоретически может получить злоумышленник. Альтернативой в развитии протокола ВВ84 является протокол В92. Преимуществом протокола В92 перед ВВ84 является использование фотонов с двумя типами поляризации (вместо четырех), что позволяет упростить схему реализации, однако обеспечивает меньшую эффективность (уменьшается количество принятых фотонов), и гарантированную секретность ключа только на расстоянии до 20 км, тогда как ВВ84 – на расстоянии до 50 км. В настоящее время в коммерческих системах распределения ключа применяется протокол ВВ84.

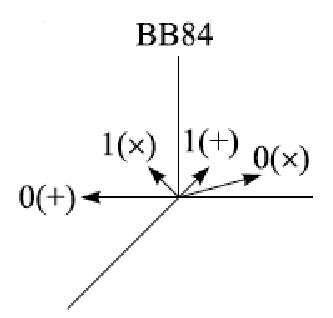
Рассмотрим более подробно основные протоколы квантового распределения ключей.

#### Квантовый протокол ВВ84

Первый протокол квантовой криптографии, как уже говорилось в теме 1, был предложен и опубликован в 1984 году Беннетом и Брассардом.

В протоколе BB84 используются 4 квантовых состояния фотонов, например, направление вектора поляризации, одно из которых отправитель выбирает в зависимости от передаваемого бита: 90° или 135° для «1», 45° или 0° для «0». Одна пара квантовых состояний соответствует «0» и «1» и принадлежит базису «+». Другая пара квантовых состояний соответствует «0» и «1» и принадлежит базису «х». Внутри обоих базисов состояния ортогональны, но состояния из разных базисов являются попарно неортогональными (неортогональность необходима для детектирования попыток съема информации).

Базисы повернуты друг относительно друга на 45° (рис. 12.1).



***Рис. 12.1****. Состояния поляризации фотонов, используемые в протоколе*

Схема квантового распределения ключей шифрования ВВ84 работает следующим образом.

1. Сначала отправитель генерирует и посылает получателю последовательность фотонов, поляризация которых выбрана случайным образом и может составлять 0, 45, 90 и 135°.
2. Получатель принимает эти фотоны и для каждого из них случайным образом решает, замерять его поляризацию как перпендикулярную или диагональную.
3. Затем по открытому каналу получатель объявляет для каждого фотона, какой тип измерений им был сделан (перпендикулярный или диагональный), но не сообщает результат этих измерений, например 0, 45, 90 или 135°.
4. По этому же открытому каналу отправитель сообщает ему, правильный ли вид измерений был выбран для каждого фотона.
5. Затем отправитель и получатель отбрасывают все случаи, когда получатель сделал неправильные замеры. Если квантовый канал не перехватывался, оставшиеся виды поляризации и будут поделенной между получателем и отправителем секретной информацией, или ключом. Этот этап работы квантовокриптографической системы называется *первичной квантовой передачей*.

На рис. 12.2 рассматривается простой пример создания общего секретного ключа в квантово-криптографической системе.

Следующим важным этапом является *оценка попыток перехвата* информации в квантово-криптографическом канале связи. Это может производиться отправителем и получателем по открытому каналу путем сравнения и отбрасывания случайно выбранных ими подмножеств полученных данных.

Если такое сравнение выявит наличие перехвата, отправитель и получатель отбрасывают все свои данные и начинают *повторное выполнение первичной квантовой передачи*.

В противном случае они оставляют прежнюю поляризацию, принимая фотоны с горизонтальной или 45°-й поляризацией за двоичный «0», а с вертикальной или 135°-й поляризацией – за двоичную «1».

*Отправитель посылает фотоны, имеющие одну из четырех возможных поляризаций, которую она выбирает случайным образом.*



*Для каждого фотона Получатель выбирает случайным образом тип измерения: он изменяет либо прямолинейную поляризацию (+), либо диагональную (*х*).*



*Получатель записывает результаты изменения и сохраняет в тайне.*



*Получатель открыто объявляет, какого типа измерения он проводил, а Отправитель сообщает ему, какие измерения были правильными.*



*Отправитель и Получатель сохраняют все данные, полученные в тех случаях, когда Получатель применял правильное измерение. Эти данные затем переводятся в биты (0 и 1), последовательность которых и является результатом первичной квантовой передачи.*



***Рис. 12.2.*** *Принципы первичной квантовой передачи*

Более эффективной проверкой для отправителя и получателя является проверка на четность, осуществляемая по открытому каналу. Например, Отправитель может сообщить: «Просмотрен 1-й, 4-й, 4-й, 8-й... и 998-й из моих 1000 бит, и они содержат четное число единиц». Тогда получатель подсчитывает число «1» на тех же самых позициях. Можно показать, что, если данные у получателя и отправителя отличаются, проверка на четность случайного подмножества этих данных выявит количество ошибок. Достаточно повторить такой тест 20 раз с 20 различными случайными подмножествами, чтобы вычислить процент ошибок. Если ошибок слишком много, то считается, что производился перехват в квантово-криптографической системе.

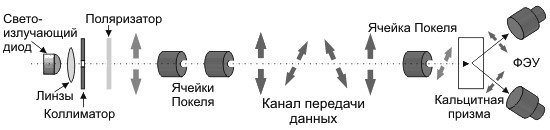
Пример определения количества фотонов, принятых получателем, показан в таблице 12.1.

*Таблица 12.1* **Формирование квантового ключа по протоколу ВB84**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Передаваемый бит | 0 | 1 | 0 | 1 |
| Поляризация передачи | H | V | -450 | 450 |
| Поляризация приема | V/H | V/H | V/H | V/H |
| Двоичный сигнал приема | 0 | 1 | ? | ? |

Реализация протокола осложняется присутствием шума, который может вызвать ошибки. Вносимые ошибки могут быть обнаружены и устранены с помощью подсчета четности, при этом один бит из каждого блока отбрасывается.

Схема реализации однонаправленного канала с квантовым шифрованием показана на рис. 12.3. Передающая сторона находится слева, а принимающая – справа. Ячейки Покеля служат для импульсной вариации поляризации потока квантов передатчиком и для анализа импульсов поляризации приемником. Передатчик может формировать одно из четырех состояний поляризации (0, 45, 90 и 135 градусов).



***Рис. 12.3.*** *Практическая схема реализации идеи квантовой криптографии (с ячейками Покеля)*

Собственно передаваемые данные поступают в виде управляющих сигналов на эти ячейки. В качестве канала передачи данных может использоваться оптическое волокно. В качестве источника света может использоваться светоизлучающий диод или лазер. Свет фильтруется, поляризуется и формируется в виде коротких импульсов малой интенсивности.

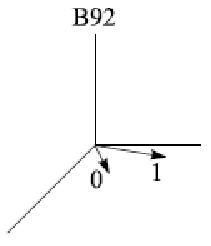
На принимающей стороне после ячейки Покеля ставится кальцитовая призма, которая расщепляет пучок на два фотодетектора (ФЭУ), измеряющих две ортогональные составляющие поляризации.

При формировании передаваемых импульсов (квантов) приходится решать проблему их интенсивности. Если квантов в импульсе 1000, есть вероятность того, что 100 квантов по пути будет отведено злоумышленником на свой приемник. Анализируя позднее открытые переговоры между передающей и принимающей стороной, он может получить нужную ему информацию. В идеале число квантов в импульсе должно быть около одного. Здесь любая попытка отвода части квантов злоумышленником приведет к существенному росту числа ошибок у принимающей стороны. В этом случае принятые данные должны быть отброшены и попытка передачи повторена.

Но делая канал более устойчивым к перехвату, сталкиваемся с проблемой «темнового» шума (выдача сигнала в отсутствии фотонов на входе) приемника (ведь мы вынуждены повышать его чувствительность). Для того чтобы обеспечить надежную транспортировку данных, логическому нулю и единице могут соответствовать определенные последовательности состояний, допускающие коррекцию одинарных и даже кратных ошибок.

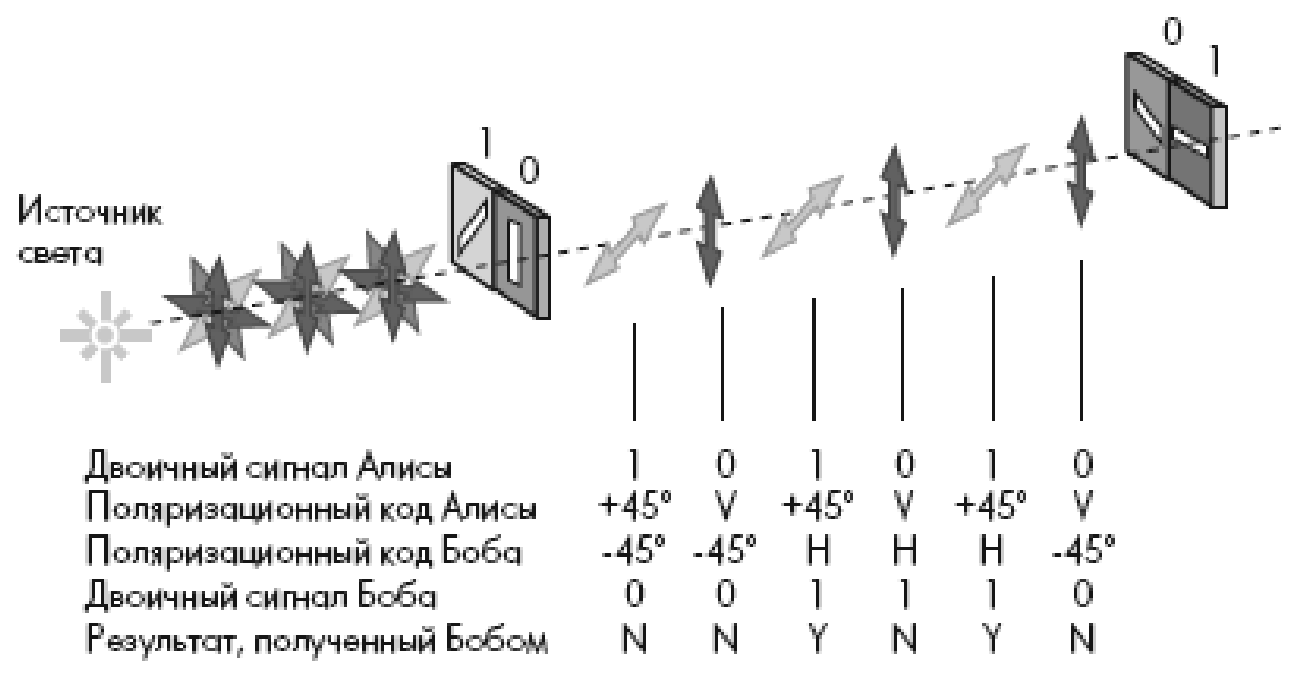
#### Квантовый протокол В92

В протоколе используются фотоны, поляризованные в двух различных направлениях для представления нулей и единиц. Фотоны, поляризованные вдоль направления +450, несут информацию о единичном бите, фотоны, поляризованные вдоль направления 0о(V) – о нулевом бите. Эти состояния удобно для наглядности изображать графически (рис. 12.4).



***Рис. 12.4.*** *Поляризационные состояния, используемые в протоколе В92*

Алгоритм работы протокола В92 представлен на рис. 12.5. Отправитель посылает фотоны, поляризованные в направлениях *0* и *+450*, представляющие нули и единицы. Причем последовательность фотонов, посылаемая отправителем, случайно ориентирована. Получатель принимает фотоны через фильтры, ориентированные под углом *900* и *1350(-450).* При этом если переданный фотон будет анализирован получателем при помощи фильтра, ориентированного под углом *900* по отношению к передаваемому фотону, то фотон не пройдет через фильтр. Если же этот угол составит *450*, то фотон пройдет через фильтр с вероятностью *0,5*.



***Рис. 12.5.*** *Формирование квантового ключа по протоколу В92*

Для определения поляризации получатель анализирует принимаемые фотоны, используя выбранный случайным образом один из двух неортогональных базисов «+» или «-». Если получатель анализирует посланный фотон фильтром с ортогональным направлением поляризации, то он не может точно определить, какое значение данный фотон представляет: *1*, соответствующее фотону, который не проходит, или *0*, соответствующее фотону, который не проходит с вероятностью *0,5*. Если же направления поляризации между посланным фотоном и фильтром, неортогональны, то получатель может определить, что принят фотон, соответствующий *0*. Если фотон был принят удачно, то очередной бит ключа кодируется *0* (если фотон был принят фильтром, ориентированным под углом *1350*), либо *1* (если фотон был принят фильтром, ориентированным по направлению *H*) (таблица 12.2).

*Таблица 12.2* **Формирование квантового ключа по протоколу В92**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Передаваемый бит | 1 | 0 | 1 | 0 |
| Поляризация передачи | 450 | V | 450 | V |
| Поляризация приема | -450 | -450 | H | H |
| Двоичный сигнал приема | 0 | 0 | 1 | 1 |
| Результат приема | - | - | + | - |

В первой и четвертой колонке поляризации при передаче и приеме ортогональны, и результат детектирования будет отсутствовать. В колонках 2 и 3 коды двоичных разрядов совпадают, и поляризации не ортогональны. По этой причине с вероятностью *50%* может быть положительный результат в любом из этих случаев (и даже в обоих). В таблице предполагается, что успешное детектирование фотона происходит для случая, представленного в колонке 3. Именно этот бит становится первым битом общего секретного ключа передатчика и приемника. Отсюда минимальное количество фотонов, которое может быть принято. То есть в результате передачи такого ключа около *25%* фотонов будут правильно детектированы.

После этого по открытому каналу связи получатель может передать отправителю, какие *25* фотонов из каждых *100* были получены. Данная информация и будет служить ключом к новому сообщению. При этом чтобы злоумышленник не узнал информацию о ключе, по открытому каналу связи можно передать информацию только о том, какие по порядку фотоны были приняты, не называя состояния фильтров и полученные значения поляризации. После этого отправитель может передавать сообщения, зашифрованные этим ключом.

Для обнаружения факта съема информации в данном протоколе используют контроль ошибок, аналогичный контролю ошибок в протоколе ВВ84, то есть сверяют случайно выбранные биты ключа. Если обнаруживаются несовпадения, то можно говорить о несанкционированном съеме информации.

#### Квантовый протокол E91 (EPR)

Дальнейшего улучшения надежности криптосистемы можно достичь, используя *эффект EPR* (Einstein–Podolsky–Rosen). Эффект EPR возникает, когда сферически симметричный атом излучает два фотона в противоположных направлениях в сторону двух наблюдателей. Фотоны излучаются с неопределенной поляризацией, но в силу симметрии их поляризации всегда противоположны.

Важной особенностью этого эффекта является то, что поляризация фотонов становится известной только после измерения. На основе EPR Экерт предложил криптографическую схему, которая гарантирует безопасность пересылки и хранения ключа.

Отправитель генерирует некоторое количество EPR фотонных пар. Один фотон из каждой пары он оставляет для себя, второй посылает своему партнеру. При этом, если эффективность регистрации близка к единице, при получении отправителем значения поляризации 1, его партнер зарегистрирует значение 0, и наоборот.

Таким образом, партнеры всякий раз, когда требуется, могут получить идентичные псевдослучайные кодовые последовательности. Практически реализация данной схемы проблематична из-за низкой эффективности регистрации и измерения поляризации одиночного фотона.

При горизонтальной поляризации (H) и использовании вертикального поляризатора (V) результат очевиден – фотон не будет зарегистрирован. При 450 поляризации фотона и вертикальном поляризаторе (V) вероятность регистрации 50%. Именно это обстоятельство и используется в квантовой криптографии.

Результаты анализа при передаче двоичных разрядов представлены в таблице 12.3. Здесь предполагается, что для передатчика логическому нулю соответствует поляризация V, а единице – +450, для принимающей стороны логическому нулю соответствует поляризация -450, а единице – Н.

*Таблица 12.3*

***Результаты анализа при передаче двоичных разрядов***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Передаваемый бит | 1 | 0 | 1 |  | 0 |
| Поляризация передачи | +450 | V | +450 |  | V |
| Поляризация приема | -450 | -450 | H |  | H |
| Двоичный сигнал приема | 0 | 0 | 1 |  | 1 |
| Результат приема | - | - | + |  | - |

В первой и четвертой колонке поляризации передачи и приема ортогональны, и результат детектирования будет отсутствовать. В колонках 2 и 3 коды двоичных разрядов совпадают, и поляризации не ортогональны. По этой причине с вероятностью 50% может быть позитивный результат в любом из этих случаев (и даже в обоих). В таблице предполагается, что успешное детектирование фотона происходит для случая колонки 3. Именно этот бит становится первым битом общего секретного ключа передатчика и приемника.

### 12.4. Квантовый криптоанализ

В результате развития квантовых компьютеров и квантовой криптографии появился *квантовый криптоанализ*. Он обладает преимуществами перед традиционным криптоанализом, потому что использует особенности работы квантового компьютера.

Основной строительной единицей квантового компьютера является *кубит* (*qubit,* Quantum Bit). Классический бит имеет, как известно, лишь два состояния – 0 и 1, тогда как множество состояний кубита значительно больше. Это означает, что кубит в одну единицу времени равен и 0, и 1, а классический бит в ту же единицу времени равен либо 0, либо 1. Таким образом, квантовым компьютерам присущ естественный параллелизм вычислений. Например, если квантовая память состоит из двух кубитов, то мы параллельно работаем со всеми ее возможными состояниями: 00, 01, 10, 11. За счет возможности параллельной работы с большим числом вариантов квантовому компьютеру необходимо гораздо меньше времени для решения задач определенного класса. К таким задачам относятся задачи разложения числа на простые множители, поиск в большой базе данных и другие.

Так, в основе системы RSA лежит предположение о том, что решение математической задачи о разложении больших чисел на простые множители на классических компьютерах требует экспоненциально большого числа операций и астрономического времени. Для решения такой задачи квантовый алгоритм, построенный на параллельных процедурах, дает возможность вычислить простые множители больших чисел за практически приемлемое время и взломать шифр RSА. Таким образом, для шифра RSA квантовый компьютер, а следовательно, квантовый криптоанализ создает реальную угрозу и снижает его криптостойкость до практически неприемлемой величины.

Процедура квантового криптоанализа может быть применена ко всем классическим шифрсистемам. Остается только создать квантовый компьютер достаточной мощности.

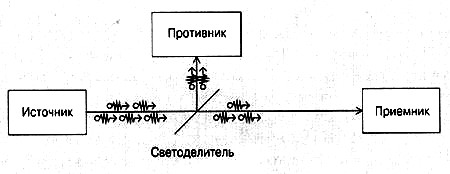
**12.5. Проблемы практической реализации систем квантовой**

### криптографии

В идеальных системах квантовой коммуникации перехват данных невозможен, так как он моментально обнаруживается участниками обмена по возникающим ошибкам в передаче. Однако реальные системы отличаются от идеальных.

Во-первых, аппаратура участников информационного обмена несовершенна, что приводит к появлению ошибок приема-передачи. В этих обстоятельствах наличие определенного уровня ошибок не должно восприниматься системой как попытка подслушивания. А наличие собственного фона ошибок позволяет противнику осуществлять перехват, маскируя неизбежно возникающие при этом искажения под собственные ошибки системы.

Во-вторых, в реальных линиях передачи существует затухание сигнала, что вынуждает отправителя увеличивать мощность импульса, т.е. число фотонов в нем, либо приводит к потере части импульсов в канале. В первом случае, если импульс содержит много фотонов, поляризованных одинаковым образом, с помощью светоделителя от него можно сделать отвод и тестировать уже его, не трогая основной сигнал. Такой перехват следует осуществлять как можно ближе к отправителю – там уровень сигнала выше (рис. 12.6). Во втором случае затухание сигнала приводит к увеличению общего уровня ошибок, и у противника увеличиваются шансы замаскировать перехват под собственные ошибки системы.



***Рис. 12.6.*** *Использование многофотонных импульсов для передачи сигнала делает возможным перехват данных путем «отвода» части фотонов*

В-третьих, существует более эффективная стратегия перехвата, чем простое угадывание базиса. Законы квантовой механики запрещают лишь идеальное клонирование квантовой системы, неидеальное клонирование при этом остается возможным. В настоящее время доказана теоретическая возможность успешного однократного копирования состояния квантовой системы с вероятностью успеха 5/6, а с ростом числа копий эта вероятность снижается до 2/3. Эксперименты по клонированию фотонов показывают результат, близкий к предсказанному теорией. Это дает противнику возможность копировать фотон и затем анализировать его поляризацию в двух различных базисах. При этом будут возникать ошибки, но их уровень будет ниже, чем при простом угадывании базиса. И если базис окажется сопоставим с собственным фоном ошибок системы, прослушивание становится возможным. Поэтому в распоряжении противника всегда есть возможность перехватить какую-то часть передаваемых битов, замаскировав неизбежно сопровождающие такой перехват ошибки под собственные ошибки системы.

Для отсеивания собственных ошибок в реальных системах квантовой криптографии необходимо применять различные протоколы коррекции, а для снижения значимости перехваченных противником битов нужно использовать процедуру усиления секретности. Для этого проще всего вырабатывать несколько «слепков» ключа, а итоговый рабочий ключ получать простым побитовым суммированием по модулю 2 этих «слепков». Тогда, чтобы наверняка определить хотя бы один бит ключа, злоумышленнику нужно знать соответствующие биты во всех «слепках». Другой возможный способ заключается в том, чтобы вырабатывать ключи из сформированного битового вектора с помощью хэш-функций.

Таким образом, в отличие от идеальных реальные системы квантовой коммуникации не способны обеспечить абсолютную секретность передаваемых данных. Это обусловлено наличием у них фона собственных ошибок, под которые можно замаскировать попытки перехвата, а также затуханием в каналах связи из-за необходимости использования многофотонных импульсов. Последнее делает возможным неразрушающий перехват данных и является практически неустранимым фактором, так как качество каналов не всегда поддается контролю, например, в радиоканале между наземным центром управления и низкоорбитальным спутником.

#### Вопросы и задания для самостоятельной работы

1. Приведите пример порядка открытого распределения ключей по квантовому каналу.
2. Какова природа секретности квантовой криптографии?
3. Назовите два основных направления развития квантовой криптографии.
4. Почему квантовый криптоанализ делает возможным быстрое вскрытие криптосистемы RSA?
5. Каковы практические проблемы квантовой криптографии?

**ЧАСТЬ 2**

**СРЕДСТВА КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ**

# ИНФОРМАЦИИ

## ТЕМА 1. НОРМАТИВНО-ПРАВОВЫЕ ОСНОВЫ КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ

**1.1. Действующие стандарты криптографической защиты**

### информации

Криптографическая защита информации обладает одной очень важной особенностью, резко выделяющей ее из общей массы прикладных направлений информационных технологий. Качество предлагаемых решений здесь оценить значительно сложнее, чем в других направлениях. Действительно, если качество алгоритма сжатия данных измеряется непосредственно, то для оценки стойкости шифра нужно выполнить огромный объем работы без какой-либо гарантии адекватности результата.

До того момента, когда индустрия отчетливо осознала этот факт, было совершено очень большое количество ошибок. Была выпущена масса программных продуктов, предлагающих шифрование в качестве основной или вспомогательной функции, в которых использовались нестойкие шифры. Это способно нанести пользователям таких средств весьма ощутимый ущерб, в том числе и экономический, так как создает иллюзию защищенности, в то время когда реальной защиты нет.

Следовательно, нужен механизм, гарантирующий качество предлагаемых в области криптозащиты решений. В полном объеме данная задача может быть решена лишь в рамках государственного подхода – путем принятия соответствующих стандартов.

Вмешательство государства в эту сферу необходимо и потому, что открытая, или гражданская, криптография еще сравнительно молода, а специальные государственные структуры обладают в области защиты информации несравненно большим опытом, причем значительная часть знаний, которыми они владеют, до сих пор остается засекреченной. Именно поэтому действующие государственные стандарты в области криптографии разрабатываются или целиком спецслужбами, как в России, или при таком участии спецслужб, когда за ними остается решающая роль, как, например, в США.

Таким образом, роль стандартов в области криптозащиты – это не столько обеспечение совместимости решений от разных производителей, сколько определенная гарантия качества решений, заверенных государством.

Федеральный орган исполнительной власти, осуществляющий функции по оказанию государственных услуг, управлению государственным имуществом в сфере технического регулирования и метрологии в России – это Ростехрегулирование (до 2004 г. был Госстандарт).

Приказом Ростехрегулирования от 28 декабря 2007 г. был создан *технический комитет по стандартизации «Криптографическая защита информации»*, получивший сокращенное наименование ТК 26. В марте 2008 г. было проведено первое заседание ТК 26 «Криптографическая защита информации».

Приказом Федерального агентства по техническому регулированию и метрологии от 12.08.2011 № 4402 председателем ТК 26 утвержден действительный член Академии криптографии Российской Федерации, доктор физикоматематических наук, профессор Кузьмин А.С. Заместители председателя ТК 26: кандидат технических наук Игорь Качалин и генеральный директор «ИнфоТеКС» Андрей Чапчаев (ответственный секретарь ТК 26).

ФСБ России, как федеральный орган исполнительной власти, обеспечивающий информационную безопасность Российской Федерации, в пределах своих полномочий руководит деятельностью технического комитета *по вопросам функционального и предметного содержания стандартизации в области криптографической защиты* сведений, составляющих государственную тайну или относимых к охраняемой в соответствии с законодательством Российской Федерации информации ограниченного доступа. ФСБ также выполняет иные функции, касающиеся вопросов организации разработки, принятия и применения документов по стандартизации шифровальных (криптографических) средств защиты, а также вопросов их применения в защищенных системах.

Ростехрегулирование, как национальный орган Российской Федерации по стандартизации, осуществляет *методическое руководство деятельностью технического комитета и координацию* его деятельности с другими техническими комитетами по работам в области защиты информации с использованием криптографических средств.

Таким образом, ТК 26 – это постоянно действующий коллегиальный орган, призванный организовать разработку и экспертизу проектов национальных, межгосударственных и международных стандартов в закрепленной области деятельности, обеспечить анализ технических регламентов и стандартов на предмет их обновления и дальнейшего использования, а также другие функции.

Технический комитет в своей работе руководствуется: Законом Российской Федерации «О государственной тайне»; федеральными законами:

«О техническом регулировании»;

«О Федеральной службе безопасности»;

«Об информации, информационных технологиях и о защите информации»;

«О коммерческой тайне»; «О персональных данных»;

другими нормативными правовыми актами, регулирующими отношения в

области защиты информации; документами Национальной системы стандартизации Российской Феде-

рации.

В настоящее время в России действуют пять стандартов криптографической защиты:

* ГОСТ 28147-89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования.
* ГОСТ Р 34.10-2012. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи.
* ГОСТ Р 34.11-2012. Информационная технология. Криптографическая защита информации. Функция хэширования.
* ГОСТ Р 34.12-2015. Информационная технология. Криптографическая защита информации. Блочные шифры.
* ГОСТ Р 34.13-2015. Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров.

#### Стандарт симметричного шифрования ГОСТ 28147-89

Для зашифрования в этом режиме открытый текст сначала разбивается на левую и правую половины L и R. На i-м цикле используется подключ ki:

Ri + 1 = Li

**= xor).**

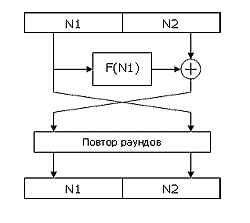


**(**

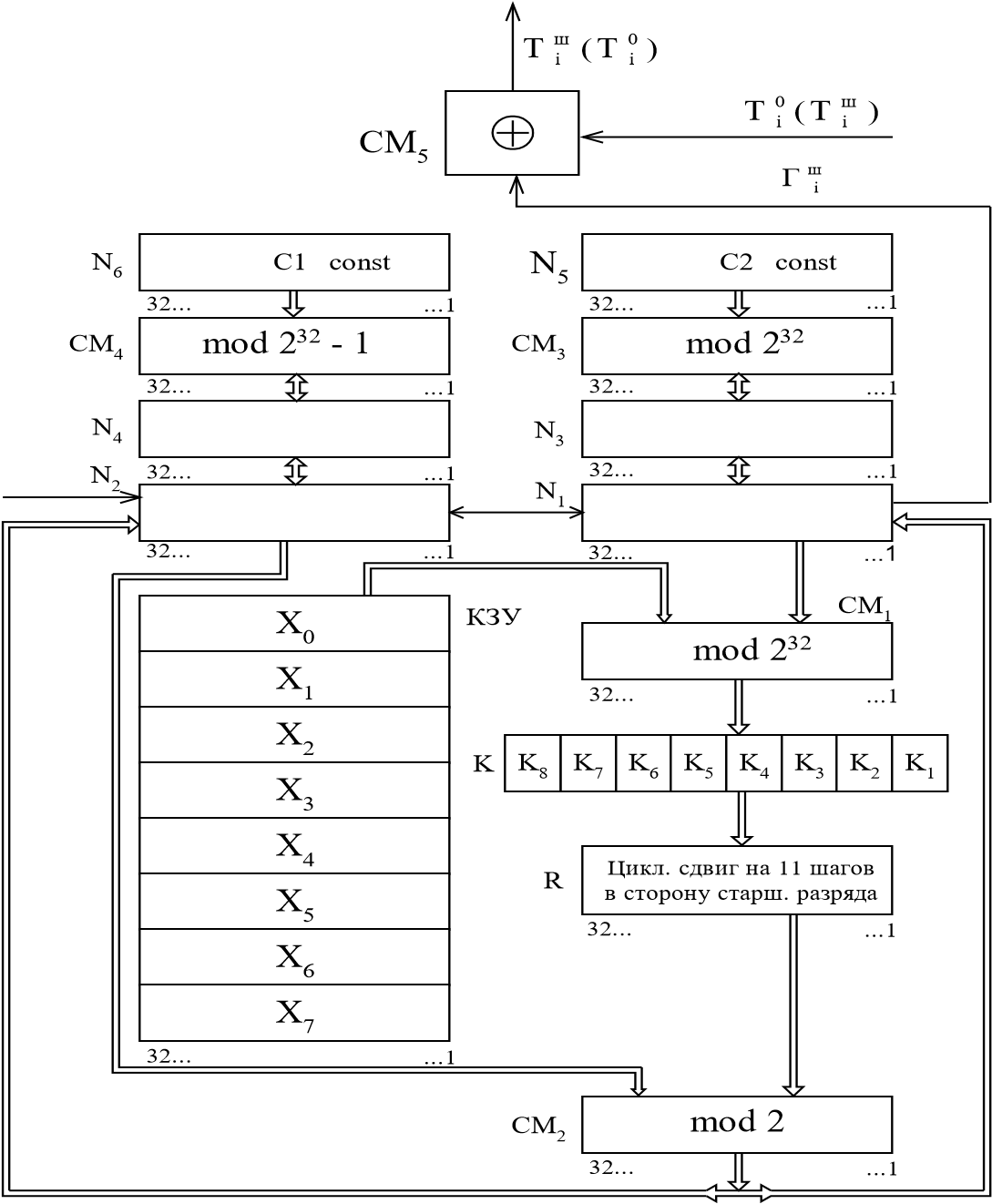


Для генерации подключей исходный 256-битный ключ разбивается на восемь 32-битных блоков: K1…K8.

Расшифрование выполняется так же, как и зашифрование, но инвертируется порядок подключей Ki.



***Рис. 1.1.*** *Схема сеть Фейстеля*



***Рис. 1.2.*** *Структурная схема алгоритма ГОСТ 28147-89*

Функция f(Li,Ki) вычисляется следующим образом:

*Ri-1 и Ki складываются по модулю 232.*

Результат разбивается на восемь 4-битовых подпоследовательностей, каждая из которых поступает на вход своего S-блока. Общее количество Sблоков ГОСТа – восемь, т.е. столько же, сколько и подпоследовательностей.

...

Сложение по модулю 232

Циклический сдвиг на 11 бит влево

Ri

-

1

/32/

Ki

f(Ri

-

1

,Ki

)

/32/

/32/

S1

S2

S3

S8

***Рис. 1.3.*** *Блок-схема функции f**алгоритма ГОСТ 28147-89*

*Каждый S-блок представляет собой перестановку чисел от 0 до 15.*Первая 4-битная подпоследовательность попадает на вход первого S-блока, вторая – на вход второго и т.д.

Если S-блок выглядит так:

1, 15, 13, 0, 5, 7, 10, 4, 9, 2, 3, 14, 6, 11, 8, 12

и на входе S-блока 0, то на выходе будет 1, если 5, то на выходе будет 7 и т.д.

Выходы всех восьми S-блоков объединяются в 32-битное слово, затем все слово циклически сдвигается влево на 11 бит. Пример S-блоков:

№ S-блока Значение

1. 4 10 9 2 13 8 0 14 6 11 1 12 7 15 5 3
2. 14 11 4 12 6 13 15 10 2 3 8 1 0 7 5 9
3. 5 8 1 13 10 3 4 2 14 15 12 7 6 0 9 11
4. 7 13 10 1 0 8 9 15 14 4 6 12 11 2 5 3
5. 6 12 7 1 5 15 13 8 4 10 9 14 0 3 11 2
6. 4 11 10 0 7 2 1 13 3 6 8 5 9 12 15 14
7. 13 11 4 1 3 15 5 9 0 10 14 7 6 8 2 12
8. 1 15 13 0 5 7 10 4 9 2 3 14 6 11 8 12

Государственные организации и многие коммерческие структуры обязаны использовать для защиты данных сертифицированные средства защиты.

Получение сертификата возможно только в том случае, если *«в указанных криптосредствах реализованы криптографические алгоритмы, объявленные государственными или отраслевыми стандартами Российской* *Федерации» –* это ГОСТ 28147-89 и два новых стандарта ГОСТ Р 34.12-2015 и ГОСТ Р 34.132015.

ГОСТ 28147-89 может работать в 4 различных режимах:

*Простая замена* (для ключевой информации) – самый быстрый.

*Гаммирование* (операция побитового сложения по модулю 2 является обратной самой себе и наиболее просто реализуется аппаратно).

*Гаммирование с обратной связью* (очередной элемент гаммы вырабатывается с помощью определенных преобразований предыдущей последовательности информации).

Влияния на криптостойкость обратная связь не оказывает, служит дополнительной защитой от некоторых типов атак.

*Имитовставка –* контрольная комбинация, зависящая от открытых данных и секретной ключевой информации, для того, чтобы обнаружить все случайные или преднамеренные изменения в зашифрованных данных.  *Достоинства ГОСТ* 28147-89:

* бесперспективность силовой атаки (XSL-атаки в учет не берутся, т.к. их эффективность на данный момент полностью не доказана);
* эффективность реализации и, соответственно, высокое быстродействие на современных компьютерах.  *Недостатки ГОСТ* 28147-89:

Стандарт не специфицирует алгоритм генерации S-блоков (таблицы замен). С одной стороны, это может являться дополнительной секретной информацией (помимо ключа), а с другой – поднимает ряд проблем:

* *нельзя определить криптостойкость* алгоритма, не зная таблицы замен

(таблицы бывают «удачные» и «неудачные»);

* реализации алгоритма от различных производителей могут использовать разные таблицы замен и *могут быть несовместимы* между собой;
* генерацией и проверкой S-блоков для реализаций алгоритма занимался ФАПСИ (при лицензировании алгоритма) – Федеральное агентство правительственной связи и информации. Не было уверенности, что эта организация не выдаст «специальных» блоков. В настоящее время упразднено. Его функции распределены между ФСБ и Минобороны РФ.

Недостатки были устранены в новых стандартах симметричного шифрования ГОСТ Р 34.12-2015 и ГОСТ Р 34.13-2015. Однако, учитывая тот факт, что сертификаты на СКЗИ выдаются на пять лет, а новые стандарты вступили в действие с января 2016 года, СКЗИ на алгоритмах ГОСТ 28147-89 могут использоваться до 2020 года.

#### ГОСТ Р 34.12-2015

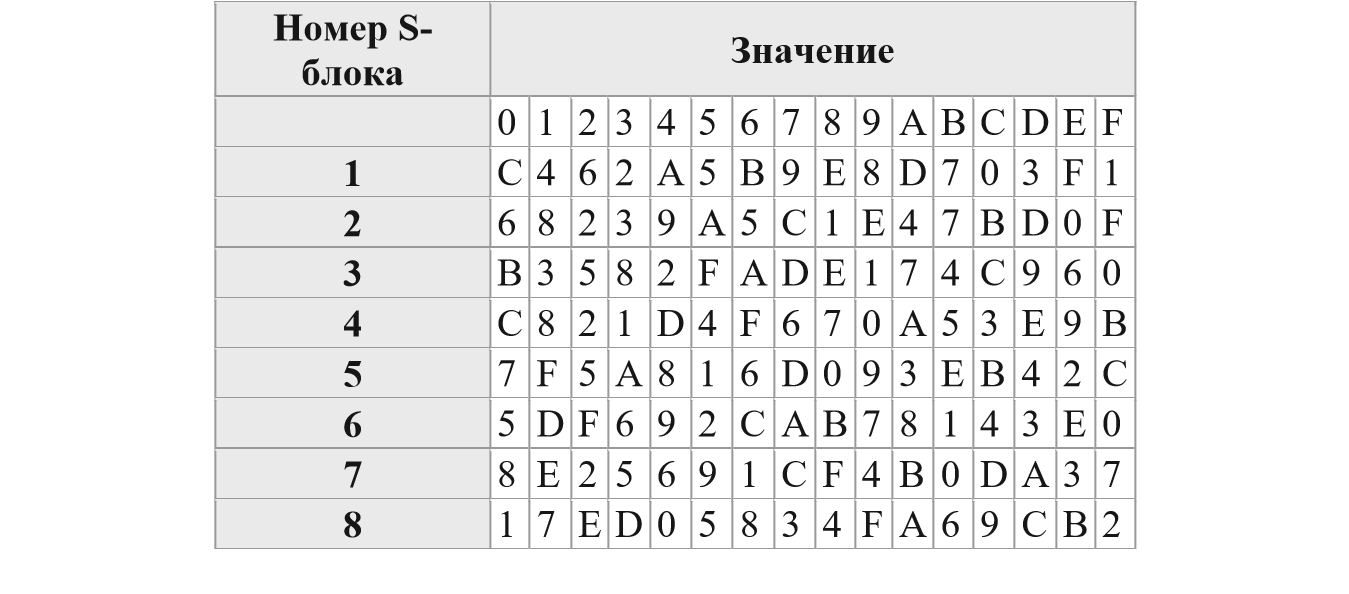
В июне этого года в России был принят новый стандарт блочного шифрования – ГОСТ Р 34.12-2015. Этот стандарт помимо старого ГОСТ 28147-89, который теперь называется «Магма» и имеет фиксированный набор подстановок, содержит описание нового блочного шифра «Кузнечик».

Таким образом, ГОСТ З 34.12-2015 определяет два базовых алгоритма блочного шифрования с длинами блоков *n* = 128 бит и *n* = 64 бит и длинами ключей *k* = 256 бит:  шифр с длиной блока *n* = 128 бит – блочный шифр «Кузнечик» («Kuznyechik») – *SP-сеть*;

 шифр с длиной блока *n* = 64 бит – блочный шифр «Магма» («Magma» – *сеть Фейстеля*.

Алгоритмы могут применяться в криптографических методах обработки и защиты информации, в том числе для обеспечения конфиденциальности, аутентичности и целостности информации при ее передаче, обработке и хранении в автоматизированных системах.

В отличие от ГОСТ 28147-89, в котором заполнение узлов замены (S-блоков) производилось разработчиками СКЗИ, шифр «Магма» содержит единственный у*зел замены, определенный Техническим комитетом по стандартизации «Криптографическая защита информации» (ТК 26):*



Используемый набор подстановок был выбран исходя из обеспечения наилучших характеристик, определяющих невозможность применения дифференциального и линейного методов криптографического анализа. Данный узел замен предлагался ТК 26 при международной стандартизации ГОСТ 28147-89 в составе стандарта шифрования ISO/IEC 18033-3 и рекомендовался отечественным разработчикам СКЗИ.

Фиксированный набор подстановок позволит максимально упростить разработку взаимодействующих информационно-телекоммуникационных систем.

Вторым шифром, включенным в стандарт, является опубликованный в 2013 г. российскими специалистами криптографический алгоритм «Кузнечик» с длиной входного блока, равной 128 битам, и 256-битовым ключом.

Так же, как и AES, алгоритм «Кузнечик» основан на использовании SPсети, однако при его разработке удалось реализовать ряд синтезных решений, которые позволяют исключить выявленные за последние годы недостатки алгоритма AES.

Новый шифр «Кузнечик» представляет собой SP-сеть: преобразование, состоящее из нескольких одинаковых раундов, при этом каждый раунд состоит из нелинейного и линейного преобразований, а также операции наложения ключа. В отличие от сети Фейстеля, при использовании SP-сети преобразуется весь входной блок, а не его половина. Такая структура иногда также называется AES-like (похожей на AES), однако в отличие от последнего у «Кузнечика» есть ряд особенностей:

* линейное преобразование может быть реализовано в с помощью регистра сдвига;
* ключевая развертка реализована с помощью сети Фейстеля, в которой в качестве функции используется раундовое преобразование исходного алгоритма.

Шифрование основано на последовательном применении нескольких однотипных раундов, каждый из которых содержит три преобразования: сложение с раундовым ключом, преобразование блоком подстановок и линейное преобразование.

128-битный входной вектор очередного раунда складывается побитно с раундовым ключом:



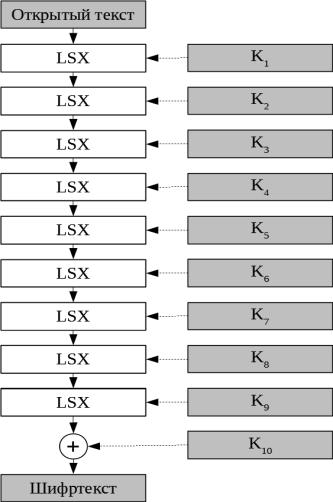
В качестве нелинейного преобразования была выбрана подстановка, использованная в хеш-функции «Стрибог». Как показали 3 года интенсивных исследований отечественными и зарубежными специалистами, ее характеристики позволяют эффективно обеспечивать защиту от современных методов криптоанализа.

Линейное преобразование может быть реализовано не только как обычно в блочных шифрах – матрицей, но и с помощью РСЛОС – линейного регистра сдвига с обратной связью, который движется 16 раз.

Сам регистр реализуется над полем Галуа по модулю неприводимого многочлена степени 8: .

Алгоритм развертки ключа основан на схеме Фейстеля и использует в качестве функции усложнения итерационное преобразование алгоритма «Кузнечик». При таком подходе итерационные ключи шифрования имеют сложные нелинейные связи, что затрудняет применение атак со связанными ключами и метода биклик.

Выработка раундовых ключей производится из мастер-ключа. Первые два получаются разбиением мастер-ключа пополам. Далее для выработки очередной пары раундовых ключей используется 8 итераций сети Фейстеля, где, в свою очередь, в качестве раундовых ключей используется счетчиковая последовательность, прошедшая через линейное преобразование алгоритма.



***Рис. 1.5.*** *Блок-схема шифрования по алгоритму «Кузнечик»*

Расшифрование реализуется обращением базовых преобразований и применением их в обратном порядке.

##### ГОСТ 34.13-2015

Стандарт ГОСТ 34.13-2015 определяет шесть режимов работы алгоритмов блочного шифрования:

* режим *простой замены* (Electronic Codebook, ЕСВ);
* режим *гаммирования* (Counter, CTR);
* режим *гаммирования с обратной связью по выходу* (Output Feedback,

OFB);

* режим *простой замены с зацеплением* (Cipher Block Chaining, СВС);  режим *гаммирования с обратной связью по шифртексту* (Cipher Feedback, CFB);
* режим *выработки имитовставки* (Message Authentication Code algorithm).

Данные режимы могут использоваться в качестве режимов для блочных шифров с произвольной длиной блока *n*.

##### Стандарт цифровой подписи ГОСТ Р 34.10-1994

Первый отечественный стандарт цифровой подписи ГОСТ Р34.10-94 был опубликован Госстандартом РФ в мае 1994 года и вступил в действие с 1 января 1995 года.

В предварительном варианте он был введен как ведомственный стандарт на цифровую подпись ЦБ РФ и использовался в этом качестве с сентября 1993 года по декабрь 1994 года.

Алгоритмы вычисления и проверки подписи в ГОСТ Р 34.10 устроены на основе алгоритма El Gamal. В нем используются следующие параметры:

р – большое простое число длиной от 509 до 512 бит либо от 1020 до

1024 бит; q – простой сомножитель числа (р-1), имеющий длину 254...256 бит; а – любое число, меньшее (р-1), причем такое, что аq mod p=1; х – некоторое число, меньшее q; у = аx mod р.

Кроме того, этот алгоритм использует однонаправленную хэш-функцию

Н(х).

##### Стандарт вычисления хеш-функции ГОСТ Р 34.11-94

Стандарт ГОСТ Р 34.11-94 определяет хэш-функцию, основанную на использовании стандартного симметричного алгоритма ГОСТ 28147-89.

##### Стандарт цифровой подписи ГОСТ Р 34.10-2001

Алгоритм ГОСТ Р 34.10-2001 аналогичен алгоритму ГОСТ Р 34.10-94 (в старом стандарте некоторые операции выполнялись *в конечном поле GF(p),* а в новом – *в группе точек эллиптической кривой над полем GF(p)*.

ГОСТ Р 34.10-2001 основан на эллиптических кривых. Его *стойкость определяется сложностью взятия дискретного логарифма в группе точек эллиптической кривой,* а также на стойкости хэш-функции по ГОСТ Р 34.11.

К сообщению *М* дописывается цифровая подпись размером 512 бит и текстовое поле. В текстовом поле могут содержаться, например, дата и время отправки или различные данные об отправителе.

##### Стандарт цифровой подписи ГОСТ Р 34.10-2012

Полное название: «*ГОСТ Р 34.10-2012. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи*» – российский стандарт, описывающий алгоритмы формирования и проверки электронной цифровой подписи. Принят и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 7 августа 2012 года № 215-ст вместо ГОСТ Р 34.102001.

##### Стандарт вычисления хеш-функции ГОСТ Р 34.11-2012

«ГОСТ Р 34.11-2012. Информационная технология. Криптографическая защита информации. Функция хэширования» – действующий российский криптографический стандарт, определяющий алгоритм и процедуру вычисления хеш-функции ***(****проектное название «Стрибог»).*



*Отличительные особенности по сравнению со старым стандартом:*

В ГОСТ Р 34.11–2012 размер блоков сообщения и внутреннего состояния хэш-функции составляет 512 бит против 256 бит в ГОСТ Р 34.11–1994. Также новый стандарт определяет две функции хэширования с длинами хэш-кода 256 и 512 бит, в то время как в старом стандарте длина хэш-кода может быть только 256 бит. Возможность вариации выходного хэша может быть полезна в случае встроенных реализаций с ограниченными ресурсами, или наличия каких-то дополнительных требований в области криптографии.

Основное отличие современной хэш-функции от старой – функция сжатия. В ГОСТ Р 34.11–2012 используется функции сжатия, в основе которой лежат *три преобразования: нелинейное биективное преобразование (обозначается S), перестановка байт (обозначается P), линейное преобразование (обозначается L)*. В ГОСТ Р 34.11–1994 используется функция сжатия, основанная на симметричном блочном шифре ГОСТ Р 28147-89, также эта функция использует операции перемешивания.

При вычислении новой хэш-функции, если размер сообщения не кратен размеру обрабатываемого блока (для современного стандарта – 512 бит, для старого стандарта – 256 бит), то такой блок дополняется вектором (00 … 01). При вычислении старой хэш-функции неполный блок дополняется значением (00 … 0). Считается, что дополнение (00 … 01) лучше, чем (00 … 0) с криптографической точки зрения.

Еще одно отличие состоит в том, что стандарт ГОСТ Р 34.11-1994 не определял значение инициализационного вектора. В то время как в стандарте ГОСТ Р 34.11-2012 значение инициализационного вектора фиксировано и определенно в стандарте: для хэш-функции с размером выходного хэша 512 бит это вектор (00 … 0), для хэш-функции с размером выходного хэш-кода 256 бит – (000000010 … 100000001) (все байты равны 1).

***Зарубежные стандарты*** криптографической защиты представлены также стандартами симметричного шифрования, стандартами цифровой подписи и стандартами хэширования. В качестве примера ниже представлены самые распространенные из них.

Стандарты симметричного шифрования:

* Data Encryption Standard (DES, устарел)
* Triple-DES
* Advanced Encryption Standard (AES)  RSA исходный алгоритм открытого ключа
* OpenPGP
* CipherSaber

Стандарты цифровой подписи:

* Digital Signature Standard (DSS), based on the Digital Signature Algorithm (DSA)
* RSA

Стандарты хэширования:

* MD5 128-bit (устаревающий)
* SHA доступен в вариантах 160, 256, 384 и 512-бит
* HMAC
* PBKDF2 (RFC 2898) Стандарты инфраструктуры открытых ключей:
* X.509 Public Key Certificates

### 1.2. Федеральный закон об электронной подписи

*Федеральный закон об электронной подписи № 63-ФЗ* принят Государственной Думой 25 марта 2011 года. Одобрен Советом Федерации 30 марта 2011 года. Последняя редакция на момент написания книги – 30.12.2015.

Настоящий Федеральный закон регулирует отношения в области использования электронных подписей при совершении гражданско-правовых сделок, оказании государственных и муниципальных услуг, исполнении государственных и муниципальных функций, при совершении иных юридически значимых действий, в том числе в случаях, установленных другими федеральными законами (в ред. Федерального закона от 05.04.2013 N 60-ФЗ).

**1.3. Нормативно-правовые акты ФСБ по обеспечению**

### безопасности персональных данных с использованием СКЗИ

В соответствии с требованиями ФСБ для обеспечения безопасности персональных данных необязательно применение средств шифрования операторами.

Несмотря на это, без криптографической защиты не обойтись при передаче персональных данных по незащищенному каналу связи, сюда входит, например, удаленная работа сотрудников с базой данных клиентов, обмен информацией между филиалами и головным офисом, передача личной информации работников третьим лицам. В том или ином виде подобные задачи присутствуют практически в каждой организации.

Можно выделить три основных документа по криптографической защите персональных данных в Российской Федерации:

1. Методические рекомендации по обеспечению с помощью криптосредств безопасности персональных данных при их обработке в информационных системах персональных данных с использованием средств автоматизации.
2. Типовые требования по организации и обеспечению функционирования шифровальных (криптографических) средств, предназначенных для защиты информации, не содержащей сведений, составляющих государственную тайну, в случае их использования для обеспечения безопасности персональных данных при их обработке в информационных системах персональных данных.
3. Приказ ФСБ № 378 от 10 июля 2014 г. «Об утверждении состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных с использованием средств криптографической защиты информации, необходимых для выполнения установленных Правительством Российской Федерации требований к защите персональных данных для каждого из уровней защищенности».

***Методические рекомендации по обеспечению с помощью криптосредств безопасности персональных данных*** при их обработке в информационных системах персональных данных с использованием средств автоматизации (утв. ФСБ РФ 21 февраля 2008 г. N 149/54-144), разработаны в соответствии с п. 2 постановления Правительства Российской Федерации от 17 ноября 2007 года N 781 «Об утверждении ***Положения об обеспечении безопасности персональных данных*** при их обработке в информационных системах персональных данных».

Документ был принят еще во времена действия документов ФСТЭК (58го Приказа, 781 Постановления Правительства) и дополнял их содержание. Рассматриваемый нормативный документ не был зарегистрирован в Минюсте.

Требования вышеуказанного документа (а также других нормативных актов в области криптографической защиты персональных данных) не распространяются на следующие случаи:

 обработка персональных данных без использования средств автоматизации;  работа с персональными данными, составляющими государственную

тайну;  использование технических средств, расположенных за пределами РФ.

В документе сказано, что в случае использования средств криптографической защиты нужно разрабатывать модель угроз по требованиям как ФСТЭК, так и ФСБ (за редким исключением). Операторы могут сами составлять модели угроз и нарушителя, лишь при необходимости привлекая лицензиатов ФСБ. Все угрозы в документе делятся на атаки и угрозы, не являющиеся атаками, приведены примеры распространенных угроз. Вы можете руководствоваться Методикой как справочным материалом при написании модели по новым требованиям.

Модель угроз верхнего уровня определяет характеристики безопасности персональных данных и других объектов защиты. В детализированной модели угроз обозначены требуемые условия криптозащиты.

На модель угроз влияют различные факторы: условия создания и использования персональных данных, формы представления персональных данных, характеристики безопасности и т.д.

Кроме привычных характеристик – целостности, конфиденциальности и доступности – выделяют также неотказуемость, учетность, аутентичность и адекватность.

Пример модели угроз верхнего уровня:

* Угроза конфиденциальности персональных данных.
* Угроза целостности персональных данных.
* Угроза доступности персональных данных.

Документ посвящен не только вопросам формирования модели угроз, но и особенностям составления адекватной модели нарушителя.

Все нарушения в Методических рекомендациях делятся на два класса: прямые и косвенные нарушения безопасности персональных данных (угрозы, создающие условия для возникновения прямых угроз). Выделяется 6 основных типов нарушителей: Н1, Н2, Н3, Н4, Н5, Н6. Чем выше цифра, тем больше возможностей, нарушитель каждого следующего типа наследует возможности предыдущего. Оператор самостоятельно определяет уровень подготовки нарушителей, доступные им инструменты и делает предположение о сговоре. В документе указаны основные характеристики нарушителя каждого типа.

В документе определены 6 уровней криптозащиты: KC1, KC2, KC3, KB1, KB2, KA1 и 6 классов криптосредств с аналогичными названиями. Информационные системы персональных данных также разделяются на 6 классов, в зависимости от наивысшей категории нарушителя. АК1 – если наивысшая категория нарушителя Н1, АК2 – если Н2, АК3 – если Н3, АК4 – если Н4, АК5 – если Н5, АК6 – если Н6. Соответственно распределены средства криптозащиты: АК1 – КС1, АК2 – КС2, АК3 – КС3, АК4 – КВ1, АК5 – КВ2, АК6 – КА1.

***Типовые требования*** ***по организации и обеспечению функционирования шифровальных (криптографических) средств,*** предназначенных для защиты информации, не содержащей сведений, составляющих государственную тайну, в случае их использования для обеспечения безопасности персональных данных при их обработке в информационных системах персональных данных», утвержденные руководством 8 Центра ФСБ России 21.02.2008 № 149/6/6-622 – документ официально опубликован не был.

Типовые требования были написаны в тот же период, что и Методические рекомендации, не зарегистрированы в Минюсте. В документе содержится полезная для изучения информация. Подробно описаны обязанности пользователей криптосредств, обозначены основные правила для них:

* не допускать копирования ключевой информации;
* не разглашать информацию о ключах;
* не записывать на ключевые носители постороннюю информацию и т.д.

Описан процесс уничтожения ключа, основные требования к помещениям, представлены типовые формы журналов. На основании информации, содержащейся в документе можно построить некоторые полезные инструкции.

***Приказ ФСБ РФ***от 10 июля 2014 г. **N 378** «Об утверждении состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных с использованием средств криптографической защиты информации, необходимых для выполнения установленных правительством Российской Федерации требований к защите персональных данных для каждого из уровней защищенности».Зарегистрировано в Минюсте России 18 августа 2014 г.

На сегодняшний день это главный документ в области криптографической защиты персональных данных, и его действие распространяется на все информационные системы персональных данных, в которых используются в качестве защиты криптографические средства защиты информации. Приказом определены требования не только к криптографической защите, но и к режиму обеспечения безопасности помещений, порядок хранения носителей информации и другие организационные меры в зависимости от уровня защищенности системы.

Отдельно указано, что оператору следует использовать СЗИ, прошедшие оценку соответствия – сертифицированные по требованиям безопасности. Защитные меры описаны очень подробно, включают в себя требования к оснащенности помещений (замки, приспособления для опечатывания, решетки на окна и т.д.). В отличие от положений Методических рекомендаций в Приказе 378 класс СКЗИ определяется относительно уровня защищенности и актуального типа угроз. Возможности злоумышленника учитываются лишь при определении класса СКЗИ для 4 уровня защищенности.

Зависимость от уровня защищенности и типа угроз достаточно очевидна, и оператор почти всегда может выбрать класс СКЗИ из нескольких вариантов.

Документ отличается четкой логикой изложения – достаточно знать уровень защищенности своей системы – требования к информационным системам персональных данных каждого уровня представлены в отдельных разделах. Требования наследуются от более низких уровней к более высоким, информационная система персональных данных 1-го уровня защищенности должна отвечать требованиям для информационных систем персональных данных 2-го, 3го и 4-го уровней.

*Таблица 1.1*

**Класс СКЗИ**



**1.4. Нормативно-правовые акты ФСБ по обеспечению**

### функционирования и эксплуатации СКЗИ

**Положение «*О разработке, производстве, реализации и эксплуатации шифровальных (криптографических) средств защиты информации*»** (Положение ПКЗ-2005). Зарегистрировано в Минюсте РФ 3 марта 2005 г. № 6382.

ПРИКАЗ ФСБ ОТ 09 ФЕВРАЛЯ 2005 Г. № 66 ОБ УТВЕРЖДЕНИИ ПОЛОЖЕНИЯ О РАЗРАБОТКЕ, ПРОИЗВОДСТВЕ, РЕАЛИЗАЦИИ И ЭКСПЛУАТАЦИИ ШИФРОВАЛЬНЫХ (КРИПТОГРАФИЧЕСКИХ) СРЕДСТВ ЗАЩИТЫ ИНФОРМАЦИИ (ПОЛОЖЕНИЕ ПКЗ-2005).

Настоящее Положение разработано во исполнение Федерального закона от 3 апреля 1995 года N 40-ФЗ «О федеральной службе безопасности» и Положения о Федеральной службе безопасности Российской Федерации, утвержденного Указом Президента Российской Федерации от 11 августа 2003 года N 960.

Положение регулирует отношения, возникающие при *разработке, производстве, реализации и эксплуатации* шифровальных (криптографических) средств защиты информации с ограниченным доступом, *не содержащей сведений, составляющих государственную тайну* (далее – информация конфиденциального характера).

**1.5. Виды работ и услуг, составляющих лицензируемую**

### деятельность с использованием СКЗИ

Данным постановлением утвержден перечень работ и услуг, при осуществлении которых требуется получение лицензии ФСБ:

* разработка, модернизация, изготовление, монтаж, установка, наладка, ремонт, передача (продажа) *средств шифрования* (криптографии);
* разработка, модернизация, изготовление, монтаж, установка, наладка, ремонт и сервисное обслуживание, передача (продажа) *информационных и телекоммуникационных систем*, защита которых обеспечена средствами шифрования (криптографии);
* разработка, модернизация, изготовление, монтаж, установка, наладка, ремонт и сервисное обслуживание *средств производства ключевых документов*;
* производство устройств, применяемых с целью подтверждения доступа к информации, оборудованию в телекоммуникационных, информационных системах, в которых используются средства шифрования (криптографии);
* обслуживание средств шифрования (криптографии), предусмотренные эксплуатационными и техническими документами на данное оборудование, кроме случаев выполнения указанных работ для обеспечения личных потребностей юридических лиц или индивидуальных предпринимателей;
* услуги по шифрованию информации, не содержащей данных, которые являются государственной тайной, при помощи средств шифрования (криптографии), осуществляемые в интересах юридических лиц, индивидуальных предпринимателей, частных лиц;
* услуги по имитозащите информации, не содержащей сведений, которые являются государственной тайной, при помощи средств шифрования (криптографии), осуществляемые в интересах юридических лиц, индиивидуальных предпринимателей, частных лиц;
* предоставление каналов связи физическим и юридическим лицам с целью передачи информации при помощи средств шифрования (криптографии);
* изготовление и распределение ключевой информации и (или) ключевых документов с применением программных, аппаратных, программно-аппаратных средств, комплексов и систем, предназначенных для изготовления и распределения ключевых документов средств шифрования (криптографии).

**1.6. Порядок лицензирования деятельности с использованием**

### СКЗИ

Лицензирование деятельности в области шифрования производится на основании Постановления Правительства РФ от 16.04.2012 N 313 «Об утверждении Положения о лицензировании деятельности по разработке, производству, распространению шифровальных (криптографических) средств, информационных систем и телекоммуникационных систем, защищенных с использованием шифровальных (криптографических) средств, выполнению работ, оказанию услуг в области шифрования информации, техническому обслуживанию шифровальных (криптографических) средств, информационных систем и телекоммуникационных систем, защищенных с использованием шифровальных (криптографических) средств (за исключением случая, если техническое обслуживание шифровальных (криптографических) средств, информационных систем и телекоммуникационных систем, защищенных с использованием шифровальных (криптографических) средств, осуществляется *для обеспечения собственных нужд* юридического лица или индивидуального предпринимателя)».

К шифровальным (криптографическим) средствам (средствам криптографической защиты информации), включая документацию на эти средства, относятся:

а) средства шифрования – аппаратные, программные и программноаппаратные шифровальные (криптографические) средства, реализующие алгоритмы криптографического преобразования информации для ограничения доступа к ней, в том числе при ее хранении, обработке и передаче;

б) средства имитозащиты – аппаратные, программные и программно-

аппаратные шифровальные (криптографические) средства (за исключением средств шифрования), реализующие алгоритмы криптографического преобразования информации для ее защиты от навязывания ложной информации, в том числе защиты от модифицирования, для обеспечения ее достоверности и некорректируемости, а также обеспечения возможности выявления изменений, имитации, фальсификации или модифицирования информации;

в) средства электронной подписи;

г) средства кодирования – средства шифрования, в которых часть криптографических преобразований информации осуществляется с использованием ручных операций или с использованием автоматизированных средств, предназначенных для выполнения таких операций;

д) средства изготовления ключевых документов – аппаратные, программные, программно-аппаратные шифровальные (криптографические) средства, обеспечивающие возможность изготовления ключевых документов для шифровальных (криптографических) средств, не входящие в состав этих шифровальных (криптографических) средств;

е) ключевые документы – электронные документы на любых носителях

информации, а также документы на бумажных носителях, содержащие ключевую информацию ограниченного доступа для криптографического преобразования информации с использованием алгоритмов криптографического преобразования информации (криптографический ключ) в шифровальных (криптографических) средствах;

ж) аппаратные шифровальные (криптографические) средства – устройства и их компоненты, в том числе содержащие ключевую информацию, обеспечивающие возможность преобразования информации в соответствии с алгоритмами криптографического преобразования информации без использования программ для электронных вычислительных машин;

з) программные шифровальные (криптографические) средства – программы для электронных вычислительных машин и их части, в том числе содержащие ключевую информацию, обеспечивающие возможность преобразования информации в соответствии с алгоритмами криптографического преобразования информации в программно-аппаратных шифровальных (криптографических) средствах, информационных системах и телекоммуникационных системах, защищенных с использованием шифровальных (криптографиче-

ских) средств;

и) программно-аппаратные шифровальные (криптографические) сред-

ства – устройства и их компоненты (за исключением информационных систем и телекоммуникационных систем), в том числе содержащие ключевую информацию, обеспечивающие возможность преобразования информации в соответствии с алгоритмами криптографического преобразования информации с использованием программ для электронных вычислительных машин, предназначенных для осуществления этих преобразований информации или их части.

Настоящее Положение *не распространяется* на деятельность с использованием:

а) шифровальных (криптографических) средств, предназначенных для защиты информации, содержащей сведения, составляющие государственную тайну;

б) шифровальных (криптографических) средств, а также товаров, со-

держащих шифровальные (криптографические) средства, реализующих либо симметричный криптографический алгоритм, использующий криптографический ключ длиной, не превышающей 56 бит, либо асимметричный криптографический алгоритм, основанный либо на методе разложения на множители целых чисел, размер которых не превышает 512 бит, либо на методе вычисления дискретных логарифмов в мультипликативной группе конечного поля размера, не превышающего 512 бит, либо на методе вычисления дискретных логарифмов в иной группе размера, не превышающего 112 бит;

в) товаров, содержащих шифровальные (криптографические) средства, имеющих либо функцию аутентификации, включающей в себя все аспекты контроля доступа, где нет шифрования файлов или текстов, за исключением шифрования, которое непосредственно связано с защитой паролей, персональных идентификационных номеров или подобных данных для защиты от несанкционированного доступа, либо имеющих электронную подпись;

г) шифровальных (криптографических) средств, являющихся компонентами программных операционных систем, криптографические возможности которых не могут быть изменены пользователями, которые разработаны для установки пользователем самостоятельно без дальнейшей существенной поддержки поставщиком и техническая документация (описание алгоритмов криптографических преобразований, протоколы взаимодействия, описание интерфейсов и т.д.) на которые является доступной;

д) персональных смарт-карт (интеллектуальных карт), криптографиче-

ские возможности которых ограничены использованием в оборудовании или системах, указанных в подпунктах «е» – «и» настоящего пункта, или персональных смарт-карт (интеллектуальных карт) для широкого общедоступного применения, криптографические возможности которых недоступны пользователю и которые в результате специальной разработки имеют ограниченные возможности защиты хранящейся на них персональной информации;

е) приемной аппаратуры для радиовещания, коммерческого телевидения или аналогичной коммерческой аппаратуры для вещания на ограниченную аудиторию без шифрования цифрового сигнала, кроме случаев использования шифрования исключительно для управления видео- или аудиоканалами и отправки счетов или возврата информации, связанной с программой, провайдерам вещания;

ж) оборудования, криптографические возможности которого недоступны пользователю, специально разработанного и ограниченного для осуществления следующих функций:

* исполнение программного обеспечения в защищенном от копирования виде;
* обеспечение доступа к защищенному от копирования содержимому, хранящемуся только на доступном для чтения носителе информации, либо доступа к информации, хранящейся в зашифрованной форме на носителях, когда эти носители информации предлагаются на продажу населению в идентичных наборах;
* контроль копирования аудио- и видеоинформации, защищенной авторскими правами;

з) шифровального (криптографического) оборудования, специально

разработанного и ограниченного применением для банковских или финансовых операций в составе терминалов единичной продажи (банкоматов), POSтерминалов и терминалов оплаты различного вида услуг, криптографические возможности которых не могут быть изменены пользователями;

и) портативных или мобильных радиоэлектронных средств граждан-

ского назначения (например, для использования в коммерческих гражданских системах сотовой радиосвязи), которые не способны к сквозному шифрованию (то есть от абонента к абоненту);

к) беспроводного оборудования, осуществляющего шифрование ин-

формации только в радиоканале с максимальной дальностью беспроводного действия без усиления и ретрансляции менее 400 м в соответствии с техническими условиями производителя (за исключением оборудования, используемого на критически важных объектах);

л) шифровальных (криптографических) средств, используемых для защиты технологических каналов информационно-телекоммуникационных систем и сетей связи, не относящихся к критически важным объектам;

м) товаров, у которых криптографическая функция гарантированно за-

блокирована производителем.

Из документов ФСТЭК (Федеральная служба по техническому и экспортному контролю):

*Критически важный объект* – объект, оказывающий существенное влияние на национальную безопасность Российской Федерации, прекращение или нарушение функционирования которого приводит к чрезвычайной ситуации или к значительным негативным последствиям для обороны, безопасности, международных отношений, экономики, другой сферы хозяйства или инфраструктуры страны, либо для жизнедеятельности населения, проживающего на соответствующей территории, на длительный период времени.

#### 1.7. Правила сертификации СКЗИ по требованиям ФСБ

К средствам криптографической защиты, на которые могут распространяться правила сертификации относятся:

* сами средства шифрования,
* средства защиты от навязывания ложной информации (имитозащита),
* средства электронной цифровой подписи (ЭЦП),
* средства кодирования,
* средства изготовления криптографических ключей,
* сами криптографические ключи (независимо от их носителя).

Обязанность сертифицировать СКЗИ *существует для сведений, составляющих государственную тайну*. Это прямо вытекает из Постановления Правительства от 26 июня 1995 г. № 608 «О сертификации средств защиты информации» (в редакции Постановлений Правительства РФ от 23.04.1996 № 509, от 29.03.1999 № 342, от 17.12.2004 № 808).

Сертификация шифровальных средств является *обязательной* также в случаях, прямо прописанных в законодательстве.

На сегодняшний день к таким нормативно-правовым актам относится Постановление Правительства от 17 ноября 2007 г. № 781 «*Об утверждении Положения об обеспечении безопасности персональных данных при их обработке в информационных системах персональных данных*». В пункте 5 Положения говорится, что «*средства защиты информации, применяемые в информационных системах, в установленном порядке проходят процедуру оценки соответствия*». А единственной известной на сегодня процедурой оценки соответствия по линии ФСБ является сертификация.

Число сертифицированных ФСБ СКЗИ сегодня мало, поэтому ФСБ разрешило для защиты персональных данных использовать не только сертифицированные в ФСБ средства шифрования, но и имеющие *положительное заключение*.

Сферой, в которой также требуется наличие сертификата на СКЗИ, является защита критически важных объектов. Критически важный объект – это объект, оказывающий существенное влияние на национальную безопасность Российской Федерации, прекращение или нарушение функционирования которого приводит к чрезвычайной ситуации или к значительным негативным последствиям для обороны, безопасности, международных отношений, экономики, другой сферы хозяйства или инфраструктуры страны, либо для жизнедеятельности населения, проживающего на соответствующей территории, на длительный период времени (из документов ФСТЭК – Федеральной службы по техническому и экспортному контролю).

#### 1.8. Таможенные ограничения на ввоз и вывоз СКЗИ

Все шифровальные средства по таможенным правилам разделены на две группы:

1. Продукция, которая попадает в «*Перечень категорий товаров (продукции), являющихся шифровальными (криптографическими) средствами или содержащих в своем составе шифровальные (криптографические) средства, технические и криптографические характеристики которых подлежат нотификации*» (Перечень НТФ) с упрощенной процедурой ввоза.
2. Продукция с функцией шифрования, не попадающая в Перечень НТФ, ввозимая на основании *разовой лицензии Минпромторга России, выданной на основании заключения ЦЛСЗ* (центр по лицензированию, сертификации и защите государственной тайны ФСБ России) о возможности ввоза шифровального средства. Лицензия выдается на определенное количество конкретных шифровальных средств.

Действующие нормы Таможенного союза (*Республика Беларусь, Республика Казахстан и Российская Федерация в соответствии с Договором от 6 октября 2007 года*) регламентируют деятельность по ввозу и импорту оборудования в сфере информационных технологий в соответствии с Соглашениями между этими странами от 9 июня 2009 года:

* Положение о порядке ввоза на таможенную территорию Таможенного союза и вывоза с таможенной территории Таможенного союза шифровальных (криптографических) средств.
* Шифровальные (криптографические) средства, ввоз которых на таможенную территорию Таможенного союза и вывоз с таможенной территории Таможенного союза ограничен.

Единым перечнем товаров, к которым применяются запреты или ограничения на ввоз и вывоз государствами – членами Таможенного союза, в рамках Евразийского экономического сообщества в торговле с третьими странами, определен спектр средств связи, радиосвязи, компьютеров, карт, другого *оборудования, содержащего криптографию, и программного обеспечения с функцией шифрования.* Для ввоза или импорта данного оборудования требуется получение разрешительных документов.

Для ввоза шифровальных средств необходима нотификация ФСБ или лицензия Минпромторга (наименование разрешения определяется в зависимости от типа оборудования,  *например, действие нотификации ФСБ не распространяется на базо-*

*вые станции LTE, маршрутизаторы и др. оборудование соответствующего уровня криптографии. В отношении него необходимо оформить лицензию Минпромторга на импорт криптографических (шифровальных) средств)*.

***Нотификация ФСБ*** –документальное оповещение Государства о факте содержания в выпускаемой предприятием продукции различных элементов криптографического (шифровального) характера.

Нотификация ФСБ – это документ, являющийся обязательным условием при ввозе на территорию Таможенного союза или вывозе с территории Таможенного союза оборудования, содержащего в себе элементы шифрования.

Подавать заявление на лицензирование Минпромторга может только тот импортер, который уже обладает лицензией ФСБ на деятельность по передаче (продаже) шифровальных средств. Она необходима не только для импорта оборудования с криптографией, но также для деятельности компаний по продаже, монтажу, наладке и другим видам работ с указанными средствами.

Ввоз шифровальных средств без лицензии и только *на основании заключения ЦЛСЗ* предусмотрен в случаях:

* при ввозе и вывозе шифровальных средств для осуществления ремонта или замены в соответствии обязательствами по договору (контракту, соглашению);
* при временном ввозе и временном вывозе шифровальных средств в целях:
* проведения научно-технической экспертизы;
* научных исследований;
* экспонирования на выставках;

 при ввозе и вывозе шифровальных средств в целях обеспечения собственных нужд организаций без права их распространения и оказания третьим лицам услуг в области шифрования;  при транзитных перевозках шифровальных средств через территорию государств – участников Таможенного союза.

Особенности представления документов, удостоверяющих (подтверждающих) соблюдение обязательных мер (требований) указаны в *Положении о порядке ввоза на таможенную территорию Таможенного союза и вывоза с таможенной территории Таможенного союза шифровальных (криптографических) средств*, утвержденном Решением Коллегии Евразийской экономической комиссии от 16 августа 2012 г. N 134 «О нормативных правовых актах в области нетарифного регулирования».

##### Вопросы и задания для самостоятельной работы

1. На какую деятельность с использованием СКЗИ не распространяется Положение о лицензировании деятельности по разработке, производству, распространению шифровальных (криптографических) средств?
2. Что такое нотификация ФСБ?
3. При осуществлении каких работ требуется получение лицензии ФСБ?
4. В каких случаях сертификация шифровальных средств является обязательной?
5. Какие средства шифрования можно использовать для защиты персональных данных?
6. В чем состоит принципиальное отличие нового стандарта ГОСТ Р 34.10-2012 на алгоритм формирования и проверки ЭЦП от старых ГОСТ Р 34.10?
7. Какова длина подписи по ГОСТ 34.10-2012?
8. Какая криптосистема называется криптосистемой общего использования?
9. Какие требования предъявляются к хэш-функциям, используемым в КСЗИ для формирования цифровой подписи?
10. Как выглядит цифровая подпись для сообщения?
11. Перечислите КСЗИ, использующие механизм электронной цифровой подписи.
12. Каков размер блока шифруемого текста в криптосистеме «Магма» ГОСТ Р 34.12-2015?
13. Перечислите основные режимы работы, предусмотренные в стандарте шифрования данных ГОСТ Р 34.13-2015.
14. Перечислите основные режимы работы, предусмотренные в стандарте шифрования данных DES.
15. Назовите действующий стандарт шифрования в Америке.
16. Какой криптографический алгоритм лег в основу стандарта шифрования AES?
17. Что общего в стандартах DES и «Магма»?
18. Назовите размер ключа (в битах), используемого в криптосистеме «Кузнечик» стандарта ГОСТ Р 34.12-2015.
19. Почему ГОСТ28147-89 более устойчив к вскрытию путем полного перебора по множеству возможных значений ключа, чем DES?
20. Хэш-значение какой длины (в байтах) вычисляет стандарт ГОСТ Р34.11-2012?
21. Каков размер блоков сообщения при вычислении хэш-функций в новом стандарте ГОСТ Р 34.11-2012?

## ТЕМА 2. КЛАССИФИКАЦИЯ СРЕДСТВ КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ

*Средства криптографической защиты информации (СКЗИ) –* это совокупность программных и технических элементов систем обработки данных, способных функционировать самостоятельно или в составе других систем и осуществлять криптографическое преобразование информации для обеспечения ее безопасности.

Замечательная особенность СКЗИ заключается в том, что для некоторых направлений (например, защищенная электронная почта и виртуальные частные сети) у средств криптографической защиты информации *не существует альтернатив.*

Основные задачи СКЗИ:

* создание и использование *носителей ключевой* информации;
* *шифрование данных*, хранимых в базе данных или в электронном виде на различных носителях информации;
* электронная *цифровая подпись* и связанные с ней виды шифрования, в частности, проверка авторства;
* криптографические *интерфейсы*;  задачи идентификации и *аутентификации.*

**2.1. Классификация средств криптографической защиты**

### информации по различным признакам

К средствам криптографической защиты информации (СКЗИ) относятся аппаратные, программно-аппаратные и программные средства, реализующие криптографические алгоритмы преобразования информации.

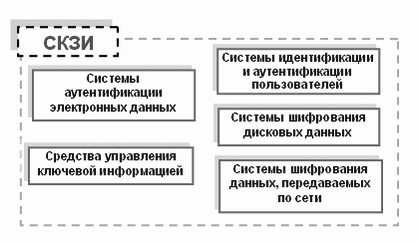
Наряду с термином «средство криптографической защиты информации» часто используется термин *шифратор*.Термин шифратор означает аппарат или программу, реализующую алгоритм шифрования. Понятие СКЗИ включает в себя шифратор, но в целом является более широким.

Классификация средств криптографической защиты информации может проводиться по различным таксономическим признаком. Таким признаком может быть вид прикладных задач защиты информации, решаемых в процессе информационного обмена, уровень защиты или способ реализации СКЗИ.

СКЗИ могут быть разработаны в виде программ, аппаратно и в программно-аппаратном комплексе. Таким образом, все СКЗИ *по способу реализации* СКЗИ можно разделить на программные, программно-аппаратные и аппаратные СКЗИ:

* *программные СКЗИ* – реализация одного или нескольких криптографических алгоритмов на языке программирования высокого или низкого уровня, в виде модулей, отдельных библиотек или выделенных программ с функцией криптографической защиты;
* *аппаратные СКЗИ –* реализация криптографических алгоритмов или их отдельные участки в микросхемах, процессорах и специализированных блоках (системы встроенной защиты) и аппаратных модулях (системы наложенной защиты), совмещенные со средствами вычислительной техники или встраиваемые в автоматизированные системы;
* *программно-аппаратные СКЗИ* – комплексы, состоящие из взаимосвязанной программной и аппаратной части с функциями криптографической защиты.

Независимо от способа реализации все СКЗИ, обеспечивающие высокий уровень защиты информации, можно разделить на пять основных групп в соответствие с прикладными задачами по криптографической защите (рис. 2.1).



***Рис. 2.1.****Основные группы СКЗИ*

Первую группу образуют *системы идентификации* и *аутентификации пользователей*. Такие системы применяются для ограничения доступа случайных и незаконных пользователей к ресурсам компьютерной системы. Общий алгоритм работы этих систем заключается в том, чтобы получить от пользователя информацию, удостоверяющую его личность, проверить ее подлинность и затем предоставить (или не предоставить) этому пользователю возможность работы с системой.

Вторую группу средств, обеспечивающих повышенный уровень защиты, составляют *системы шифрования дисковых данных*. Основная задача, решаемая такими системами, состоит в защите от несанкционированного доступа к данным, расположенным на дисковых носителях.

Обеспечение конфиденциальности данных на дисковых носителях обычно осуществляется путем их шифрования с использованием симметричных алгоритмов шифрования. Основным классификационным признаком для комплексов шифрования служит уровень их встраивания в компьютерную систему.

*По уровню объекта шифрования* СКЗИ можно разделить на две группы:

* СКЗИ, шифрующие на уровне файлов (защищаются отдельные файлы) – *архиваторы типа WinRAR;*
* СКЗИ, шифрующие на уровне дисков (защищаются диски целиком) – *программа шифрования Diskreet пакета Norton Utilities.*

*По способу функционирования* системы шифрования дисковых данных делят на:

* системы «прозрачного» шифрования,  *пример: шифрование папки Temp и Мои документы при использовании EFS Win2000;*
* системы, специально вызываемые для осуществления шифрования, *пример: архиваторы со встроенными средствами парольной защиты.*

К третьей группе средств, обеспечивающих повышенный уровень защиты, относятся *системы шифрования данных, передаваемых по компьютерным сетям*. Можно выделить два основных способа шифрования таких данных и разделить СКЗИ по этим способам:

* канальное шифрование (защищается вся передаваемая по каналу связи информация, включая служебную);
* оконечное (абонентское) шифрование (защищенным оказывается только содержание сообщения, вся служебная информация остается открытой).

Четвертую группу средств защиты составляют *системы аутентификации электронных данных*.

При обмене электронными данными по сетям связи возникает проблема аутентификации автора документа и самого документа, т.е. установление подлинности автора и проверка отсутствия изменений в полученном документе.

Для аутентификации электронных данных применяют код аутентичности сообщения (имитовставку) или электронную цифровую подпись. При этом используются разные типы систем шифрования – симметричные и асимметричные.

Пятую группу средств, обеспечивающих повышенный уровень защиты, образуют *средства управления ключевой информацией*. Под ключевой информацией понимается совокупность всех используемых в компьютерной системе или сети криптографических ключей.

Распределение ключей является самым ответственным процессом в управлении ключами. Этот процесс должен гарантировать скрытность распределяемых ключей, а также оперативность и точность их распределения. Различают два основных *способа распределения ключей* между пользователями компьютерной сети:

 применение одного или нескольких центров распределения ключей;  прямой обмен сеансовыми ключами между пользователями.

### 2.2. Требования к средствам криптографической защиты информации

Требования к современным СКЗИ, в сущности, не отличаются от сформулированных требований к криптосистемам в разделе 1.4 и 2.7. Они не зависят от принадлежности к какой-либо из рассмотренных выше групп и способов реализации. Они общие для всех рассмотренных криптосистем и СКЗИ на их основе.

1. Зашифрованное сообщение должно поддаваться чтению только при наличии ключа, попытка же чтения без предварительного знания ключа должна быть необходимо сопряжена с вычислительно сложной задачей, время решения которой на современной компьютерной технике превышает время жизни защищаемой информации.
2. Число операций, необходимых для определения использованного ключа шифрования по фрагменту шифрованного сообщения и соответствующего ему открытого текста, должно быть не меньше общего числа возможных ключей. При этом следует учитывать, что на практике при лобовой атаке криптоаналитику приходится перебрать, как правило, только половину всех возможных ключей.
3. Число операций, необходимых для расшифровывания информации путем перебора всевозможных ключей, должно иметь строгую нижнюю оценку и выходить за пределы возможностей современных компьютеров (с учетом возможности использования распределенных вычислений).
4. Знание алгоритма шифрования не должно влиять на надежность защиты (принцип Кирхгофа).
5. Незначительное изменение ключа должно приводить к существенному изменению вида зашифрованного сообщения – так называемый принцип распространения ошибки.
6. Структурные элементы алгоритма шифрования должны быть неизменными, т.е. должен быть реализован их контроль целостности.
7. Дополнительные биты, вводимые в сообщение в процессе шифрования (например, при дополнении открытого текста до длины, кратной длине блока алгоритма шифрования), должны быть полностью и надежно скрыты в шифрованном тексте.
8. Длина шифрованного текста должна быть равной длине открытого текста.
9. Не должно быть простых и легко устанавливаемых зависимостей между ключами, последовательно используемыми в процессе шифрования.
10. Любой ключ из множества возможных должен обеспечивать надежную защиту информации, т.е. из ключевого множества должны быть исключены заведомо слабые для данного алгоритма ключи.
11. Алгоритм должен допускать как программную, так и аппаратную реализацию, при этом изменение длины ключа не должно вести к качественному ухудшению алгоритма шифрования.

### 2.3. Программные СКЗИ. Особенности и примеры

Программные СКЗИ – это средства защиты информации, в которых реализация одного или нескольких криптографических алгоритмов выполнена на языке программирования высокого или низкого уровня, в виде модулей, отдельных библиотек или выделенных программ с функцией криптографической защиты.

Основным достоинством программных СКЗИ является их *дешевизна и гибкость*. Основной недостаток программных СКЗИ связан с основным достоинством – возможность легкой модификации. Программа может быть достаточно просто модифицирована злоумышленником. Для устранения угрозы модификации требуется организация проверки целостности.

С точки зрения защиты информации программные средства с криптографическими функциями для универсальных ЭВМ более уязвимыми, чем специализированные шифраторы, еще и потому, что при создании всевозможных текстовых редакторов, СУБД, коммуникационных программ, архиваторов и т.д. их разработчики в первую очередь руководствуются принципом максимального удобства для пользователя и принципом безотказного функционирования, а вопросы гарантированной защиты отодвигаются на второй и даже третий план.

Принципы безотказного функционирования и удобства программных продуктов диктуют необходимость введения различных видов избыточности, в частности, таких понятий, как формат носителя данных и формат файла. Как следствие использование форматов приводит к ослаблению криптографической схемы. Кроме того, алгоритмы криптографической защиты реализуются не только специалистами-криптографами, но и программистами, не имеющими специальной подготовки в области криптографии, можно отметить относительно невысокую стойкость большинства широко распространенных программных средств с криптографическими функциями.

Основными причинами низкой криптостойкости, как правило, являются следующие:

* применение нестойких криптоалгоритмов;
* слабость ключевой системы;
* ошибки в проектировании программ.

Наиболее опасными и трудноустранимыми являются ошибки проектирования – зачастую подсистема криптографической защиты информации не проектируется изначально, а добавляется к уже существующему продукту, что не позволяет учесть все возможные проблемы применения криптографии.

Например, встроенное средство шифрования документов Microsoft Word XP реализует стойкий криптоалгоритм и достаточно корректную ключевую систему, однако из-за того, что формат хранения документов предусматривает возможность хранения нескольких версий одного и того же документа в одном файле (зашифрованных одним ключом), криптоаналитик может восстановить ключ шифрования.

Серьезной проблемой программных СКЗИ является использование оперативной памяти системы для операций с криптографическим ключом – конечный промежуток времени криптографический ключ присутствует в памяти в открытом виде и может быть из нее извлечен.

Еще одна проблема связана скорее с недостатками программирования, а не со спецификой программных СКЗИ, например, неаккуратное использование временных файлов, при котором в них может оставаться ценная для криптоанализа информация.

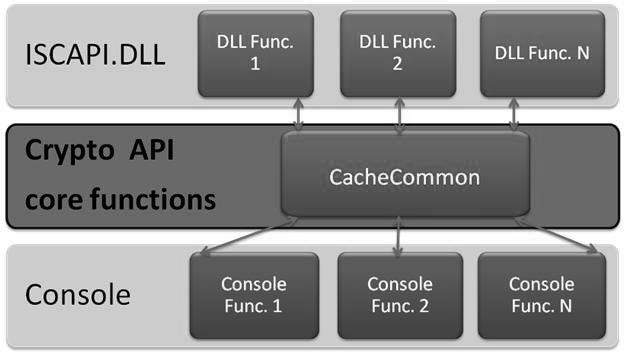
Серьезной проблемой программных СКЗИ является генерация случайных чисел, используемых для формирования секретного ключа. Часто для генерации ключевого материала используются показания системных часов, данные из оперативной памяти и прочая псевдослучайная информация. Следует помнить, что действительно случайными являются только некоторые физические процессы и человеческий фактор (см. ч. 1 п. 8.3). Генерируемая последовательность, основанная только на состоянии ЭВМ, не может быть признана истинно случайной и, вполне вероятно, подчиняется некоторой закономерности, а следовательно, при ее использовании может быть получен слабый ключ.

В некоторых современных компьютерах имеется встроенный аппаратный датчик случайных чисел, однако в контексте использования программных СКЗИ следует помнить, что этот датчик доступен ОС, поэтому для его гарантированной стойкости необходимо использование доверенной ОС.

Начиная с 2000 года программные СКЗИ встраиваются в операционные системы персональных компьютеров. Например, в MS Windows, начиная с Windows 2000, предусмотрена система шифрования файлов на NTFS-дисках. Это прозрачное шифрование, технология которого основана на сертификатах открытых ключей.

Другим примером встроенных СКЗИ является криптографический интерфейс прикладных программ компании Microsoft CryptoAPI (Cryptographic Application Programming Interface) (рис. 2.2). Входит в состав операционных систем Microsoft. Большинство функций CryptoAPI поддерживается начиная с Windows 2000.

CryptoAPI представляет собой набор функций, предназначенных для работы различных криптографических сервисов (шифрования, хэш-функций, ЭЦП, проверки сертификатов). С помощью функций CryptoAPI можно создавать прикладное ПО, призванное решать криптографические задачи различной степени сложности. Особо стоит отметить, что CryptoAPI содержит только описание функций криптографических примитивов; непосредственная их реализация содержится в отдельной библиотеке, называемой *криптопровайдером*.



***Рис. 2.2.*** *Схема работы CryptoAPI*

Криптопровайдер, или Cryptographic Service Provider (CSP), – это специализированное программное обеспечение (ПО), предназначенное для решения криптографических задач в семействе операционных систем Windows. Криптопровайдер представляет собой независимую динамическую библиотеку (DLL) криптографических алгоритмов, доступных прикладным программистам посредством интерфейса CryptoAPI. При взаимодействии с любым криптопровайдером приложения вызывают функции CryptoAPI, которые обращаются к системным библиотекам операционной системы, одна из которых фильтрует вызовы этих функций и передает их далее соответствующим функциям криптопровайдера через CryptoSPI (Cryptographic System Program Interface).

Основные криптографические сервисы, как контроль целостности, аутентификация пользователей и данных, шифрование дисковых данных и канальное шифрование, можно реализовать, используя базовые криптографические примитивы: хеш-функции, схемы шифрования и электронно-цифровой подписи. Это делает обоснованным иерархический подход к созданию СКЗИ: на базе основного устройства (аппаратного или программного), реализующего основные криптографические примитивы. Посредством использования библиотек функций создаются прикладные программные продукты, осуществляющие дисковое (в т.ч. прозрачное), абонентское, канальное шифрование, ЭЦП, аутентификацию пользователей и данных. Такой подход реализован, например, в разработках фирмы «Анкад», где основным устройством является программный эмулятор платы Crypton Emulator в совокупности с программными средствами Crypton ArcMail, Crypton Disk, Crypton IPMobile. Схема работы изображена на рис.

2.3.

*«Crypton IPMobile»* предназначен для организации виртуальной частной сети c шифрованием сетевого трафика по алгоритму ГОСТ 28147-89.

Средство *«Crypton ArcMail»* обеспечивает конфиденциальность, проверку авторства и целостности файлов, каталогов и областей памяти. «Crypton ArcMail» в едином сервисе предоставляет функции архивирования, электронной цифровой подписи (ЭЦП) и шифрования. «Crypton ArcMail» создает подписанный и/или зашифрованный архив, который можно отправлять адресату (адресатам) по открытым каналам связи, в т.ч. по сети Интернет.

Средство *«Crypton Disk»* подключается к шифратору и позволяет создавать секретные логические диски, содержимое которых шифруется в режиме и доступно только для владельца диска. При чтении какой-либо программной информации с секретного диска эта информация расшифровывается, а при записи – зашифровывается. Секретный диск существует в виде файла-контейнера (можно запретить незарегистрированным пользователям случайное или преднамеренное уничтожение файлов-контейнеров); для законного владельца секретный диск предстает в виде еще одного логического диска. Crypton Disk позволяет пользователям закрывать доступ ко всем логическим дискам по истечении установленного времени и/или комбинации горячих клавиш.

Растет число программных СКЗИ, создающихся в рамках концепции open source*(open-source software* – программное обеспечение с открытым исходным кодом). В первую очередь к ним следует отнести программный комплекс Филипа Циммермана – PGP (Pretty Good Privacy). PGP – программное СКЗИ, позволяющее (в зависимости от версии функциональность может различаться) шифровать данные (файловое шифрование или создание защищенного диска – PGP-диск), подписывать сообщения и управлять ключами. Проект PGP в настоящее время принадлежит Network Associates.

Таким образом, учитывая уязвимость программных СКЗИ, к программным системам защиты информации следует относиться с особой осторожностью; тем не менее программные СКЗИ весьма эффективны для защиты информации, не содержащей государственной тайны.

Примеры программных СКЗИ в соответствие со схемой классификации рис. 2.1 сведены в табл. 2.1.

*Таблица 2.1*

|  |  |
| --- | --- |
| Группа СКЗИ | Примеры |
| Системы идентификации и аутентификации пользователей | Встроенные средства ОС |
| Системы шифрования дисковых данных | Встроенные средства ОС, PGP, Secret Disk |
| Системы шифрования данных, передаваемых по сетям | Комплексы ЗАСТАВА (VPN + программное  СКЗИ) |
| Системы аутентификации электронных данных | PGP |
| Средства управления ключевой информацией | Сервер сертификатов ЗАСТАВА |

### 2.4. Аппаратные и программно-аппартные СКЗИ. Особенности и примеры

Аппаратные СКЗИ – это средства защиты информации, в которых реализация криптографических алгоритмов или их отдельные участки выполнены в микросхемах, процессорах и специализированных блоках (системы встроенной защиты) и аппаратных модулях (системы наложенной защиты), совмещенных со средствами вычислительной техники или встраиваемых в автоматизированные системы.

Программно-аппаратные СКЗИ – это комплексы, состоящие из взаимосвязанной программной и аппаратной части с функциями криптографической защиты.

Историческим примером аппаратных СКЗИ являются шифраторы докомпьютерной эры. Это табличка Энея (рис. 1.3, ч. 1), шифровальный диск Альберти (рис. 6.2, ч. 1) и, наконец, дисковые шифрующие машины (приложение 3). Самым видным представителем последних стал шифратор времен Второй мировой войны Enigma.

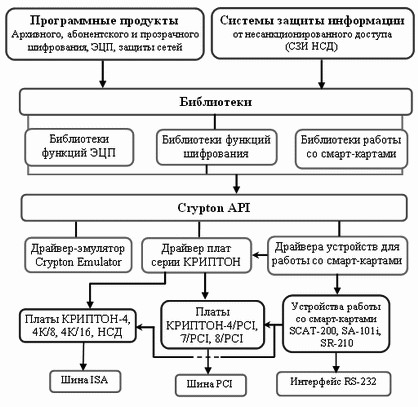
Современные аппаратные СКЗИ нельзя строго отнести к аппаратным, строго говоря, это программно-аппаратные СКЗИ: программные – потому что управление процессом реализуется программно; аппаратные – потому что программная часть неподконтрольна операционной системе (ОС) и реализована аппаратно (за счет создания и применения специализированных процессоров) основных криптографических функций – криптографических преобразований, управления ключами, криптографических протоколов и т.д.

Программно-аппаратные средства криптографической защиты информации сочетают гибкость программного решения с надежностью аппаратного. При этом за счет гибкой программной оболочки можно быстро менять пользовательский интерфейс, конечные функции продукта, производить его конечную настройку; а аппаратная компонента позволяет защитить от модификации алгоритм криптографического примитива, обеспечить высокую защищенность ключевого материала и зачастую более высокую скорость работы.

Использование аппаратных средств снимает проблему обеспечения целостности системы. В большинстве современных систем защиты от НСД применяется зашивка программного обеспечения в ПЗУ или в аналогичную микросхему. Таким образом, для внесения изменений в ПО необходимо получить доступ к соответствующей плате и заменить микросхему. В случае использования универсального процессора реализация подобных действий потребует применения специального оборудования, что еще более затруднит проведение атаки. Использование специализированного процессора с реализацией алгоритма работы в виде интегральной микросхемы полностью снимает проблему нарушения целостности этого алгоритма.

На практике зачастую функции аутентификации пользователя, проверки целостности, криптографические функции, образующие ядро системы безопасности, реализуются аппаратно, все остальные функции – программно. Ниже приведены примеры программно-аппаратных СКЗИ.

Например, в программно-аппаратных средствах фирмы «Анкад» основным устройством являются платы *«КРИПТОН»*. Платы используются в совокупности с программными средствами, представленными выше, такими как Crypton ArcMail, Crypton Disk, Crypton IPMobile. Схема работы данного средства изображена на рис. 2.3.



***Рис. 2.3.****Структура средств криптографической защиты информации фирмы «Анкад»*

Примеры СКЗИ компании «Код Безопасности»:

Secret Net – сертифицированное средство защиты информации от несанкционированного доступа для операционных систем семейства MS Windows.

СКЗИ М-506А-XP – это программно-аппаратный комплекс для защиты информации под управлением ОС MS Windows 2000/XP/2003, *шифрует* по ГОСТ 28147-89 *в режиме гаммирования*.

Основные возможности СКЗИ М-506А-XP по защите информации:

* защита информации от несанкционированного доступа;
* шифрование сетевого трафика;
* доверенная загрузка (защита от загрузки с внешних носителей);
* шифрование локальных ресурсов;
* контроль аппаратной конфигурации компьютера;
* контроль потоков конфиденциальной информации.

Достоинства СКЗИ М-506А-XP:

* наличие сертификата ФСБ и защита информации, составляющей государственную тайну;
* объединение функций криптографии и возможностей Secret Net;  клиент-серверная архитектура.

#### Защита информации от несанкционированного доступа

СКЗИ М-506А-XP реализовано на основе системы Secret Net, поэтому оно содержит все механизмы защиты информации от несанкционированного доступа, средства управления и регистрации, которые есть в этом продукте:

* доверенная загрузка (защита от загрузки с внешних носителей);
* идентификация и аутентификация пользователей;
* разграничение доступа к устройствам;
* контроль печати конфиденциальной информации;
* полномочное управление доступом;
* контроль целостности программ и данных;
* гарантированное уничтожение данных;
* контроль аппаратной конфигурации компьютера;  функция электронной цифровой подписи файлов;  шифрование файлов. СКЗИ М-506А-XP позволяет *шифровать файлы на локальных ресурсах*.

#### Управление безопасностью

СКЗИ М-506А-XP имеет архитектуру «клиент-сервер», это позволяет централизованно управлять безопасностью огромного количества компьютеров:

* регистрация событий рабочих станций;
* контроль действий сотрудников и администраторов;  контроль потоков конфиденциальной информации;  механизм отчетов.

Примеры аппаратных и аппаратно-программных СКЗИ в соответствие со схемой классификации рис. 2.1 сведены в табл. 2.2.

*Таблица 2.2*

|  |  |
| --- | --- |
| Группа СКЗИ | Примеры |
| Системы идентификации и аутентификации пользователей | АККОРД |
| Системы шифрования дисковых данных | Crypton Soft, ГРЯДА |
| Системы шифрования данных, передаваемых по сетям | Crypton ArcMail (базовое устройство – плата  КРИПТОН) |
| Системы аутентификации электронных данных | Crypton Sign, Crypton ArcMail (базовое устройство – плата КРИПТОН) |
| Средства управления ключевой информацией | Crypton Tools (базовое устройство – плата  КРИПТОН) |

### 2.5. Критерии выбора СКЗИ

На сегодняшний день пользователи средств защиты информации, реализующих криптографические алгоритмы, осуществляют выбор СКЗИ на основании ряда критериев, среди которых можно выделить:

* надежность криптоалгоритмов (длина ключа, стойкость алгоритма);
* легальность использования в соответствии с существующими стандартами и нормативно-правовой базой;
* наличие сертификатов государственных органов;  экспортно-импортные ограничения.

Кроме выбора алгоритма, не менее остро стоит проблема выбора способа реализации: аппаратная (аппаратно-программная) или чисто программная. Основным критерием здесь становится стоимость защищаемой информации и величина бюджета компании на цели приобретения и эксплуатации СКЗИ. Программные СКЗИ более дешевые, гибкие, но менее надежные. Аппаратные и программно-аппаратные средства надежны, но дорогие и сложные в плане модернизации.

Для обобщенной таблицы российского рынка СКЗИ можно составить следующие критерии.

* Фирма-производитель. Согласно общедоступным данным (Интернет), в России на данный момент порядка 20 компаний-разработчиков СКЗИ.
* Тип реализации (аппаратная, программная, аппаратно-программная). Обязательное разделение, которое имеет, тем не менее, весьма нечеткие границы, поскольку существуют, например, СКЗИ, получаемые путем установки некоторой программной составляющей – средств управления и непосредственно криптобиблиотеки, и в итоге они позиционируются как аппаратно-программное средство, хотя на самом деле представляют собой только ПО.
* Наличие действующих сертификатов соответствия ФСБ России и классы защиты. Обязательное условие для российского рынка СКЗИ, более того, 90% решений будут иметь одни и те же классы защиты.
* Реализованные криптографические алгоритмы (указать ГОСТы). Также обязательное условие – наличие ГОСТ 28147-89.
* Поддерживаемые операционные системы. Достаточно спорный показатель, важный для программно реализованной криптобиблиотеки и совершенно несущественный для чисто аппаратного решения.
* Предоставляемый программный интерфейс. Существенный функциональный показатель, одинаково важный как для «технического», так и «потребительского» взгляда.
* Наличие реализации протокола SSL/TLS. Однозначно «технический» показатель, который можно расширять с точки зрения реализации иных протоколов.
* Поддерживаемые типы ключевых носителей. «Технический» критерий, который дает весьма неоднозначный показатель для различных типов реализации СКЗИ -аппаратных или программных.
* Интегрированность с продуктами и решениями компании Microsoft, а также с продуктами и решениями других производителей. Оба критерия больше относятся к программным СКЗИ типа «криптоби-блиотека», при этом использование этих критериев, например, для аппаратного комплекса построения VPN представляется весьма сомнительным.
* Наличие дистрибутива продукта в свободном доступе на сайте производителя, дилерской сети распространения и сервиса поддержки (временной критерий). Все эти три критерия однозначно являются «потребительскими», причем выходят они на первый план только тогда, когда конкретный функционал СКЗИ, сфера применения и круг решаемых задач уже предопределены.

Таким образом, во-первых, выбор СКЗИ изначально должен отталкиваться от сферы применения, что значительно сужает спектр возможных решений.

Во-вторых, «технический» и «потребительский» взгляды на СКЗИ не должны вступать в противоречия, наличие уникальных функциональных возможностей СКЗИ не должно превалировать над здравым смыслом при выборе компании-производителя с широкой сетью распространения продукта, доступной ценовой политикой и адекватным сервисом технической поддержки решения.

### 2.6. Основные принципы построения СКЗИ

С ростом сложности средств связи и информационных технологий возрастает и сложность обеспечения безопасности с помощью СКЗИ. Об этом говорят как западные исследования, проводимые с начала 80-х годов, так и отечественные работы настоящего времени. Основные трудности связаны со следующими факторами:

* средство реализации криптографического алгоритма в компьютерной системе представляет собой равноправный с прочими ресурс (является программой и использует данные компьютерной системы);
* ключевая информация СКЗИ является данными компьютерной системы с возможностью доступа со стороны других программ и с прохождением при обработке также через ряд внешних по отношению к СКЗИ программных модулей;
* функционирование СКЗИ происходит не автономно, а выполняется под управлением операционной системы и различных программ-посредников, которые, при желании, могут произвольно искажать вводимую и выводимую СКЗИ информацию;
* программная среда, в которой работает СКЗИ, устроена иерархично, т.е. для выполнения типовых функций все программы используют одни и те же фрагменты кода и данные;
* работа СКЗИ сопряжена с возникновением ошибочных ситуаций в аппаратной и программной среде компьютерной системы.

В связи с этим для обеспечения безопасности информации в современных информационно-телекоммуникационных системах, основанных на передовых информационных технологиях, необходимо эффективно решать следующий круг сложных научно-технических задач:

* обеспечивать оптимальную, формально проверяемую реализацию криптографических алгоритмов в рамках эксплуатируемых в информационнотелекоммуникационных системах программных и аппаратных платформ;
* обеспечивать при проектировании СКЗИ меры обеспечения отказоустойчивости, защиты от сбоев и искажений аппаратной компоненты;
* обеспечивать защищенность СКЗИ и его ресурсов (ключевой информации и др.) от несанкционированного доступа со стороны других программ;
* гарантировать качество управления СКЗИ со стороны операционной системы и программ-посредников, в том числе и в условиях ошибочных и преднамеренных действий пользователей.

Следует также отметить, что реализация СКЗИ в сложных универсальных операционных средах типа Windows или Unix требует проведения значительных объемов поисковых исследований для определения точек встраивания СКЗИ в операционную систему и обеспечения корректности их работы.

Разработка всякого технического средства начинается с формулирования технического задания (ТЗ). Специфика криптографических средств защиты информации определяет следующие обязательные разделы ТЗ:

1. точное назначение данных СКЗИ;
2. реализуемые СКЗИ функции;
3. требования по уровню защиты информации;
4. требования по производительности криптографических преобразований;
5. требования по защите от НСД (инженерной криптографии);
6. реальные условия эксплуатации;
7. специальные требования;
8. требования к программному обеспечению. Важнейшим этапом построения СКЗИ является тестирование.

**2.7. Принципы построения аппаратных СКЗИ**

Концепция построения аппаратных СКЗИ ставит две основных цели:

* максимальное повышение уровня защиты информации и защиты от

НСД;

* максимальное увеличение быстродействия криптопреобразований.

Для повышения уровня защиты информации и защиты от НСД могут быть использованы следующие методы инженерной криптографии:

1. обеспечение минимально возможного взаимодействия прикладного ПО и ОС персонального компьютера (ПК) с аппаратными средствами (АС);
2. применение резервированного (дублированного) генератора случайных чисел для формирования физических ключей заданной длины;
3. загрузка исполняемых криптографических алгоритмов из энергонезависимой памяти во встроенное ОЗУ (RAM) микропроцессора в шифрованном виде;
4. исполнение криптографических алгоритмов только из встроенного ОЗУ микропроцессора (с предварительным расшифрованием непосредственно в данном ОЗУ);
5. хранение шифрованных данных в ЭНП в структурированном виде с применением различных методов контроля (например, контрольное суммирование загружаемых программных модулей) для обеспечения контроля достоверности и целостности информации;
6. передача данных (ключевой информации, загружаемых криптографических алгоритмов, результатов промежуточных вычислений и т.д.) по внутренним шинам АС только в шифрованном виде;
7. «прошивка» отдельных ключей (например, ключа, который используется при инициализации АС) в «железе» с обеспечением гарантированной невозможности извлечения;
8. применение «оперативно загружаемой схемотехники» в аппаратных ускорителях;
9. использование внешних ключей, загружаемых пользователем по отдельному интерфейсу со специальных внешних накопителей (электронная карточка ридера, например);
10. применение в АС специального защитного экрана, обеспечивающего защиту от НСД самого АС (при необходимости);
11. включение в состав АС аппаратного расширения BIOS ПК, позволяющего контролировать (до загрузки ОС) целостность ОС и прикладного ПО (например, физически расположенного на жестком диске ПК).

Важным требованием является возможность полного самотестирования СКЗИ, выполняемого каждый раз после аппаратного (по включению питания) или программного сброса. Алгоритмы самотестирования должны быть встроенными в СКЗИ.

Следует отметить, что современные СКЗИ могут быть адаптированы к требованиям конечного пользователя за счет перепрограммирования и/или расширения системного ПО (BIOS), встроенного в СКЗИ.

Специальные требования накладывают ограничения на конструктивное исполнение аппаратных СКЗИ (топологию печатной платы и т.д.) в части: 1) допустимого уровня электромагнитных помех, излучаемых СКЗИ;

1. помехозащищенности СКЗИ;
2. защиты от электростатического разряда;
3. защиты от внешних электромагнитных полей; 5) защиты от электрических перегрузок;

6) защиты от тепловых перегрузок (перегрева).

Эти требования должны быть сформулированы в техническом задании на СКЗИ, а их выполнение проконтролировано при приеме в эксплуатацию.

Повышение надежности АСКЗИ может быть достигнуто за счет «горячего» резервирования (дублирования/троирования) СКЗИ. Практическая целесообразность такого резервирования определяется свойствами защищаемой информации. Возможны два варианта реализации резервирования:

* установка 2/3 одноканальных плат СКЗИ в соответствующие свободные слоты шины ISA/PCI/ USB персонального компьютера;
* разработка специального конструктивного решения СКЗИ со встроенными двумя-тремя каналами резервирования.

Недостатком в обоих случаях является, как правило, не резервированный ПК, в который устанавливается СКЗИ.

### 2.8. Принципы построения программных и программно-аппаратных СКЗИ

Существуют два основных метода проектирования – метод снизу вверх и метод сверху вниз. Остальные описываемые методы по своей сути сводятся к этим двум или являются их сочетанием.

*Метод «снизу вверх»* предполагает проектирование, начинающееся с основного аппаратного оборудования системы. При проектировании все модули системы разбиваются на ряд слоев, причем нулевой слой образует аппаратура. Каждый следующий слой последовательно добавляет новые функции, используя уже реализованные в предыдущих слоях. На самом верхнем слое должна достигаться функциональность, полностью отвечающая всем требованиям, поставленным перед разработчиком.

К недостаткам метода проектирования снизу вверх относят:

* необходимость с самого начала принимать решение о выборе способа реализации компонент СКЗИ – с помощью аппаратуры, микропрограмм или программ, что сделать очень трудно;
* возможность проектирования программной части только после разработки аппаратуры;
* расхождение между конечным продуктом и определенным в ТЗ.

При разработке программ, реализующих криптографические алгоритмы, данный метод применим только отчасти и только в тех случаях, если система команд процессора, на котором будет работать проектируемая программа, имеет в своем составе специфичные команды, реализующие элементарные базовые криптографические преобразования, либо если предполагается использовать уже готовое аппаратное обеспечение (разработанное ранее, приобретенное готовое или, быть может, регламентированное требованиями ТЗ).

При использовании *метода проектирования «сверху вниз» (иерархический метод)* исходят от того набора пользовательских функций, которые должны быть реализованы в разрабатываемой системе, и последовательно, выделяя отдельные слои, опускаются вплоть до аппаратуры. В этом случае процесс проектирования заключается в следующей последовательности.

Определяется абстракция описания компонент СКЗИ высшего уровня.

Далее систематически проводится анализ, достаточно ли определены компоненты, чтобы можно их было реализовать, используя некоторые примитивные понятия. Если нет, то каждая функция каждой компоненты представляется функциями компонент следующего слоя, которому соответствует более низкий уровень абстракции, и снова проводится анализ на возможность их реализации. В иерархическом методе целесообразно использовать принцип модульного проектирования и структурный принцип.

*Принцип модульного проектирования* заключается в разделении программ на функционально самостоятельные части (модули), обеспечивающие заменяемость, кодификацию, удаление и дополнение составных частей.

Преимущества использования модульного принципа состоят в следующем:

* упрощается отладка программ, т.к. ограниченный доступ к модулю и однозначность его внешнего проявления исключает влияние ошибок в других, связанных с ним, модулях на его функционирование;
* обеспечивается возможность организации совместной работы коллективов разработчиков, т.к. каждый программист имеет дело с независимой от других частью программы;
* повышается качество программы, т.к. относительно малый размер модулей и, как следствие, небольшая сложность их позволяют провести более полную проверку программы.

*Структурный принцип* имеет фундаментальное значение и составляет основу большинства реализаций. Согласно этому принципу, для построения программы требуются только три основных составляющих блока:

* функциональный блок;
* конструкция обобщенного цикла;
* конструкция принятия двоичного решения.

Структурный принцип формализует процесс проектирования, позволяя постепенно продвигаться от более абстрактных функциональных блоков к более конкретным, до тех пор, пока каждый из них может быть реализован с помощью имеющихся функций либо языка программирования, либо операционной системы, либо с помощью аппаратных средств.

### 2.9. Основные подходы к обеспечению надежности СКЗИ

В современных условиях при СКЗИ одним из ключевых моментов программной или программно-аппаратной реализации алгоритма является обеспечение его надежности.

*Надежность –* это свойство объекта сохранять во времени значения всех параметров, характеризующих способность выполнять требуемые функции в заданных режимах и условиях применения, технического обслуживания, ремонта, хранения и транспортировки.

Между надежностью аппаратных средств и программного обеспечения имеется принципиальное различие. Программа в большинстве случаев не может отказать случайно. Ошибки в программном обеспечении, допущенные при его создании, зависят от технологии, от организации и квалификации исполнителей и, в принципе, не являются функцией времени.

Имеющийся опыт разработки программных шифраторов показывает, что любые, даже самые несущественные на первый взгляд, ошибки приводят к нежелательным последствиям. Поэтому для современных программных систем обязательным требованием становится не просто реализация технического задания или, в частном случае, алгоритма, а его реализация с надлежащим качеством.

Существуют два основных общепринятых подхода к обеспечению качества СКЗИ от угрозы отказа функционирования, которые применимы, в том числе, и к программам, реализующим криптографические алгоритмы. Это отказоустойчивость (fault) и предотвращение неисправностей (fault avoidance).

Отказоустойчивость предусматривает, что ошибки, которые не удалось выявить на этапе разработки и тестирования СКЗИ, обнаруживаются во время работы программы и парируются за счет использования программной, информационной и временной избыточности. Предотвращение неисправностей связано с анализом природы ошибок, возникающих на разных фазах этапа разработки СКЗИ, и причин их возникновения.

Из-за невозможности обеспечения абсолютной надежности программных и программно-аппаратных СКЗИ при разработке, даже при отсутствии злоумышленных воздействий, разработчики используют методы оперативного обнаружения дефектов при выполнении программ и искажений данных путем введения в них временной, информационной и программной избыточности. Эти же виды избыточности используются для оперативного восстановления искаженных данных.

Временная избыточность состоит в использовании некоторой части производительности ЭВМ для контроля исполнения программ и восстановления (рестарта) вычислительного процесса.

Так, например, в состав СКЗИ может быть введена процедура периодического контроля. Для борьбы со сбоями возможно применить процедуры периодического контроля, которые сводятся к выполнению предопределенных детерминированных тестовых процедур (для функций шифрования это может быть, например, прогон тестовых примеров).

Информационная избыточность состоит в дублировании накопленных исходных и промежуточных данных, обрабатываемых программами. Избыточность используется для обеспечения достоверности данных при проведении криптографических преобразований и обычно реализуется в виде вычисления имитовставок.

Программная избыточность используется для контроля и обеспечения достоверности наиболее важных решений по управлению и обработке информации. Она заключается в сопоставлении результатов обработки одинаковых исходных данных разными программами и исключения искажения результатов, обусловленных различными аномалиями.

Вместе с тем, обеспечение отказоустойчивости программной реализации СКЗИ, хотя и направлено в целом на повышение надежности, не может решить такую проблему, как обеспечение соответствия программной реализации криптографическому алгоритму. Для решения этой проблемы используют методы предотвращения неисправностей.

Основные подходы к предотвращению неисправностей связаны с анализом природы возникновения ошибок, возникающих при разработке программных и программно-аппаратных СКЗИ, что в свою очередь подразумевает анализ самого процесса разработки.

Процесс разработки СКЗИ принято описывать моделями жизненного цикла программ различных классов и назначения. В модели жизненный цикл структурируется рядом крупных фаз или этапов, каждый из которых характеризуется достаточно определенными целями и результатами. Анализ существующих наработок в данной области позволяет сделать вывод о том, что основными фазами создания СКЗИ являются:

 анализ и спецификация требований;  проектирование;  исполнение.

Рассмотрим содержание каждой из фаз в контексте решения задачи предотвращения неисправностей СКЗИ.

Фаза анализа и спецификации требований

Анализ всей совокупности требований к системе – технического задания – выполняется на начальной фазе создания программ. ТЗ составляется на основании перечня требований, предъявленных к системе заказчиком (классы решаемых задач, их характеристики и особенности, режим работы автоматизированной системы, сопряжение с внешними объектами, пропускная способность, время ответа и т.п. при заданных ограничениях на стоимость, длительность разработки и др.). Цель создания ТЗ – уточнить и сформулировать задачи, возлагаемые на систему, согласовать требования заказчика и возможности исполнителя, составить техническое задание на разработку СКЗИ. Это делается для того, чтобы удостовериться в том, что от программ требуются только те системные требования, которые могут быть достигнуты.

Достоинства формальных методов заключается в том, что системы, разработанные с использованием подобного подхода, имеют принципиально высокое качество. При этом повышение качества достигается двумя путями:

 построением спецификаций в виде ясного, исчерпывающего, недвусмысленного и легкого для проверки математического утверждения;  осуществлением верификации во время разработки СКЗИ.

Разработка формальной спецификации требует значительных усилий. Однако, как показывает практика, большинство ошибок, обнаруживаемых в конце жизненного цикла программ, и, следовательно, наиболее дорогих и сложных для исправления, возникает из-за ошибок в спецификации. Таким образом, для предотвращения неисправностей СКЗИ рассмотренной фазе создания необходимо уделять особенное внимание.

С фазой анализа и спецификацией требований связаны системные ошибки. *Системные ошибки* определяются прежде всего неполной информацией о реальных процессах, происходящих в источниках и потребителях информации. Применительно к реализации криптографических алгоритмов можно говорить о неверном понимании, либо трактовке элементов алгоритма.

Фаза проектирования

Фаза проектирования как самостоятельный этап в большей степени может быть выделена при проектировании сложных программных комплексов, составной частью которых являются модули криптографических преобразований. В случае реализации самих модулей криптографических преобразований фаза проектирования может быть выражена не столь отчетливо или практически отсутствовать.

С фазой проектирования связаны алгоритмические ошибки. К *алгоритмическим* относят ошибки, обусловленные некорректной постановкой задач, решаемых отдельными частями ПО. К ним также относят ошибки связей модулей и функциональных групп программ. В большинстве случаев их также можно свести к ошибкам в спецификациях.

Фаза исполнения

Фаза исполнения включает в себя кодирование, интегрирование, а также тестирование и отладку. С ней связаны программные ошибки. Программные ошибки по количеству и типам, в первую очередь, определяются степенью автоматизации программирования и глубиной формализованного контроля текстов программ. Программные ошибки сильно зависят от выбранного языка программирования. Имеющаяся статистика показывает, что наибольший вес имеют ошибки неполной программной реализации функций алгоритма или неверный порядок реализации функций.

Таким образом, на основании анализа фаз создания СКЗИ и допускаемых на них ошибок можно сделать вывод о том, что двумя основными разновидностями ошибок являются:

* неверное специфицирование как всего программного комплекса, так и отдельных его составляющих;
* функциональное несоответствие программы алгоритму.

Предотвращение данных ошибок – путь к обеспечению защиты от сбоев и неисправностей СКЗИ, а также обеспечение точной реализации заданных криптографических алгоритмов в программных и программно-аппаратных СКЗИ.

#### Вопросы и задания для самостоятельной работы

1. Какие таксономические признаки лежат в основе классификации СКЗИ?
2. В каком режиме осуществляется шифрование в системах прозрачного шифрования?
3. Какие основные способы шифрования используются в СКЗИ для передачи данных по сетям?
4. Какой механизм аутентификации используется в СКЗИ для подтверждения подлинности электронного документа?
5. Что является основным классификационным признаком средств управления ключевой информацией?
6. В чем состоит криптографическая задача СКЗИ для обеспечения целостности?
7. Какие методы используются в СКЗИ с целью обеспечения аутентификации?
8. В чем состоит основной принцип Керкгоффса при разработке криптографических систем?
9. Что является критерием выбора СКЗИ?
10. Как называется числовая комбинация, используемая для проверки целостности?
11. Что общего и чем отличаются код аутентичности и код аутентификации?
12. Перечислите преимущества и недостатки программных СКЗИ.
13. Какие факторы влияют на надежность СКЗИ?
14. Какие методы используются разработчиками для оперативного обнаружения дефектов при выполнении программ и искажений данных?
15. Что является основным критерием при выборе способа реализации: аппаратная (аппаратно-программная) или чисто программная?
16. Приведите примеры областей применения СКЗИ, для которых не существует альтернатив обеспечения информационной безопасности.
17. Что такое криптопровайдер?
18. Какую роль играет прикладной криптографический интерфейс CryptoAPI, встроенный в операционные системы Microsoft?

## ТЕМА 3. СРЕДСТВА КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ НА ПЕРСОНАЛЬНОМ КОМПЬЮТЕРЕ

**3.1. Задачи обеспечения информационной безопасности на**

### персональном компьютере с использованием СКЗИ

Персональные компьютеры (ПК) обладают всеми свойствами ЭВМ других классов, поэтому все задачи защиты информации на ПК и используемые СКЗИ аналогичны рассмотренным выше. Однако персональным компьютерам присущ ряд таких свойств, которые, с одной стороны, благоприятствуют защите, а с другой – затрудняют ее и усложняют.

К основным из указанных свойств относятся:

* малые габариты и вес, что делает их не просто транспортабельными, а легко переносимыми;
* наличие встроенного внутреннего ЗУ большого объема, сохраняющего записанные данные после выключения питания;
* наличие сменного ЗУ большого объема и малых габаритов;
* наличие устройств сопряжения с каналами связи;
* оснащенность программным обеспечением с широкими функциональными возможностями;
* массовость производства и распространения; • относительно низкая стоимость.

Перечисленные и некоторые другие особенности создали объективные предпосылки для массового распространения ПК практически во всех сферах деятельности современного общества, резкого повышения интенсивности циркуляции информации, децентрализации процессов ее хранения и обработки, существенного изменения структуры и содержания информационных технологий.

С точки зрения общих подходов к защите особенно существенными, в отличие от систем обработки данных с большими ЭВМ, являются две особенности ПК:

* отсутствие наряду с защитой информации непосредственно в ЭВМ общей организации защиты – организации и обеспечения технологических процессов циркуляции и обработки потоков информации; охраны территории, зданий и помещений; подбора, обучения и организации работы персонала и т.п.;
* заботу о защите информации должны проявлять не специалисты-профессионалы, а сами пользователи, которые не только не являются профессионалами в области защиты, но нередко вообще имеют лишь навыки непосредственного решения ограниченного набора задач.

Этими особенностями и обусловлена необходимость самостоятельного рассмотрения вопросов защиты информации в персональных ЭВМ с акценти-

рованием внимания именно *на внутренней защите –* защите от *несанкционированного доступа* (НСД).

Если сравнивать большие ЭВМ и ПК, то для последних угроза НСД особенно велика, и обусловлено это следующими обстоятельствами:

* доступность ПК для посторонних;
* ПК могут использоваться коллективно для обработки информации, что ведет к обезличиванию ответственности;
* оснащение несъемными накопителями, хранящими огромное количество данных;
* широкое распространение съемных носителей;
* *ПК создавалось как индивидуальное средство, а значит, не были предусмотрены специальные средства защиты информации от НСД.*

Сегодня существуют три вида средств, используемых для несанкционированного получения сведений: *человек, программа, аппаратура*. Задача системы защиты ПК от НСД заключается в перекрытии всех возможных каналов утечки.

Основные способы ее реализации, применяемые в ПК с помощью СКЗИ,

|  |  |
| --- | --- |
| можно классифицировать следующим образом: |  |
| • сохранение физической целостности ПК, носителей информации – | |

*обеспечение целостности;*

* распознавание пользователей и компонентов, используемых для работы с данными, – *аутентификация*;
* ограничение доступа к информации, нуждающейся в защите, – *конфиденциальность*;
* архивация – криптографическое закрытие данных, сохраняемых на носителях, а также в процессе их обработки;
* запись всех попыток доступа к защищаемым сведениям.

### 3.2. Криптографическая защита жестких дисков и съемных носителей

Существует несколько технологий шифрования данных на жестких дисках с помощью СКЗИ:

* программное шифрование;
* аппаратное шифрование;
* шифрование до загрузки операционной системы;
* шифрование после загрузки операционной системы;
* шифрование на уровне приложений;
* шифрование на уровне файлов и папок;
* шифрование всего тома;
* шифрование для каждого пользователя;
* шифрование компьютера;
* создание секретного диска.

При рассмотрении *ключей шифрования* часто анализируется длина ключа, но вопросы получения, хранения и использования ключей зачастую являются гораздо более важными. Вопросы хранения и защиты ключей шифрования являются самым слабым местом технологии шифрования.

*Оптимальный алгоритм шифрования* должен быть разработан и реализован таким образом, чтобы единственный способ взлома шифра заключался в подборе единственного правильного ключа из огромного пространства ключей, которое представляет собой диапазон значений, которые может принимать ключ. Этот тип атаки называется методом перебора.

Алгоритмы с использованием симметричных ключей используют, как правило, пространство ключей от 40 до 512 бит. Количество возможных значений ключа равно максимальному численному значению, которое можно выразить с помощью данного количества бит. 40 бит дают максимальное численное значение, равное 1 099 511 627 775 (240 – 1), довольно большое число, хотя современные компьютеры могут с легкостью перебрать все возможные значения 40-разрядного пространства ключей для расшифровки данных. Однако каждый бит, добавляемый к пространству ключей, удваивает количество возможных ключей, поэтому 41-разрядное пространство ключей содержит 2 199 023 255 552 возможных ключей. Увеличение пространства ключей быстро увеличивает количество возможных ключей до такого значения, при котором атаки методом перебора становятся неэффективными при использовании современного оборудования и известных методов.

Алгоритмы реализуются конкретными разработчиками и иногда содержат ошибки, снижающие эффективность алгоритма. В алгоритмах шифрования после экспертной оценки или успешной атаки иногда обнаруживаются фундаментальные изъяны. В некоторых случаях подобные изъяны обнаруживаются спустя несколько лет после публикации алгоритма и его широкого распространения. Если пространство ключей имеет размер, например, 128 бит, это еще не означает, что при создании 128-разрядного ключа используются подлинно случайные числа или все пространство ключей. Ключи необходимо где-то хранить. Если это местоположение можно обнаружить и получить ключ, можно вскрыть зашифрованные данные.

Таким образом, СКЗИ, реализующие технологии шифрования, могут обладать одной или несколькими из указанных ниже уязвимостей:

* изъян в алгоритме шифрования;
* неправильная реализация;
* ключи имеют низкий уровень энтропии;
* ключи легко вскрываются.

Проверка и тестирование стойкости реализации алгоритмов шифрования позволяет значительно снизить вероятность возникновения подобных уязвимостей, поэтому сертификация реализации алгоритмов шифрования является надежной гарантией от ошибок.

При оценке технологии шифрования важно учитывать приведенные ниже факторы.

* *Выбор алгоритма.*

Например, хорошо протестированы алгоритмы: 3DES, AES, Blowfish,

ГОСТ 28147-89 – симметричного шифрования, RSA, Elliptic Curves Ciphering (ECC), ГОСТ Р 34.10 – алгоритмы (или семейство алгоритмов, в случае с ECC) на основе открытых ключей.

* *Создание ключей.*

Эффективный ключ должен быть в достаточной степени случайным и иметь достаточную длину.

* *Производные ключи.*

Созданные ключи не могут быть более стойкими, чем исходные. 256разрядный ключ AES на основе 56-разрядного ключа DES не будет более стойким, чем 56-разрядный ключ DES.

* *Хранилище ключей.*

Например, смарт-карты обеспечивают более высокий уровень безопасности хранилища ключей, чем программные решения.

* *Кэширование ключей.*

Например, технология, кэширующая ключ шифрования в памяти, может позволить злоумышленнику восстановить его из системной памяти или файла подкачки, если кэш неправильно реализован и не защищен.

* *Самое слабое звено.*

Во многих средах самым слабым звеном являются пользователи мобильных ПК.

* *Баланс удобства использования и безопасности.*

Все решения подвержены атакам методом перебора – проблема заключается только в том, сколько времени займет подобная атака. Доступ к данным для пользователей не должен быть слишком сложным.

***Программное шифрование*** является стандартным для большинства технологий и продуктов защиты данных. При программном шифровании криптографические операции выполняются центральным процессором (ЦП) компьютера. Когда компьютер выключен либо находится в спящем режиме или режиме гибернации, ключи шифрования обычно хранятся в зашифрованном виде на диске. Обычно исходный ключ хранится отдельно от компьютера, например на USB-устройстве, которое используется для расшифровки данных ключа. Когда компьютер включен, ключи шифрования обычно хранятся в оперативной памяти.

*Преимущества:*

* Возможность обновления и изменения реализации. Технологии программного шифрования можно в любое время обновить для исправления изъянов реализации, добавления новых возможностей или использования преимуществ новых алгоритмов.
* Отсутствие необходимости использовать специальное оборудование. Поскольку технологии программного шифрования не требуют специального оборудования, их можно использовать на всех компьютерах в организации.

*Недостатки:*

* Уязвимости программного обеспечения. Решения, использующие только программное обеспечение, подвержены атакам, целью которых является попытка обойти безопасный режим работы. Типичная атака заключается в изменении двоичных файлов операционной системы таким образом, чтобы предотвратить шифрование, изменить ключи или существенно ослабить шифрование.
* Обнаружение ключей. Если ключ шифрования хранится на компьютере, атака может привести к раскрытию его значения. Важно понимать, каким образом защищен ключ. Если ключ расшифровывается только с помощью данных другого ключа, хранящегося на том же компьютере, следует ожидать, что опытному и терпеливому злоумышленнику удастся вскрыть ключ.

***Аппаратное шифрование*.** Некоторые механизмы шифрования используют специальное криптографическое оборудование (например, криптографические модули на основе плат, подключаемых к ПК по PCI) для изолирования криптографических операций от ЦП и обеспечения повышенного уровня безопасности хранилища ключей. Подобное оборудование обычно предоставляет средства безопасного хранения одного или нескольких ключей шифрования, а также функции для выполнения криптографических операций на аппаратном уровне, поэтому ключ никогда бывает доступен другим программным или аппаратным компонентам.

*Преимущества:*

* Ключи шифрования защищены от уязвимостей программного обеспечения и операционной системы. Аппаратное шифрование, как правило, гарантирует, что закрытые составляющие пар ключей изолированы от памяти, контролируемой операционной системой.
* Криптографические операции защищены от уязвимостей программного обеспечения и операционной системы. Аппаратное шифрование не зависит от используемой операционной системы и не подвержено уязвимостям внешнего программного обеспечения.

*Недостатки:*

* Ограниченная совместимость. Например, доверенный платформенный модуль, используемый технологией BitLocker, невозможно установить на старые компьютеры. Кроме того, этот модуль не всегда доступен для новых компьютеров.
* Сложность обновления. При обнаружении изъяна в оборудовании, как правило, требуется заменить его. Замена оборудования также требуется в случае, если его изготовитель добавляет новые возможности или поддержку новых алгоритмов.

***Шифрование до загрузки операционной системы****.* Можно установить микропрограмму, работающую на уровне BIOS, которая обеспечивает шифрование всех данных, записываемых на том жесткого диска, и расшифровку всех данных, считываемых с него. Данные операции можно сделать прозрачными для операционной системы и, таким образом, применять их к файлам операционной системы.

При наличии криптографического оборудования, например доверенного платформенного модуля, его можно использовать для обеспечения более высокого уровня безопасности шифрования и расшифровки до загрузки операционной системы. Компьютеры, оснащенные доверенным платформенным модулем, также могут создавать зашифрованный ключ, привязанный к определенным параметрам платформы, таким как код основной загрузочной записи (MBR), загрузочный сектор NTFS, блок загрузки NTFS и диспетчер загрузки NTFS. Ключи такого типа можно расшифровать только при совпадении параметров платформы со значениями, при которых был создан ключ. Этот процесс называется запечатыванием ключа в доверенном платформенном модуле, а процесс его расшифровки называется распечатыванием.

Доверенный платформенный модуль также может запечатывать и распечатывать данные, созданные за его пределами. Практические преимущества данной функции заключаются в том, что возможность распаковки ключа может зависеть от изменения определенных характеристик платформы, предположительно вследствие действий злоумышленника, направленных на преодоление таких мер безопасности, как шифрование.

Поскольку шифрование применяется к файлам операционной системы, ключ для расшифровки этих файлах необходимо предоставить до загрузки операционной системы. Этот ключ может быть различным для разных решений и может создаваться на основе персонального идентификационного номера (ПИН) или ключа, хранящегося на устройстве, например на USB-носителе или смарткарте.

*Преимущества:*

* Файлы операционной системы защищены от автономной атаки. Все системные файлы и файлы конфигурации защищены решением для шифрования всего тома. Даже если злоумышленнику удастся подключить защищенный том к другой операционной системе – провести автономную атаку – он не сможет сделать ничего, кроме вывода операционной системы из строя.
* Повышенный уровень защиты файлов операционной системы. Криптографическое оборудование, например доверенный платформенный модуль 1.2 с совместимыми обновлениями BIOS, предоставляет возможность проверки целостности критически важных компонентов первоначальной загрузки.

*Недостатки:*

* Обязательное использование стратегии восстановления данных. Любой сбой BIOS, доверенного платформенного модуля или механизма хранения ключей приведет к невозможности чтения всех данных на компьютере. Сбои оборудования зачастую сложнее диагностировать, классифицировать и устранять, чем сбои программного обеспечения. Ремонт может занять больше времени, если оборудование необходимо вернуть изготовителю или отремонтировать за пределами организации. По этой причине необходимо разработать и часто тестировать эффективную и надежную стратегию резервного копирования и восстановления ключей.
* Усложнение процесса обновления программного обеспечения. Поскольку файлы операционной системы и другие файлы зашифрованы и проверяются по электронной подписи, для обновления этих файлов может требоваться специальный процесс. Это может привести к необходимости выполнения дополнительных операций на компьютерах, использующих технологию шифрования до загрузки операционной системы.

***Шифрование после загрузки операционной системы*** может выполняться операционной системой или любым запущенным на компьютере приложением. Примером технологии шифрования после загрузки операционной системы является Encryption File System (EFS). Эта технология встроена в операционную систему Windows, поэтому ее нельзя использовать для шифрования самой операционной системы. Однако эта технология является эффективным средством защиты данных пользователя и приложений.

*Преимущества:*

* Сбои шифрования не приводят к тому, что компьютер становится непригодным к использованию. Даже в случае сбоя технологии шифрования компьютер остается работоспособным. Таким образом, зашифрованные данные можно восстановить без применения другого компьютера.

*Недостатки:*

* Отсутствие защиты файлов операционной системы и файлов конфигурации. Если жесткий диск с конфиденциальными данными подключить к другому компьютеру или загрузить на мобильном компьютере другую операционную систему, можно изменить исходную операционную систему таким образом, чтобы обойти технологию шифрования.
* Обязательное использование стратегии восстановления данных. Любой сбой операционной системы или приложения может привести к невозможности чтения защищенных данных. По этой причине необходимо разработать и часто тестировать эффективную и надежную стратегию резервного копирования и восстановления ключей.

***Шифрование на уровне приложений.*** Шифрование также можно реализовать не на уровне BIOS или операционной системы, а на уровне приложений. Многие современные приложения, в том числе WinZip, Microsoft Office и Intuit Quicken, позволяют шифровать данные.

*Преимущества:*

* Независимость от платформы. Если приложение поддерживает несколько операционных систем, данные, зашифрованные на уровне приложения, обычно можно перенести с одной платформы на другую и расшифровать при наличии правильного ключа.
* Возможность переноса зашифрованных данных. Когда данные зашифрованы на уровне операционной системы, они обычно расшифровываются при выполнении операций с файлами, например при копировании и перемещении. Если целевая папка или система настроена на шифрование, данные могут быть зашифрованы с использованием совершенно другого набора ключей. Шифрование на уровне приложений обычно позволяет переносить данные в исходном зашифрованном виде в другое местоположение.

*Недостатки:*

* Зависимость от приложения. При преобразовании данных из одного формата в другой или их перемещении из одного контейнера в другой шифрование, как правило, не сохраняется. Например, пользователь может извлечь файл из зашифрованного архива WinZip, и извлеченный файл будет расшифрован. Если пользователь не удалит файл по окончании работы с ним, данные могут быть раскрыты.

***Шифрование на уровне файлов и папок*** позволяет защитить определенные файлы и папки, а также содержащиеся в них данных. Это решение защищает только файлы, настроенные для шифрования. Все остальные данные на компьютере шифроваться не будут. Типичным подходом к шифрованию на уровне файлов и папок является создание уникального ключа шифрования для каждого файла или папки. Этот подход имеет дополнительное преимущество, которое заключается в возможности реализации шифрования данных каждого пользователя, как описано далее в данной главе.

*Преимущества:*

* Более высокая производительность, чем при шифровании всего тома. Производительность компьютера изменяется незначительно. Тем не менее производительность системы в любом случае снизится. Шифрование на уровне файлов и папок снижает влияние шифрования на производительность, поскольку дополнительные ресурсы затрачиваются только на те файлы, которые необходимо зашифровать в соответствии с политикой безопасности данных организации.
* Выборочное шифрование. Точные средства управления подобного решения позволяют пользователям шифровать только конфиденциальные данные, а администраторам – принудительно шифровать конкретные папки, файлы или типы данных (либо блокировать их шифрование).
* Поддержка нескольких пользователей. Владельцы файлов могут разрешить другим пользователям читать или изменять зашифрованные файлы. Эта возможность позволяет безопасно обмениваться зашифрованными файлами с другими пользователями.

*Недостатки:*

* Возможность утечки конфиденциальных данных через файлы, созданные операционной системой или приложениями. Как правило, операционная система записывает данные приложений, хранящиеся в памяти, в файлы на жестком диске. Эти данные могут содержать конфиденциальные сведения. В операционной системе Windows к таким файлам относятся системный файл подкачки и файл гибернации. Операционная система может также создавать файлы журналов и другие безобидные файлы, которые могут содержать конфиденциальные данные.
* Возможность утечки конфиденциальных данных через кэш данных уровня приложений. Приложения могут использовать собственные средства кэширования или ведения журнала, например временные файлы, через которые может происходить утечка конфиденциальных данных. Хорошим примером являются файлы восстановления, создаваемые Microsoft Word. Эту уязвимости можно в некоторой степени снизить, настроив приложение на создание временных файлов в определенной папке, все файлы в которой шифруются.
* Непреднамеренное копирование файлов в незашифрованный файл или папку. Поскольку шифруются только определенные файлы и папки, пользователь может непреднамеренно скопировать содержимое файла в другой файл, расположенный в незашифрованной папке.

***Шифрование всего тома*** дополняет шифрование на уровне файлов и папок и позволяет устранить общие проблемы, связанные с ним. Если том, который необходимо защитить, содержит файлы операционной системы, необходимо использовать технологию шифрования до загрузки операционной системы. При шифровании всего тома, содержащего файлы операционной системы, необходимо учитывать преимущества и недостатки шифрования до загрузки операционной системы.

Перечисленные ниже преимущества шифрования всего тома позволяют устранить недостатки шифрования на уровне файлов и папок, описанные выше.

*Преимущества:*

* Шифрование временных файлов операционной системы. Поскольку шифруется весь том, выполняется автоматическое шифрование любых файлов, записываемых на этот том, в том числе системного файла подкачки и файла гибернации.
* Шифрование временных файлов приложений. Все временные файлы, создаваемые приложениями, записываются на зашифрованный том и автоматически шифруются.
* Автоматическое шифрование всех файлов, создаваемых пользователем. Если пользователь копирует файл в другую папку тома, он будет автоматически зашифрован. Шифрование всего тома значительно снижает риск ошибки пользователя, которая может привести к обходу шифрования.

*Недостатки:*

* Снижение производительности. При чтении данных необходимо расшифровывать каждый блок на зашифрованном томе, а при записи на диск каждый блок необходимо зашифровывать повторно. Эта функция применяется к исполняемым файлам и файлам конфигурации операционной системы и приложений, а также ко всем файлам данных. Хотя современные технологии шифрования относительно эффективны, при шифровании всего тома следует ожидать снижение производительности от 5 до 15%.
* Ограниченная защита от атак со стороны сотрудников организации. Шифрование всего тома обеспечивает защиту от ряда автономных атак, но оно обычно обеспечивает очень ограниченную защиту от действий сотрудников организации, которые обычно имеют (или могут получить) возможность войти в систему на целевом компьютере с допустимой учетной записью.

***Шифрование для каждого пользователя.*** Шифрование можно реализовать таким способом, что несколько пользователей могут расшифровывать ключи, необходимые для шифрования и расшифровки файлов данных на компьютере, с помощью собственного уникального ключа, который может быть паролем или ключом, хранящимся на USB-устройстве или ином устройстве. При использовании этого подхода совместно с шифрованием на уровне файлов и папок с использованием индивидуальных ключей можно предоставлять пользователям доступ к отдельным файлам.

*Преимущества:*

* Точный контроль доступа к зашифрованным данным. Другие пользователи компьютера не смогут прочитать зашифрованные данные, пока владелец файла явным образом не предоставит к нему доступ. Эта функция обеспечивает контроль доступа и конфиденциальность данных.
* Возможность выборочного шифрования только конфиденциальных данных. При надлежащей реализации системы шифрования для каждого пользователя можно защитить только определенные файлы и папки.
* Возможность шифрования файлов для нескольких пользователей. При надлежащей реализации системы шифрования для каждого пользователя владелец файла может зашифровать отдельный файл для нескольких пользователей, обеспечивая общий доступ при сохранении достаточного уровня безопасности. Кроме того, эту функцию можно использовать для восстановления данных, разрешив авторизованному агенту восстановления расшифровывать защищенные файлы.

*Недостатки:*

* Безопасность всего решения равна безопасности самого слабого ключа (учетных данных). При этом подходе каждый ключ шифрования и расшифровки шифруется несколькими ключами (по одному ключу для каждого уникального пользователя). Каждый зашифрованный ключ можно атаковать отдельно, поэтому злоумышленник ищет наименее стойкий ключ, особенно если для создания данных ключа используются пароли для входа в систему, сеть и иные пароли.

***Шифрование компьютера*.** Некоторые реализации технологии шифрования не предоставляют возможности использования несколькими лицами отдельного ключа или пароля для расшифровки главных ключей, необходимых для расшифровки данных на компьютере. При подобной реализации существует только один ключ, используемый для доступа к компьютеру и ко всем зашифрованным данным.

*Преимущества:*

* Упрощенная последовательность создания ключей. Если для начала последовательности создания ключей используется только один ключ, весь механизм существенно упрощается. Меньшая сложность может обеспечить (но не всегда обеспечивает) повышенный уровень безопасности.

*Недостатки:*

* Отсутствие защиты от злоумышленников внутри организации. Любой пользователь, которому политикой разрешено входить в систему на защищенном компьютере, может получить доступ к любому файлу в незашифрованном виде.

#### Создание секретных дисков

Защищенный виртуальный диск создается в виде файла, который можно скопировать на любой компьютер и смонтировать, введя пароль.

Например, свободная программа ***TrueCrypt*** позволяет создавать полностью зашифрованный виртуальный диск, на котором после начального ввода пароля доступа данные будут автоматически зашифровываться и расшифровываться без внешнего вмешательства.

Ни один диск TrueCrypt не может быть аутентифицирован. Защищенный диск, созданный с помощью TrueCrypt, невозможно отличить от набора случайных данных.

Внутри диска можно создать дополнительный скрытый диск размером меньше оригинала (скрытый том). В случае монтирования внешнего тома скрытый том с конфиденциальными файлами остается невидим.

***BestCrypt***– платный пакет проприетарных программ для создания на жестком диске компьютера виртуального зашифрованного диска – одного или нескольких. Виртуальный диск работает, как обычный дисковый раздел.

Шифрование и расшифрование идут в фоновом режиме, и пользователь не замечает разницы в работе с обычным и зашифрованным диском, который, при необходимости, можно превратить в обычный, но нечитаемый, файл.

У программы BestCrypt также есть опция создания «скрытого» контейнера, который криптографически гарантированно нельзя определить никакими средствами.

**3.3. Средства шифрования, встроенные в операционную систему**

### Windows

В ОС Windows встроено два взаимодополняющих СКЗИ – шифрующая файловая система (EFS) и Microsoft® BitLocker™ (BitLocker), – являющихся примерами двух различных, но взаимодополняющих подходов к шифрованию данных.

Файловая система EFS – это механизм шифрования, защищающий данные в файлах и папках каждого пользователя, предоставляет доступ к зашифрованным файлам только пользователям, прошедшим проверку на работающем компьютере.

BitLocker – это механизм шифрования всего тома, шифрующий все секторы на системном томе компьютера, в том числе операционную систему, приложения и файлы данных. Технология BitLocker обеспечивает проверку целостности и шифрование до загрузки операционной системы, но не позволяет выполнять проверку подлинности пользователя.

Файловая система EFS дополняет возможности технологии BitLocker.

Вследствие фундаментально различных подходов и методов реализации технологии EFS и BitLocker обладают собственными достоинствами и недостатками и обеспечивают разные уровни безопасности в случае распространенных атак злоумышленников.

Например, файловую систему EFS можно использовать для шифрования файлов данных в сети, доступ к которым имеет множество различных пользователей, что невозможно сделать с помощью технологии BitLocker.

В таблице 3.1 приведены алгоритмы шифрования, используемые в EFS

*Таблица 3.1* **Алгоритмы шифрования EFS**

|  |  |  |
| --- | --- | --- |
| **Операционная система** | **Алгоритм шифрования по умолчанию** | **Альтернативные доступные алгоритмы** |
| Windows 2000 | DESX | (none) |
| Windows XP RTM | DESX | 3DES |
| Windows XP SP1 | AES | 3DES, DESX |
| Windows Server 2003 | AES | 3DES, DESX |
| Windows Vista | AES | 3DES, DESX |
| Windows 7 | AES | 3DES, DESX |
| Windows Server 2008 | AES | 3DES, DESX |

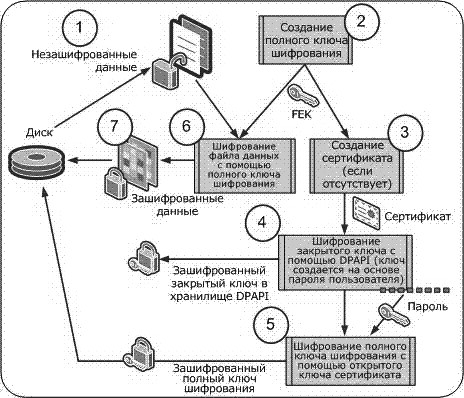
*Режим EFS с программным хранилищем ключей*

Для использования этого режима EFS требуется только компьютер с Windows XP или Windows Vista.

Логическая последовательность процесса шифрования EFS в данном режиме показана на рис. 3.1.

*Шифрование:*

1. Пользователь создает файл в зашифрованной папке.
2. Случайным образом создается симметричный ключ шифрования файлов.
3. Операционная система проверяет хранилище сертификатов пользователя на наличие сертификата, имеющего соответствующие флаги использования ключа. Если такой сертификат отсутствует, он создается автоматически.
4. Для шифрования и хранения закрытого ключа, связанного с сертификатом пользователя, используется интерфейс DPAPI.
5. Ключ шифрования файлов шифруется с помощью открытого ключа сертификата и хранится в метаданных файла.
6. Ключ шифрования файлов используется для шифрования каждого блока данных.
7. Зашифрованные блоки записываются на диск.



***Рис. 3.1.*** *Схема шифрования*

Логическая последовательность процесса расшифрования EFS в данном режиме показана на рис. 3.2.

*Расшифрование:*

1. Учетные данные пользователя для входа в систему проверяется либо локально, либо контроллером домена.
2. Пользователь пытается получить доступ к зашифрованному файлу.
3. Интерфейс DPAPI используется для загрузки закрытого ключа, связанного с сертификатом пользователя X.509. Этот ключ расшифровывается с помощью ключа, полученного на основе учетных данных пользователя.
4. Ключ шифрования файлов загружается из файла и расшифровывается с помощью закрытого ключа, полученного на предыдущем этапе.
5. Ключ шифрования файлов используется для расшифровки каждого блока файла при его запросе.
6. Расшифрованные данные передаются запросившему их приложению.



***Рис. 3.2.*** *Схема расшифрования*

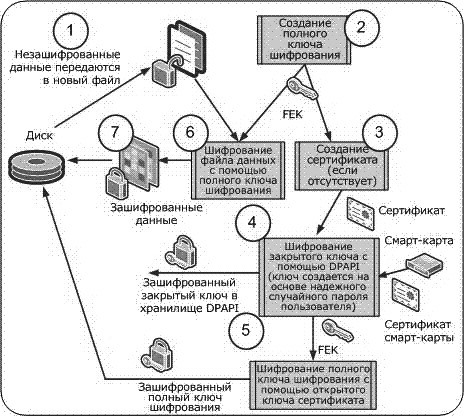
#### Режим EFS с использованием смарт-карт

Различные версии Windows предоставляют различные возможности для улучшения функций безопасности файловой системы EFS.

Параметр политики Active Directory, требующий смарт-карту для входа в систему, значительно повышает безопасность учетной записи, а следовательно, безопасность данных, защищенных с помощью EFS. Этот параметр также позволяет защитить зашифрованные данные от автономных атак, поскольку он повышает стойкость ключа, используемого EFS для шифрования DPAPI. DPAPI создает ключ на основе учетных данных пользователя.

#### Шифрование EFS с использованием смарт-карт в Windows XP

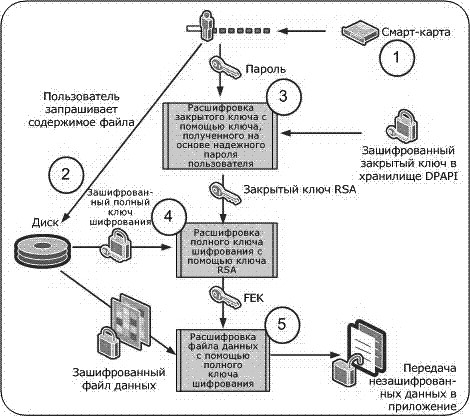
1. Пользователь создает файл в зашифрованной папке.
2. Случайным образом создается симметричный ключ шифрования файлов.
3. Операционная система проверяет хранилище сертификатов пользователя на наличие сертификата, имеющего соответствующие флаги использования ключа. Если такой сертификат отсутствует, он создается автоматически.
4. Для шифрования и хранения закрытого ключа, связанного с сертификатом пользователя, используется интерфейс DPAPI. Такое шифрование является более стойким, чем используемое в предыдущем режиме, поскольку при принудительном входе в систему с помощью смарт-карт применяется значительно более надежный пароль.
5. Ключ шифрования файлов шифруется с помощью открытого ключа сертификата и хранится в метаданных файла.
6. Ключ шифрования файлов используется для шифрования каждого блока данных.
7. Зашифрованные блоки записываются на диск.



***Рис. 3.3.*** *Схема шифрования с использованием смарт-карт в Windows XP*

*Процесс расшифрования*

1. Учетные данные пользователя для входа в систему с использованием смарт-карт проверяется либо локально, либо контроллером домена.
2. Пользователь пытается получить доступ к зашифрованному файлу.
3. Правильный закрытый ключ загружается из хранилища DPAPI пользователя, а DPAPI расшифровывает его с помощью ключа, полученного на основе надежного случайного пароля пользователя, применяемого к учетной записи пользователя при включенном параметре «Вход со смарт-картой».
4. Ключ шифрования файлов загружается из файла и расшифровывается с помощью закрытого ключа, полученного на предыдущем этапе.
5. По мере считывания приложением каждого блока файла он расшифровывается с помощью ключа шифрования файлов, а затем передается запросившему его приложению.



***Рис. 3.4.*** *Схема расшифрования с использованием смарт-карт в Windows XP*

В таблице 3.2 содержатся данные о рисках и различных режимах EFS, которые позволяют снизить каждый из них. Для компьютеров с Windows XP включен параметр пользователей домена «Вход со смарт-картой». Для компьютеров с Windows Vista включены параметры EFS «Требовать смарт-карту для EFS» и «Включить шифрование файла подкачки».

Риски, которые можно снизить при использовании определенных режимов, помечены знаком плюс (+). Знак минус (-) обозначает риски, которые нельзя существенно снизить с помощью соответствующего режима.

*Таблица 3.2*

**Снижение рисков EFS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Риск** | **Windows XP EFS** | **Windows Vista EFS** | **Windows XP EFS со смарткартой** | **Windows**  **Vista EFS со смарткартой (cached mode)** | **Windows**  **Vista EFS со смарткартой (uncached mode)** |
| Компьютер оставлен в режиме гибернации | - | - | - | - | + |
| Компьютер оставлен в спящем  (ждущем) режиме | - | - | - | - | + |

*Продолжение табл. 3.2.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Риск** | **Windows XP EFS** | **Windows Vista EFS** | **Windows**  **XP EFS со смарткартой** | **Windows**  **Vista EFS со смарткартой (cached mode)** | **Windows**  **Vista EFS со смарткартой (uncached mode)** |
| Пользователь не вышел из системы,  компьютер разблокирован | - | - | - | - | + |
| Обнаружение ло-  кального пароля или пароля домена | - | - | + | + | + |
| Сотрудник организации может прочитать зашифрованные данные | + | + | + | + | + |
| Обнаружение ключей с помощью автономной атаки | + | + | + | + | + |
| Автономные атаки на операционную систему | - | - | - | - | - |
| Сетевые атаки на операционную систему | - | - | - | - | + |
| На компьютере находятся неза-  шифрованные данные | + | + | + | + | + |
| Утечка незашифрованных данных через файл гибернации | - | - | - | - | - |
| Утечка незашифрованных данных через системный файл подкачки | - | + | - | + | + |
| Атаки на платформу | - | - | - | - | + |
| Необходимое средство проверки подлинности  оставлено на ком-  пьютере или рядом  с ним | - | - | + | + | + |
| Ошибка пользователя | - | - | - | - | - |

#### BitLocker

Шифрование диска BitLocker является встроенной функцией безопасности в операционной системе Windows 7/8, которая позволяет защитить данные, хранящиеся на фиксированных и съемных дисках, а также на диске операционной системы.

BitLocker позволяет защитить компьютер от «атак с выключением», которые проводятся путем отключения или обхода установленной операционной системы либо путем физического удаления жесткого диска для взлома данных автономно.

Для фиксированных и съемных дисков обеспечивает прочтение и запись данных на диск только для пользователей с требуемым паролем, учетными данными смарт-карты.

Обеспечивает защиту от несанкционированного доступа к данным на утерянных или украденных компьютерах

Поддерживаются следующие алгоритмы шифрования:

* AES 128,
* AES 128 c Elephant diffuser *(AES in CBC mode)* (используется по умолчанию),
* AES 256,
* AES 256 c Elephant diffuser.

Elephant Diffuser – метод обеспечения дополнительного уровня безопасности, разработанный специально для шифрования данных в Windows компанией Microsoft.

Сам ключ может храниться в Trusted Platform Module (TPM), в USBустройстве или в компьютере. В случае с TPM при загрузке компьютера ключ может быть получен из него сразу либо только после аутентификации с помощью USB-ключа (данный режим требует активации через групповые политики) или ввода PIN-кода пользователем. Таким образом, возможны следующие комбинации для доступа:

* TPM,
* TPM + PIN,
* TPM + PIN + USB-ключ,  TPM + USB-ключ.

BitLocker обеспечивает защиту от несанкционированного доступа к данным на утерянных или украденных компьютерах, выполняя следующие операции:

* Шифрование всего диска с операционной системой Windows на жестком диске. BitLocker шифрует все файлы пользователя и системные файлы на диске операционной системы, в том числе файлы подкачки и файлы гибернации.
* Проверка целостности компонентов загрузки и данных конфигурации загрузки. На компьютерах с доверенным платформенным модулем версии 1.2 или 2.0 BitLocker использует расширенные функции этого модуля по обеспечению безопасности, гарантируя возможность доступа к данным только в случае, когда отсутствуют изменения в загрузочных компонентах компьютера, а жесткий диск находится в исходном компьютере.

BitLocker интегрируется в Windows 7/8 и предоставляет предприятиям удобные в управлении и настройке функции усиленной защиты данных. Например, BitLocker может использовать существующую инфраструктуру доменных служб Active Directory (AD DS) организации для удаленного хранения ключей восстановления BitLocker.

BitLocker можно использовать для защиты съемных и несъемных дисков с данными. BitLocker шифрует все содержимое диска с данными, а в групповой политике можно настроить, чтобы перед записью данных на такой диск для него требовалось включить BitLocker. В BitLocker можно настроить следующие методы снятия блокировки для дисков с данными:

* Автоматическое снятие блокировки. Можно настроить автоматическое снятие блокировки для несъемных дисков с данными на компьютере, где шифруется диск с операционной системой. Для съемных дисков с данными на компьютере, работающем под управлением Windows 8, можно настроить автоматическое снятие блокировки после однократного использования пароля или смарт-карты. Но для съемных дисков в дополнение к автоматическому методу всегда должен действовать метод снятия блокировки с помощью пароля или смарт-карты.
* Пароль. Пользователю, открывающему диск, будет предложено ввести пароль для снятия блокировки диска. Этот метод можно использовать с устройством чтения BitLocker To Go на компьютерах с Windows Vista или Windows XP для открытия дисков, защищенных BitLocker, в режиме только для чтения.
* Смарт-карта. Пользователю, открывающему диск, будет предложено вставить смарт-карту для снятия блокировки диска.
* Учетная запись или группа Active Directory. Пользователю, группе или учетной записи компьютера Active Directory назначается ключ, и блокировка диска снимается, когда вводятся учетные данные этого пользователя или группы.

Диск может поддерживать несколько методов снятия блокировки. Например, для съемного диска с данными можно настроить автоматическое снятие блокировки на основном рабочем компьютере и запрос пароля на других компьютерах.

BitLocker поддерживает многофакторную проверку подлинности для дисков операционной системы. Если включить BitLocker на компьютере с доверенным платформенным модулем версии 1.2 или 2.0, то можно использовать дополнительные формы проверки подлинности на основе этого модуля. BitLocker позволяет блокировать обычный процесс загрузки, пока пользователь не введет пароль (которым может служить ПИН-код или парольная фраза) или не вставит устройство USB (например, устройство флэш-памяти), которое содержит ключ запуска BitLocker. Можно требовать и ввода пароля, и наличия устройства USB. Эти дополнительные меры безопасности обеспечивают многофакторную проверку подлинности и гарантию того, что компьютер не будет запущен или выведен из режима гибернации, пока не выполнены условия действующего метода проверки подлинности.

Требования BitLocker к оборудованию и программному обеспечению для дисков с операционной системой:

* Наличие доверенного платформенного модуля версии 1.2 или 2.0. Для работы BitLocker не обязателен аппаратный платформенный модуль, но его наличие позволяет обеспечить дополнительные меры безопасности в виде проверки целостности системы до запуска и многофакторной проверки подлинности.
* Наличие встроенного ПО BIOS или UEFI, отвечающего стандартам организации TCG.
* Необходимость задания порядка загрузки, где первым идет жесткий диск, а не USB-диск или компакт-диск.
* Встроенное ПО должно поддерживать чтение с USB-устройства флэшпамяти во время запуска.
* Для компьютеров, поддерживающих загрузку UEFI в собственном режиме, требуется хотя бы один раздел FAT32 для системного диска и один раздел NTFS для диска с операционной системой.
* Для компьютеров со встроенным ПО BIOS прежних версий необходимо не менее двух разделов NTFS: один для системного диска и один для диска с операционной системой.
* Для любого вида встроенного ПО раздел системного диска должен иметь размер не менее 350 мегабайт (МБ) и быть активным.
* Диск с аппаратным шифрованием, используемый в качестве загрузочного, должен находиться в неинициализированном состоянии и в состоянии неактивной защиты. Кроме того, система всегда должна загружаться в собственном режиме UEFI версии 2.3.1 или более поздней с отключенным CSM.

Требования BitLocker к оборудованию и программному обеспечению для дисков с данными:

* Для защиты с помощью BitLocker съемный или несъемный диск с данными должен иметь файловую систему exFAT, FAT16, FAT32 или NTFS.
* Для использования устройства чтения BitLocker To Go для чтения данных со съемного диска он должен иметь файловую систему exFAT, FAT16 или FAT32. Файлы с файловой системой NTFS можно разблокировать только на компьютере с Windows Server 2008 R2, Windows 7 или более поздней версии. В предыдущих версиях ОС Windows диск не будет распознан, и система предложит его отформатировать.
* Диск должен иметь размер не менее 64 МБ.

Наличие двух разделов обязательно для работы BitLocker, поскольку проверка подлинности перед запуском и проверка целостности системы должны выполняться на отдельном разделе, не совпадающем с зашифрованным диском операционной системы. Такая конфигурация способствует защите операционной системы и данных на зашифрованном диске. На системном диске также можно хранить среду восстановления Windows и другие файлы, необходимые для установки и обновления программ. Изготовители компьютеров и корпоративные заказчики могут хранить на этом диске системные средства или другие средства восстановления, что увеличивает требования к размеру системного диска. Например, если на системном диске вместе с файлом запуска BitLocker хранится среда восстановления Windows, то размер диска увеличивается до 350 МБ. По умолчанию системный диск скрыт, и ему не назначается никакая буква. Системный диск создается автоматически при установке Windows 8.

Необработанные данные шифруются полным ключом шифрования тома, который затем шифруется основным ключом тома. Основной ключ тома, в свою очередь, шифруется одним из нескольких возможных методов в зависимости от типа проверки подлинности (предохранители ключа или доверенный платформенный модуль) и сценариев восстановления.

Полный ключ шифрования тома шифруется основным ключом тома и хранится на зашифрованном диске. Основной ключ тома шифруется подходящим предохранителем ключа и хранится на зашифрованном диске. Если защита BitLocker приостанавливается, то незащищенный ключ, которым шифруется основной ключ тома, также хранится на зашифрованном диске вместе с зашифрованным основным ключом тома.

Такая процедура хранения гарантирует, что основной ключ тома никогда не хранится без шифрования и всегда защищен, если не отключено шифрование BitLocker. Ключи также сохраняются в двух дополнительных расположениях на диске для обеспечения резервирования. Ключи могут считываться и обрабатываться диспетчером загрузки.

#### Совместное использование EFS и BitLocker

Шифрование диска Microsoft® BitLocker™ (*BitLocker*) и шифрованная файловая система (*EFS*) – это две независимые технологии, которые можно использовать совместно для обеспечения высокого уровня безопасности данных: стойкое шифрование компьютера целиком (BitLocker) и шифрование данных каждого пользователя в отдельности (EFS).

В таблице 3.3 описаны риски и показано, как различные сочетания

BitLocker с доверенным платформенным модулем и ПИН и файловой системы EFS с программным хранилищем ключей позволяют снизить каждый из них. Риски, которые можно снизить при использовании определенных сочетаний, помечены знаком плюс (+). Знак минус (-) обозначают риски, которые нельзя существенно снизить с помощью соответствующего сочетания.

*Таблица 3.3* **Снижение рисков при совместном использовании BitLocker и файловой системы EFS**

|  |  |  |  |
| --- | --- | --- | --- |
| Риск | BitLocker с  TMP и  EFS | BitLocker с TMP, PIN и  EFS с использованием программного хранилища ключей | BitLocker с TMP, PIN и EFS с использованием смарт-карт (в режиме кэширования ключей) |
| Компьютер оставлен в режиме гибернации | - | + | + |
| Компьютер оставлен в спящем (ждущем) режиме | - | - | - |
| Пользователь не вышел из системы, компьютер разблокирован | - | - | - |
| Обнаружение локального пароля или пароля домена | - | + | + |
| Сотрудник организации может прочитать зашифрованные данные | + | + | + |
| Обнаружение ключей с помощью автономной атаки | + | + | + |
| Автономные атаки на операционную систему | + | + | + |
| Сетевые атаки на операционную систему | - | - | - |
| На компьютере находятся незашифрованные данные | + | + | + |
| Утечка незашифрованных данных через файл гибернации | + | + | + |
| Утечка незашифрованных данных через системный файл подкачки | + | + | + |
| Атаки на платформу | - | - | - |
| Необходимое средство проверки подлинности оставлено на компьютере или рядом с ним | - | + | + |
| Ошибка пользователя | + | + | + |

#### СКЗИ приложений Microsoft Office (Word, Excel, PowerPoint)

В М.Office, до версии 6.0 включительно, первым алгоритмом шифрования был обычный XOR открытого текста с паролем. В последующих версиях Microsoft Office 97 и 2000 Microsoft значительно улучшил защиту в Office: использовался сильный криптоалгоритм RC4.

Криптоалгоритмы, применявшиеся за пределами США, не могли использовать ключи длиннее 40 разрядов. Это привело к возможности организации атаки *методом перебора (brute force)*. Так, для расшифровки файла MS Word/Excel 97/2000 нужно перебрать максимум 240 ключей.

С учетом появления таблиц ключей (Rainbow tables) нужную атаку можно совершить за нескольких секунд. Появились сайты в Интернете, оказывающие подобные услуги, например www.decryptum.com (стоимость расшифровки одного файла составляет 29 долл.); программное обеспечение Guaranteed Word Decrypter (GuaWord) 1.7, Guaranteed Excel Decryptor (GuaExcel) 1.7 (производитель PSW-soft), Advanced Office Password Breaker (AOPB), Advanced Office Password Recovery (AOPR) (производитель Elcomsoft).

В Microsoft Office XP/2003 в качестве алгоритма шифрования по умолчанию был оставлен все тот же алгоритм с 40 разрядным ключом. Вместе с тем были внесены следующие изменения:

* алгоритм хеширования SHA1 (вместо MD5);
* ключ RC4 может иметь длину до 128 разрядов;  длина пароля увеличена с 16 до 255 символов.

В Office 2007 была применена новая схема шифрования:

* на смену алгоритму RC4 пришел алгоритм шифрования AES с длиной ключа 128 разрядов;
* вместо однократного хеширования пароль хешируется циклически 50

тыс. раз;

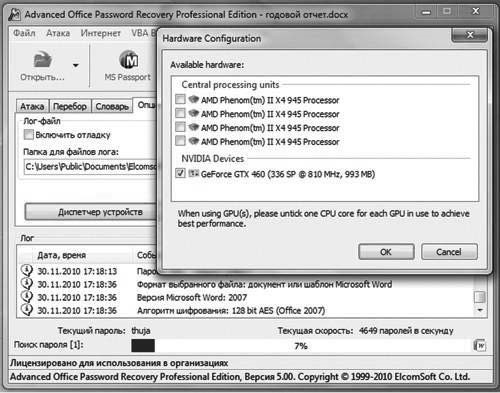
* возможно применение сторонних алгоритмов шифрования.

В результате принятых мер скорость перебора паролей составляет не более 200 паролей в секунду, что позволяет подобрать за разумное время пароли не длиннее 5–6 символов.

С появлением программного обеспечения, использующего ресурсы видеокарты, скорость перебора вырастает до 5 тыс. паролей в секунду, но этого также недостаточно для полноценной атаки методом перебора.

Шифрование и хеширование обеспечивает криптопровайдер Microsoft Enhanced RSA and AES Cryptographic Provider.

Алгоритмы шифрования, применяемые в Microsoft Office 2010, зависят от алгоритмов, доступ к которым можно получить через интерфейсы API в операционной системе Windows. Office 2010, помимо поддержки Cryptography API (CryptoAPI), также поддерживает интерфейс CNG (CryptoAPI: Next Generation), который впервые стал доступен в Microsoft Office 2007 с пакетом обновления 2 (SP2).



***Рис. 3.5.*** *Атака перебором с учетом ресурсов видеокарты*

Используя CNG, можно добиться более динамичного шифрования и применять модули сторонних производителей. Если Office использует CryptoAPI, алгоритмы шифрования зависят от алгоритмов, доступных в провайдере службы криптографии (CSP), который входит в операционную систему Windows.

В Office 2010 и Office 2007 с пакетом обновления 2 (SP2) можно использовать следующие алгоритмы шифрования: *AES, DES, DESX, 3DES.*

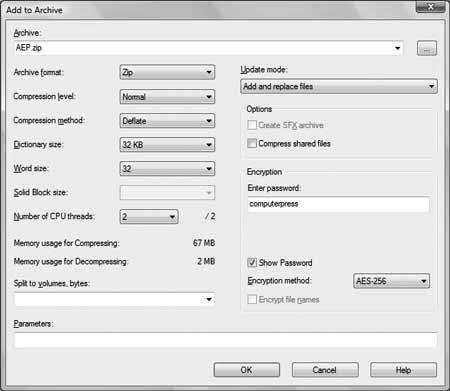
При шифровании документов в формате Open XML (DOCX, XSLX, PPTX и т.д.) алгоритмом шифрования по умолчанию выбран алгоритм *AES с длиной ключа 128 бит.* При шифровании двоичных документов (DOC, XLS, PPT) используется алгоритм *RC4.*

С точки зрения устойчивости к атакам прямого перебора пароли документов в *Microsoft Office 2010* являются на сегодня наиболее эффективными.

#### 3.4. Шифрование архиваторов

WinZip или WinRAR помимо возможности создавать архивы, позволяют зашифровывать их содержимое.

Архиватор WinZip последних версий предполагает использование шифрования *AES с 128- или 256-битным ключом*, а также стандартное для WinZip *PKZIP* шифрование *(Phil Katz ZIP).* Следует помнить, что при наличии хотя бы одного файла из архива весь архив целиком может быть вскрыт за минуту с использованием атаки при известных данных.



***Рис. 3.6.*** *Установка пароля в WinZip*

В WinRAR для шифрования используется алгоритм *AES с ключами длиной 128 бит и 256 бит.* Также WinRAR предлагает возможность шифровать не только файлы, но и имена файлов в архиве – если архив откроет случайный пользователь, он обнаружит контейнер пустым.



***Рис. 3.7.*** *Установка пароля в WinRAR*

Использование AES с 128- и 256-разрядными ключами архиваторами WinRAR и последними версиями WinZip обеспечивает повышенную стойкость защищенных архивов.

#### 3.5. СКЗИ свободного доступа

Растет число свободно доступных средств криптографической защиты информации (СКЗИ). Это программные СКЗИ, имеющие несвободные (proprietary software), открытые (open-source software) или свободные (free software) лицензии, дающие пользователю широкий спектр возможностей по применению. Большинство СКЗИ можно получить и использовать бесплатно, например из сети Интернет. Они не имеют сертификатов ФСБ и ФСТЭК, однако для индивидуальных, корпоративных и коммерческих задач защиты информации такие средства применять на территории России не запрещается.

Свободно доступные СКЗИ позволяют решить практически все задачи защиты информации в информационной среде, реализуя широкий спектр криптографических функций:

* шифрование файлов (PGP Desktop, Diskcryptor, FreeOTFE,GnuPG,

Cloudfogger),

* цифровая подпись (PGP Desktop, GnuPG),
* шифрование электронной почты (GnuPG), шифрование съемных носителей (Diskcryptor, FreeOTFE),
* шифрование жесткого диска (Diskcryptor, FreeOTFE),
* шифрование виртуального диска (Diskcryptor, FreeOTFE),
* разделение секрета (PGP Desktop, GnuPG),
* удаление остаточной конфиденциальной информации (PGP Desktop,

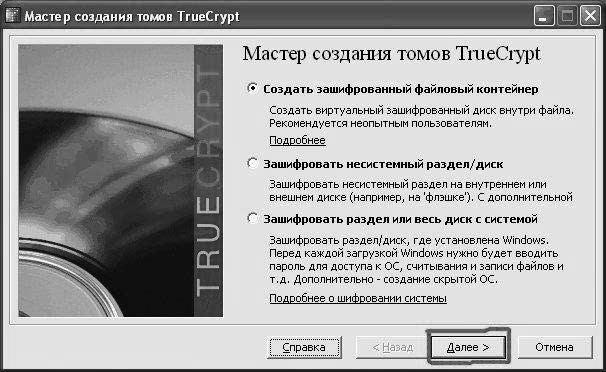
Diskcryptor),

* создание скрытых дисков (FreeOTFE),
* организация защищенного туннеля в общедоступном канале связи (OpenVPN, Hotspot Shield),
* хранение паролей, генерация паролей (KeePass),  шифрование данных в облачных хранилищах (Cloudfogger).

Рассмотрим функциональные возможности СКЗИ свободного доступа на примере широко распространенных TrueCrypt, PGP и DriveCrypt.

##### TrueCrypt

TrueCrypt – это криптографическое программное обеспечение с открытым исходным кодом. Используется для шифрования информации «*на лету*» (*разделов/дисков или устройств хранения данных, таких как USB флешпамять*) – данные *автоматически* зашифровываются и расшифровываются перед их чтением или сохранением без какого-либо участия пользователя.



***Рис. 3.8.*** *Снимок окна программы TrueCrypt, создание зашифрованного файлового контейнера*

Файловая система при этом шифруется в полном объеме (*шифруются имена файлов, каталогов, содержание каждого файла, свободное место, метаданные и т.д*.). Информация, находящаяся в зашифрованном разделе, не может быть прочитана (расшифрована) без введения правильного пароля/keyфайла или ключей шифрования.

*Таблица 3.4*

**Используемые в TrueCrypt алгоритмы**

|  |  |  |  |
| --- | --- | --- | --- |
| **Алгоритм** | **Авторы** | **Размер ключа (бит)** | **Размер блока (бит)** |
| AES | J. Daemen, V. Rijmen | 256 | 128 |
| Serpent | R. Anderson, E. Biham, L. Knudsen | 256 | 128 |
| Twofish | B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson | 256 | 128 |
| AES-Twofish |  | 256; 256 | 128 |
| AES-TwofishSerpent |  | 256; 256; 256 | 128 |
| Serpent-AES |  | 256; 256 | 128 |
| Serpent-  Twofish-AES |  | 256; 256; 256 | 128 |
| **Алгоритм** | **Авторы** | **Размер ключа (бит)** | **Размер блока (бит)** |
| TwofishSerpent |  | 256; 256 | 128 |

***PGP*** PGP (англ*. Pretty Good Privacy*) – компьютерная программа, также библиотека функций, позволяющая выполнять операции *шифрования и цифровой подписи* сообщений, файлов и другой информации, представленной в электронном виде, позволяет шифровать данные, *передаваемые по сети*. Первоначально разработана Филиппом Циммерманном в 1991 году. Существуют как коммерческие, так и некоммерческие версии программы. В 2010 году Symantec Corp. выкупил PGP.

На данный момент не известно ни одного способа взлома данных, зашифрованных PGP, при помощи полного перебора или уязвимости криптоалгоритма.

Приложение *PGP Whole Disk Encryption* предназначено для блокирования всего содержимого ПК, рабочего стола, внешнего диска или носителя USB-fash, включая загрузочные секторы, систему и файлы подкачки. Шифрование *прозрачно для пользователя, защита данных производится автоматически*.

Использует собственные алгоритмы PGP, основанные на алгоритме AES с длинами ключей 128 и 256 бит – высокопроизводительное, оптимизированное и надежное шифрование, разработанное на основе технологии PGP Hybrid Cryptographic Optimizer (HCO). Проверен FIPS 140-2 (Federal Information Processing Standard).

Пользователи могут создавать зашифрованные контейнеры для переноса и совместного использования отдельных файлов с помощью приложений PGP® *Virtual Disk, PGP® Zip* и *PGP® Self-Decrypting Archive*.

***DriveCrypt*** DriveCrypt – бесплатная программа для шифрования данных, существует и расширенная платная версия программы DriveCrypt Plus Pack.

DriveCrypt Plus Pack использует для шифрования информации стойкий алгоритм AES-256 и хеш-функцию SHA-1 для генерирования ключей из пользовательских паролей.

Для аутентификации пользователь может использовать пароль, USBтокен либо пароль и токен одновременно.

Drive Crypt Plus Pack распространяется с закрытым исходным кодом, что можно считать недостатком.

*Возможности программы:*

* Шифрование любых физических дисков, а также сменных носителей информации.
* Шифрование системного диска.
* Создание скрытой операционной системы (невидимые разделы жесткого диска, в которых может быть расположена секретная операционная система).
* Использование шифрования «на лету». Информация никогда не находится на жестком диске в незашифрованном виде.
* Специальные функции защиты от троянских программ (кейлоггеров) и программ перебора паролей (брутфорс).
* Встроенный хранитель ключей. Все ключи, хранятся в зашифрованном контейнере, для доступа к которому необходим свой пароль. Файл ключей может маскироваться под медиафайл (музыка, изображение).

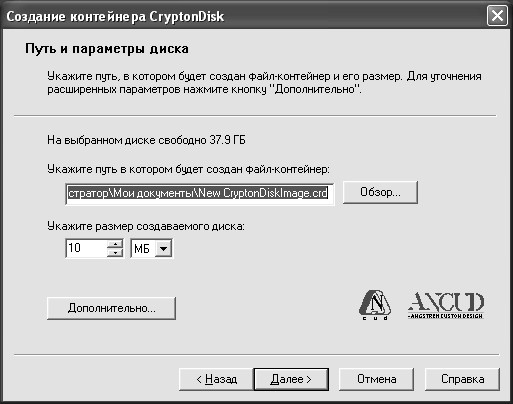
#### 3.6. СКЗИ от несанкционированного доступа

Средства защиты от несанкционированного доступа (НСД) – это программные и аппаратные средства защиты информации, позволяющие исключить возможность доступа злоумышленников или инсайдеров к персональным данным, коммерческой тайне или другой конфиденциальной информации. Рассмотрим способы защиты от НСД с помощью программных и аппаратных СКЗИ российского и иностранного производства.

***3.6.1. Программные СКЗИ***

##### Crypton Disk

СКЗИ Crypton Disk, разработанное фирмой «АНКАД», предназначено для защиты информации на дисках компьютера и защиты от несанкционированного доступа ресурсов компьютера.



***Рис. 3.9.*** *Окно создания контейнера*

Главной задачей программного шифратора является создание секретных логических дисков, содержимое которых шифруется в прозрачном (незаметном для пользователя) режиме и доступно только для владельца диска. При чтении какой-либо программной информации с секретного диска эта информация расшифровывается, а при записи – зашифровывается.

В состав аппаратно-программных комплексов входят модуль шифратора жесткого диска и подсистема защиты от НСД на базе аппаратно-программного модуля доверенной загрузки или криптобиблиотека Crypton Emulator и USBтокен.

Для решения задач обеспечения безопасности конфиденциальной информации данный комплекс имеет следующие возможности:

* работа с файл-контейнерами;
* шифрование информации по ГОСТ 28147-89;
* монтирование файлов-контейнеров как на букву, так и на папку;
* автоматическое подключение/отключение жестких дисков;
* маскировка непримонтированных контейнеров;
* хранение ключей шифрования на USB-токене РУТОКЕН;  наличие сторожевого таймера;
* блокировка по горячим клавишам.

Для полноценной работы СКЗИ Crypton Disk необходимо установить дополнительные компоненты:

1. Crypton Emulator – программное обеспечение, которое обеспечивает программную эмуляцию функций шифрования в операционных системах MS Windows, Linux и Solaris. Эмулятор обеспечивает шифрование по алгоритму ГОСТ 28147-78, по функциям эмулятор полностью совместим с УКЗД серии «КРИПТОН».
2. Crypton API – пакет программ, реализующих интерфейс между устройствами криптографической защиты данных и программами Win32 в операционных средах MS Windows.

СКЗИ Crypton Disk полностью поддерживает функции операционной системы MS Windows и является технологией шифрования виртуальных дисков.

В связи с тем, что изделие не является сертифицированным, использовать его можно для защиты информации, не составляющей государственную тайну.

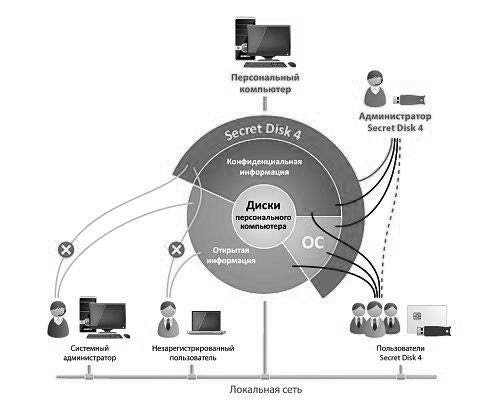
##### Secret Disk 4

Программный комплекс Secret Disk Российской компании «Аладдин Р.Д.» предназначен для защиты конфиденциальной информации и персональных данных от несанкционированного доступа, копирования, повреждения, кражи или принудительного изъятия.

Secret Disk обеспечивает защиту информации на логических дисках, отдельных жестких дисках, дисковых массивах (SAN, программных и аппаратных RAID-массивах), а также на съемных носителях (дискетах, Flash-дисках, CD, DVD, картах памяти). Сертифицированные версии обеспечивают выполнение требований законодательства по защите персональных данных и конфиденциальной информации.

СКЗИ Secret Disk 4 решает следующие задачи:

* защита от несанкционированного доступа и утечки конфиденциальной информации, хранящейся и обрабатываемой на персональном компьютере или ноутбуке;
* защита информации на съемных носителях;
* разграничение прав пользователей на доступ к защищенной информации с использованием надежной двухфакторной аутентификации (владение электронным ключом и знание пароля);
* сокрытие наличия на персональном компьютере конфиденциальных данных.



***Рис. 3.10.*** *Структурная схема работы СКЗИ Secret Disk 4*

Данные, хранящиеся на зашифрованных дисках персонального компьютера, доступны только администратору Secret Disk и пользователям, владеющими электронными ключами и зарегистрированными в Secret Disk 4. Остальные пользователи, включая системного администратора, не могут получить доступ к зашифрованным данным.

Защита конфиденциальной информации обеспечивается шифрованием данных «на лету» с помощью надежных алгоритмов шифрования. При записи данных на диск происходит их зашифрование, при чтении – расшифрование.

Находящиеся на зашифрованном диске данные всегда зашифрованы.

Защищенный диск можно подключать и отключать. Отключенный зашифрованный диск выглядит как неформатированный. Для того чтобы подключить зашифрованный диск, пользователь должен иметь USB-ключ или смарт-карту, знать его пароль и иметь право доступа к данному диску.

Для создания нового зашифрованного диска (или преобразования в зашифрованный диск уже существующего на компьютере диска с данными) необходимо иметь USB-ключ или смарт-карту с электронной лицензией на использование Secret Disk 4. Кроме того, в памяти электронного ключа должен находиться пользовательский сертификат открытого ключа и соответствующий ему закрытый ключ. Можно применять уже имеющиеся сертификаты (например, сертификат для входа в сеть и шифрования почты или электронную подпись). Если цифрового сертификата еще нет, то он может быть создан с помощью Secret Disk 4.

Secret Disk 4 позволяет защищать существующие диски (все разделы жесткого диска, включая системный), в том числе съемные, а также создавать так называемые виртуальные диски. Все содержимое виртуального диска хранится в одном файле-контейнере в зашифрованном виде. Подключенный виртуальный диск операционная система воспринимает как обычный диск. Файл подключенного виртуального диска защищен от удаления.

При создании зашифрованного диска пользователь может выбрать алгоритм шифрования диска из списка:

* AES и Twofish – более быстрые и надежные дополнительные алгоритмы шифрования, становятся доступными после установки пакета расширения

Secret Disk Crypto Extension Pack (рекомендуется установить);

* ГОСТ 28147-89 – при установке дополнительных поставщиков криптографии (криптопровайдеров) КриптоПро CSP, Signal-COM CSP или Vipnet CSP и пакета расширения Secret Disk Crypto Extension Pack;
* Triple DES и RC2 – поставляемые поставщиком службы криптографии

Microsoft Enhanced CSP, входящие в состав поставки ОС Windows и всегда доступные для использования.

Содержимое диска шифруется посекторно с использованием выбранного алгоритма шифрования и сгенерированного ключа шифрования диска.

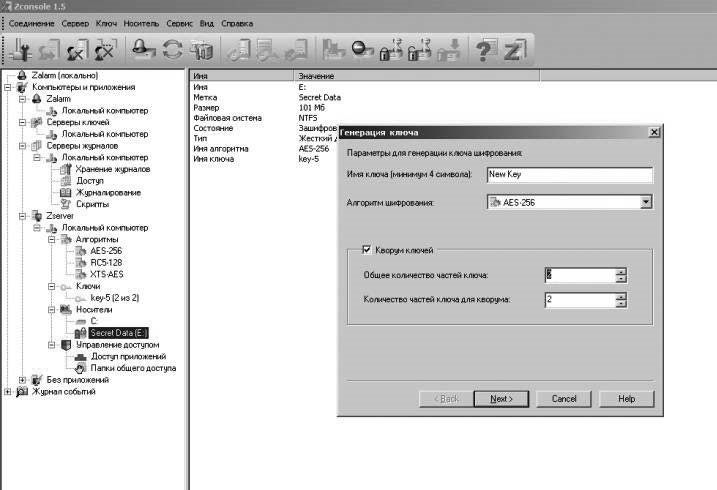
Процесс шифрования диска может быть приостановлен пользователем или даже прерван, например, из-за перебоев электропитания, однако это не повлечет за собой потерю данных. Приостановленный или прерванный процесс шифрования может быть возобновлен в любой удобный момент. По завершении процесса шифрования все содержимое диска становится зашифрованным, что обеспечивает надежную криптографическую защиту хранящихся на нем данных.

При прямом просмотре содержимое отключенного зашифрованного диска выглядит как случайная последовательность битов («белый шум»), поскольку все данные зашифрованы. По содержимому раздела диска невозможно определить, является ли данный раздел просто неформатированным, или же на нем имеется какая-то информация. Так Secret Disk обеспечивает защиту конфиденциальной информации от несанкционированного доступа, а также сокрытие наличия данных на компьютере.

Данный комплекс соответствует требованиям ФСТЭК России задания по безопасности, имеет оценочный уровень доверия «ОУД 1+» и может защищать сведения, не составляющие государственную тайну, а также персональный данные до УЗ 4 (т.е. обезличенные, общедоступные).

##### Zecurion Zserver

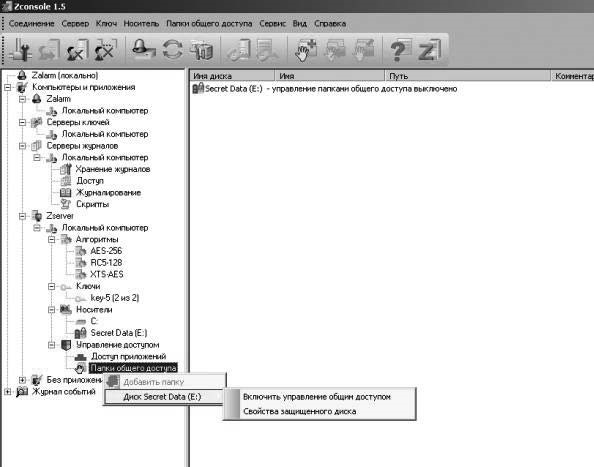
*Zecurion Zserver* относится к программам для шифрования данных, файлов и папок. Программа предназначена для надежной защиты данных, которые хранятся и обрабатываются на серверах и записываются на носители для резервного копирования. Программа осуществляет шифрование информации, размещенной на жестких дисках, на дисковых массивах (RAID-массивы любых конфигураций) и в SAN-хранилищах, методом прозрачного шифрования. Модуль для защиты резервных копий Zbackup осуществляет шифрование данных на магнитных лентах и оптических дисках.



***Рис. 3.11.*** *Окно выбора алгоритма шифрования в программе Zserver Suite*

Программа шифрования информации обеспечивает следующие функции:

* Защита конфиденциальной информации в местах хранения (в т.ч. шифрование диска).
* Защита баз данных, почтовых и файловых серверов в процессе работы.  Контроль доступа к защищаемой информации для пользователей и приложений.
* Полное скрытие факта наличия на защищаемых носителях каких-либо данных и ПО.
* Оперативное блокирование доступа к информации в случае экстренной необходимости.
* Гарантированное уничтожение информации.



***Рис. 3.12.*** *Окно управления доступом к логическому диску в программе Zserver Suite*

Для шифрования данных в Zserver используются современные стойкие алгоритмы шифрования с длиной ключа от 128 до 512 бит: RC5, AES, XTS-AES и ГОСТ 28147-89. Система не содержит встроенных криптографических средств, а модульная схема подключения внешних криптопровайдеров (в частности, КриптоПро) обеспечивает дополнительную гибкость.

Специализированные модули позволяют расширить функциональность Zecurion Zserver и установить индивидуальные настройки для критически важных параметров, таких как доступ к защищенным данным, реакция на сигнал «тревога» и другие события.

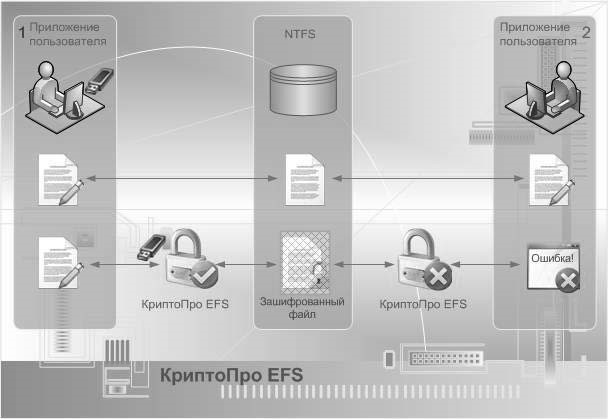
Изделие Zserver Suite не имеет сертификатов соответствия, но благодаря интеграции с сертифицированными криптопровайдерами может использоваться для защиты конфиденциальных сведений, не составляющих государственную тайну.

##### КриптоПро EFS

СКЗИ «КриптоПро EFS» предназначенo для обеспечения защиты конфиденциальной информации при ее хранении на ПЭВМ.

При использовании совместно с «КриптоПро CSP», «КриптоПро EFS» обеспечивает:

* конфиденциальность информации хранящейся на компьютере, шифрованием файлов файловой системы NTFS по алгоритму ГОСТ 28147-89;
* контроль целостности информации хранящейся на компьютере вычислением имитовставки в соответствии с ГОСТ 28147-89;
* организацию совместного доступа к данным зашифрованного файла ограниченной группе пользователей;
* организацию удаленной работы с защищенными файлами, размещаемыми в Web-папках (Web Distributed Authoring and Versioning – распределенная система хранения файлов с доступом через Web или WebDAV);
* возможность удаленной работы с зашифрованными файлами пользователей, использующих Terminal Services для удаленного входа в систему.
* восстановление данных в случае удаления пользователей из системы, компрометации или утраты закрытого ключа пользователя.



***Рис. 3.13.*** *Схема функционирования КриптоПро EFS*

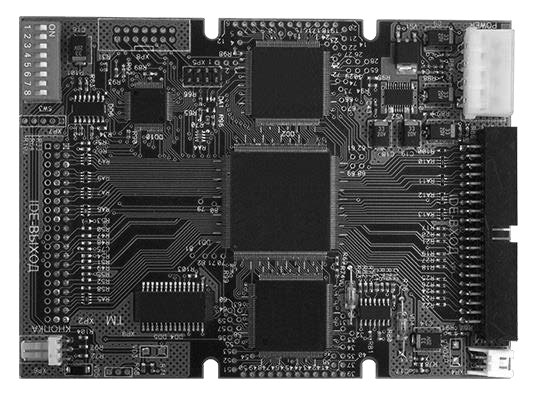
Для проведения криптографических операций КриптоПро EFS требуется, чтобы на рабочей станции был установлен криптопровайдер КриптоПро CSP 3.0 или 3.6. Для выработки сертификатов могут быть использованы «КриптоПро УЦ», MS CA. Пользователь КриптоПро EFS может хранить свой секретный ключ на отделяемых ключевых носителях – дискете, флэш-диске, смарткарте или USB-токене.

КриптоПро EFS не имеет сертификатов соответствия, но благодаря интеграции с сертифицированной криптобиблиотекой КриптоПро CSP может использоваться для защиты конфиденциальных сведений, не составляющих государственную тайну.

***3.6.2. Аппаратные СКЗИ***

##### КРИПТОН-ПШД

Криптографические комплексы М-590 («КРИПТОН – ПШД/IDE), М-575 («КРИПТОН – ПШД/SATA») и М-623 («КРИПТОН – ИНТЕГРАЛ») предназначены для защиты информации (в том числе сведений, составляющих государственную тайну) на дисках компьютера и защиты от несанкционированного доступа (НСД) ресурсов компьютера, разработаны фирмой «АНКАД».

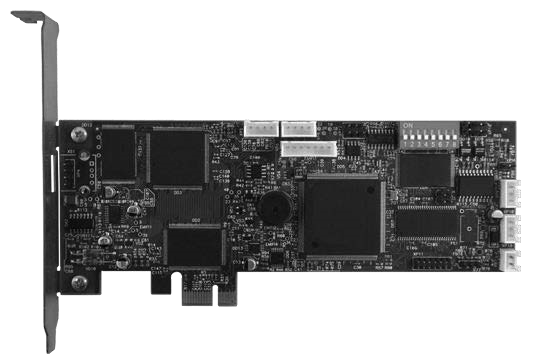


***Рис. 3.14.*** *Внешний вид изделия КРИПТОН-ПШД/IDE*

СКЗИ КРИПТОН-ПШД/IDE предназначено для обеспечения защиты информации на накопителе с жестким магнитным диском (НЖМД) путем автоматического криптографического преобразования данных при выполнении операций обмена между хост-контроллером на системной плате компьютера и НЖМД через внешний интерфейс, соответствующий стандарту AT Attachment with Packet Interface – 6 (ATA/ATAPI – 6). Основными возможностями изделия КРИПТОН-ПШД/IDE являются:  прозрачное шифрование данных по ГОСТ 28147-89 на жестком диске технологии SATA 2.0;

* защита от НСД к ключевой информации (КИ);
* функции электронного замка и доверенной загрузки;
* автоматическое перешифрование данных до старта компьютера.

Защиту от несанкционированного доступа к ключевой информации обеспечивает аппаратно-программный модуль доверенной загрузки (АПМДЗ) «КРИПТОН-ЗАМОК».



***Рис. 3.15.*** *Внешний вид изделия АПМДЗ «КРИПТОН-ЗАМОК»*

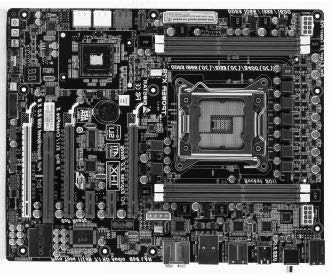
Загрузка ключей шифрования в устройство КРИПТОН-ПШД с идентификаторов Touch Memory (i-Button) производится напрямую, минуя ОЗУ и системную шину компьютера, что исключает возможность перехвата ключей.

СКЗИ КРИПТОН-ПШД/SATA имеет необходимые характеристики для работы с информацией конфиденциального характера (табл. 3.5).

*Таблица 3.5* **Характеристики изделия КРИПТОН-ПШД/IDE**

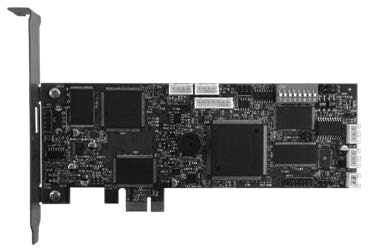
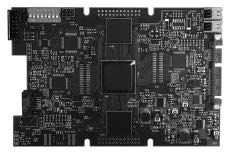
|  |  |
| --- | --- |
| Наименование параметра | Характеристика |
| Алгоритм шифрования | ГОСТ 28147-89 |
| Наличие сертификата соответствия | Сертификат ФСБ России № 020-1556 от 4 октября 2010 г. |
| Интерфейс жесткого диска | Стандарт AT Attachment with Packet Interface – 6 (ATA/ATAPI – 6) для IDE устройств и ANSI x3.298-1997 |
| Режимы работы | PIO 0-4, MWDMA 0-2, UltraDMA. |
| Скорость криптографического преобразования данных при записи и чтении с жесткого диска. | Не менее 6 Мбайт/сек |
| Датчик случайных чисел (ДСЧ), тип | Аппаратный |
| Носитель ключевой информации | Устройство памяти Touch Memory |

Комплекс «КРИПТОН-ПШД» совместно с АПМДЗ «КРИПТОН-ЗАМОК» позволяет создавать автоматизированное рабочее место, предназначенное для обработки и хранения сведений, составляющих государственную тайну (с грифом «Совершенно секретно» включительно). Схема подключения комплекса, предназначенного для защиты информации на жестких дисках, представлена на рисунке 3.16.



Материнская плата

компьютера



КРИПТОН

-

ПШД

/IDE

АПМДЗ «КРИПТОН

-

ЗАМОК»

Жёсткий диск

IDE



Идентификатор

iButton



***Рис. 3.16.*** *Схема подключения комплекса КРИПТОН-ПШД*

##### 3.6.3. Жесткие диски с технологией самошифрования

Диски с технологией самошифрования (Self-Encrypting Drives, SED) позволяют мгновенно осуществлять криптографическое преобразование информации, сделав ее нечитаемой, а также автоматически блокировать доступ к данным, если жесткий диск утерян или похищен из системы. Для таких изделий нет необходимости покупать дополнительные криптографические средства защиты информации и решать проблемы, которые могут возникнуть при встраивании сторонних СКЗИ (несоответствие с аппаратной архитектурой компьютера или с программной архитектурой операционной системы).

Принцип работы накопителя с технологией SED следующий: данные поступают на диск и шифруются перед записью с помощью специальной микросхемы (ASIC – Application Specific Integrated Circuit). Ключ шифрования данных сохраняется в защищенной, недоступной области диска. При чтении с диска данные предварительно расшифровываются и затем передаются с диска. Шифрование выполняется постоянно, его нельзя отключить. Во время работы операции шифрования на накопителях с технологией SED полностью прозрачны – аппаратно SED-накопители выглядят так же, как и диски без шифрования. Стоит заметить, что аппаратное шифрование не оказывает никакого влияния на производительность системы.

Наиболее известными разработчиками самошифрующихся дисков являются Seagate Technology, Hitachi, Western Digital, Samsung, Toshiba, Fujitsu.

##### Toshiba MKxx61GSYG

Подразделение компании Toshiba, специализирующееся на выпуске жестких дисков, разработало жесткий диск размером 2,5 дюйма со скоростью работы 7200 об/мин и функцией автоматического шифрования.

Самошифрующийся жесткий диск MKxx61GSYG предназначен для использования в бизнес-ноутбуках и других устройствах, характеризующихся повышенными требованиями к защите информации.



***Рис. 3.17.*** *Внешний вид жесткого диска Toshiba MKxx61GSYG*

Емкость дисков варьируется от 160 до 640 Gb, скорость вращения составляет 7200 оборотов в минуту, а буфер – 16 Mb. Это довольно распространенные параметры современных винчестеров для ноутбуков.

Данные шифруются с использованием алгоритма AES с 256-разрядным ключом, при введении пароля могут потребоваться особые микросхемы – «ключи», а механизм стирания информации включается при десятикратном вводе неверного пароля.

Toshiba MKxx61GSYG имеет необходимые характеристики для защиты сведений конфиденциального характера на ноутбуках.

##### Таблица 3.6 Характеристики изделия MKxx61GSYG

|  |  |
| --- | --- |
| Наименование параметра | Характеристика |
| Алгоритм шифрования | AES-256 (FIPS 197) |
| Максимальная емкость | 640 / 500 ГБ |
| 320 / 250 / 160 ГБ |
| Скорость вращения | 7200 об/мин |
| Объем буфера | 16 МБ |
| Интерфейс | Serial ATA 3.0 Gb/sec, Revision 2.6 (ATA-8) |
| Количество пластин | 2 |
| 1 |
| Наименование параметра | Характеристика |
| Среднее время поиска | 12 мс |
| Габариты | 69,85 x 100,0 x 9,5 мм |
| Масса | 115 г (модификации с двумя пластинами) или 98 г |

Жесткие диски компании Toshiba основаны на спецификации Opal Security Subsystem Class (Opal SSC) компании Trusted Computing Group (TCG). Достоинствами данных самошифрующихся дисков являются:

* повышенный уровень безопасности,
* простота внедрения, отсутствие накладных расходов в виде снижения производительности,
* прозрачность для приложений и операционных систем,
* совместимость с ведущими приложениями для управления системами безопасности ноутбуков и других клиентских компьютеров сторонних производителей.

###### Enterprise Capacity 3.5 HDD

Enterprise Capacity 3.5 HDD – жесткий диск с технологией самошифрования и объемом 6 ТБ, обеспечивает надежный доступ к большим массивам данных, стабильность и доступность. При этом он удобен в эксплуатации и демонстрирует высокую производительность.



***Рис. 3.18.*** *Внешний вид жесткого диска Enterprise Capacity 3.5 HDD*

Данные на диске Enterprise Capacity 3.5 HDD защищены от несанкционированного доступа в местах их хранения. Доступны варианты с технологией самошифрования, соответствующие стандарту FIPS 140-2 и поддерживающие безопасность данных на уровне, установленном законодательством, а также модели с технологиями самошифрования и мгновенного безопасного стирания, разработанные компанией Seagate для легкой и экономичной утилизации дисков.

Хост

Интерфейс

передачи

Самошифрующийся

диск

Seagate

AES-256



Данные

%y\*@

%y\*@

%y\*@

%y\*@

%y\*@



***Рис. 3.19.*** *Схема технологии дисков с самошифрованием SED*

Компания Seagate в корпоративной линейке Constellation ES.3 имеет накопители как с поддержкой SED (или SED по стандарту FIPS 140-2, доступное лишь для накопителей с интерфейсом SAS), так и без шифрования. Каждой модели соответствуют собственные технические характеристики. В таблице 3.7 представлены основные характеристики наиболее популярной модели ST6000NM0044.

##### Таблица 3.7 Характеристики изделия Enterprise Capacity 3.5 HDD

|  |  |
| --- | --- |
| Наименование параметра | Характеристика |
| Модель | ST6000NM0044 |
| Алгоритм шифрования | AES-256 (FIPS 140-2) |
| Максимальная емкость | 6 ТБ |
| Скорость вращения | 7200 об/мин |
| Объем буфера | 128 МБ |
| Интерфейс | Serial ATA 6.0 Gb/sec |
| Годовая интенсивность отказов | 0,63% |
| Средняя потребляемая мощность в рабочем режиме | 10,62 Вт |
| Габариты | 147,0 x 101,9 x 26,1 мм |
| Масса | 600 г |

###### MHZ2-CJ

Компания Fujitsu Europe Limited разработала серию жестких дисков MHZ2-CJ в форм-факторе 2,5", емкостью 320 ГБ и поддержкой автоматического аппаратного шифрования данных. HDD для компьютеров и внешних накопителей оснащены интерфейсом Serial ATA (SATA) сo скоростью передачи данных 3,0 Гб/с. Скорость вращения шпинделя: 7200 оборотов в минуту, что является самой высокой скоростью для данного промышленного класса.



***Рис. 3.20.*** *Внешний вид жесткого диска MHZ2-CJ*

Диск серии MHZ2-CJ поддерживает технологию 256-битного шифрования данных AES. Встроенная в чип процессора диска функция кодирования данных обеспечивает высокую степень защиты данных и улучшенную производительность по сравнению с программным шифрованием. Компаниям, работающим с большими объемами персональных и конфиденциальных данных, это позволит значительно снизить время и затраты на «очистку» жестких дисков компьютеров, которые предназначены для использования в других целях.

*Таблица 3.*8 **Характеристики изделия MHZ2-CJ**

|  |  |
| --- | --- |
| Наименование параметра | Характеристика |
| Алгоритм шифрования | AES-256 |
| Максимальная емкость | 320 / 250 / 160 / 120 / 80 ГБ |
| Скорость вращения | 7200 об/мин |
| Объем буфера | 16 МБ |
| Интерфейс | Serial ATA 3.0 Gb/sec |
| Среднее время доступа | Чтение – 10,5 мс |
| Запись – 12,5 мс |
| Средняя потребляемая мощность в состоянии холостого хода | 0,8 Вт |
| Габариты | 100,0 x 70,9 x 9,5 мм |
| Масса | 115 г |

*Преимущества* дисков с технологией самошифрования SED:

* Ключ шифрования независим от операционной системы, что исключает угрозы от вирусов и других атак.
* Специальное оборудование выполняет процесс шифрования без значительных задержек и снижения производительности операций ввода-вывода.

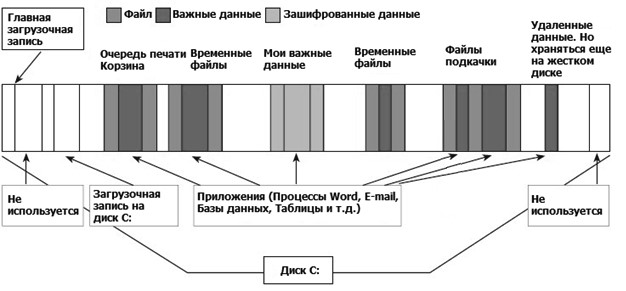
*Недостатки* дисков с технологией самошифрования SED:

* Необходимо приобрести новые диски SED (как правило, их стоимость выше, чем у других устройств) и выполнить их развертывание.
* Для обеспечения безопасности всей системы требуется заменить все установленные HDD и SSD дисками SED.
* Существующие данные необходимо перенести с установленных дисков без технологии шифрования на новые диски SED.
* Для управления несколькими дисками SED необходимо встроенное программное решение для управления ключами.
* Диски SED содержат ключ, который невозможно удалить, поэтому существует угроза раскрытия данных при утилизации диска или возврате поставщику.

###### 3.6.4. Практические рекомендации

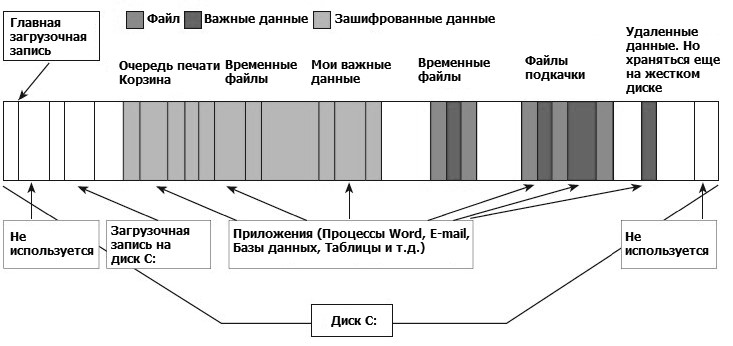
Из рассмотренных криптографических методов защиты информации на жестких дисках от несанкционированного доступа можно выделить наиболее популярные среди разработчиков СКЗИ.

*Метод пофайлового шифрования* применяется в случае, когда зашифрованное сообщение передается через открытые сети. Использовать его для защиты данных на жестких дисках не рекомендуется в связи с тем, что данная технология не шифрует временные файлы компьютера. Из рассмотренных средств данный метод шифрования применяют программные комплексы Zserver Suite и КриптоПро EFS.



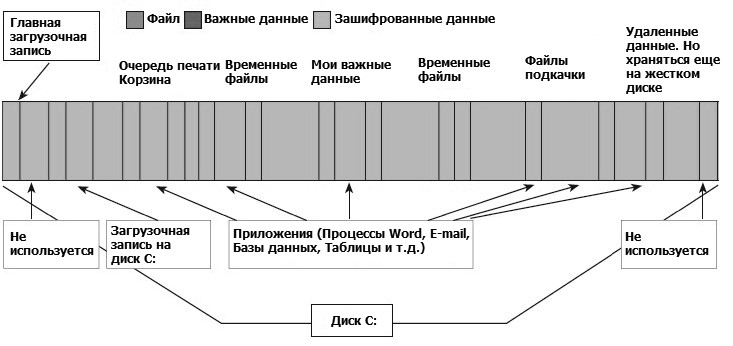
***Рис. 3.21.*** *Схема технологии пофайлового шифрования данных*

*Метод создания виртуальных логических дисков* наиболее часто применяется именно для защиты личных файлов на компьютере, но, в свою очередь, в значительной степени сказывается на производительности компьютера. К программным комплексам, использующих метод создания шифрованных дисков, относятся Crypton Disk и Secret Disk.



***Рис. 3.22.*** *Схема технологии шифрования папок и виртуальных дисков*

*Метод шифрования всего диска* является самым надежным способом защиты сведений конфиденциального характера (в том числе сведений, составляющих государственную тайну). Данный метод применяется в программноаппаратном комплексе КРИПТОН-ПШД и жестких дисках с технологией самошифрования.



***Рис. 3.23.*** *Схема технологии шифрования всего диска*

Наиболее доступным, распространенным и простым в использовании является *программное шифрование*. Необходимо учитывать потенциальную возможность обнаружения ключей злоумышленником в случае, если ключи хранятся на персональном компьютере. В зависимости от характера защищаемой информации, необходимо выбирать СКЗИ: свободно распространяемые или коммерческие (платные).

*Аппаратное шифрование* имеет преимущество перед программным в отношении скорости шифрования (как правило, аппаратный модуль имеет собственный процессор, специально ориентированный на операции шифрования/расшифрования данных); также аппаратные модули защищены от уязвимостей внешнего программного обеспечения и операционных систем, поскольку не зависят от используемой операционной системы и внешнего программного обеспечения. Недостатками аппаратного шифрования являются ограниченная совместимость (некоторые модули невозможно установить на старые или, наоборот, новые компьютеры) и сложность обновления (при обнаружении изъяна в оборудовании, как правило, потребуется его замена).

###### Вопросы и задания для самостоятельной работы

1. Каковы с точки зрения информационной безопасности преимущества и недостатки персонального компьютера?
2. Чем существенно отличаются общие подходы к обеспечению защиты ПК в отличие от систем обработки данных с большими ЭВМ?
3. Назовите три вида средств, используемых для несанкционированного доступа.
4. В чем разница технологий шифрования, использующих одинаковые криптографические алгоритмы?
5. Какие алгоритмы шифрования или их реализация являются оптимальными?
6. Почему кэширование ключей является одним из определяющих факторов при оценке технологии шифрования?
7. В чем особенность шифрования жестких дисков на ПК?
8. Какая технология шифрования позволяет защитить файлы операционной системы и в чем ее уязвимость?
9. Какие технологии шифрования уязвимы к внутренним атакам?
10. Какими возможностями должно обладать СКЗИ для защиты коммерческих данных, размещенных на общих серверах?
11. Какой подход к генерации ключа шифрования на ПК считается оптимальным и почему?
12. Какие СКЗИ встроены в ОС Windows, и в чем принципиальное отличие их технологий?
13. Чем определяется доступ к криптографическим алгоритмам в

Microsoft Office?

1. Какие криптографические задачи можно решить с помощью СКЗИ свободного доступа, и в каких случаях их использование недопустимо?
2. Приведите пример свободно доступных СКЗИ с механизмом цифровой подписи, позволяющих шифровать электронную почту.
3. В каких случаях на ПК используется метод пофайлового шифрования, и почему он не рекомендуется для защиты жестких дисков?
4. Какое влияние на производительность ПК оказывает защита методом создания виртуальных логических дисков?
5. Какой метод шифрования на ПК является самым надежным способом защиты сведений конфиденциального характера (в том числе сведений, составляющих государственную тайну)?
6. Какие преимущества и недостатки имеет аппаратное шифрование на ПК по сравнению с программным?
7. Какой способ загрузки ключей в СКЗИ исключает возможность их перехвата.
8. В чем уязвимость жестких дисков с технологией самошифрования SED?
9. Приведите пример аппаратных СКЗИ для защиты от несанкционированного доступа.

## ТЕМА4. СРЕДСТВА КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ СЕТЕВОГО ВЗАИМОДЕЙСТВИЯ

**4.1. Криптографические средства создания защищенных**

### виртуальных сетей (VPN)

Основными типами атак на сетевое взаимодействие являются:

* *анализ сетевого трафика;*
* *подмена доверенного объекта в сети;*  *отказ в обслуживании*.

Анализ сетевого трафика – это несанкционированное подключение к каналам связи, осуществление перехвата (прослушивания) информации. При этом нарушается, в основном, *конфиденциальность* информации.

«Подмена доверенного объекта в сети» – это модификация (подмена) передаваемых по каналам данных. В результате нарушается *целостность* сообщения, компрометируется *аутентификация* сообщения и сторон взаимодействия.

«Отказ в обслуживании» приводит к нарушению *доступности* информации.

Защититься от отказа в обслуживании можно различными способами, например правильной настройкой сетевого соединения или аутентификацией источника по криптографическим протоколам или алгоритмам. Предотвратить атаки на целостность и конфиденциальность информации, обеспечить аутентификацию как самой информации при ее передаче по сетям общего пользования, так и взаимодействующих сторон, можно только с помощью криптографической защиты информации.

*Шифрование и аутентификация* – единственные способы, с помощью которых можно получить достоверную информацию от проверенных лиц, – реализованы в СКЗИ VPN.

**VPN (Virtual Private Network** – виртуальная защищенная сеть) – обобщенное название технологий, позволяющих обеспечить одно или несколько сетевых соединений (логическую сеть) поверх другой сети (например, Internet) с помощью криптографических средств.

Уровень доверия к построенной логической сети не зависит от уровня доверия к базовым сетям благодаря использованию *средств криптографии (шифрования, аутентификации, инфраструктуры открытых ключей, средств для защиты от повторов и изменений*, передаваемых по логической сети сообщений).

Защитить информацию от несанкционированного доступа при ее передаче по сетям общего пользования возможно только криптографическими средствами, средствами создания VPN-соединений.

### 4.2. Технология построения криптозащищенных туннелей

VPN позволяет обеспечить одно или несколько сетевых соединений (логическую сеть) поверх общедоступной сети (например, Интернет) посредствам *туннелирования.*

Туннелирование в компьютерных сетях – процесс, в ходе которого создается *криптозащищенное логическое соединение* между двумя конечными точками посредством инкапсуляции различных протоколов.

Это метод построения сетей, при котором один сетевой протокол инкапсулируется (упаковывается) в другой (инкапсулируемый протокол относится к тому же или более низкому уровню сетевого взаимодействия, чем используемый в качестве тоннеля).

Суть туннелирования состоит в том, чтобы «упаковать» передаваемую порцию данных, вместе со служебными полями, в новый зашифрованный «конверт» для обеспечения конфиденциальности и целостности всей передаваемой порции, включая служебные поля.

Сеть

1

NetBIOS

Сеть

2

IP

Сеть

3

NetBIOS

Маршрутизатор

1

Маршрутизатор

2



Кадр

Ethernet

Кадр

NetBIOS

Кадр

Ethernet

Кадр

IP

Кадр

NetBIOS

Кадр

Ethernet

Кадр

NetBIOS

***Рис. 4.1.*** *Принцип туннелирования*

VPN состоит из двух частей: «внутренняя» сеть, которая «запаковывается» в пакеты второй сети – «внешней», по которой проходит *инкапсулированное соединение* (обычно используется Интернет).

Подключение удаленного пользователя к VPN производится посредством сервера доступа, который подключен как к внутренней, так и к внешней (общедоступной) сети. Сервер доступа требует прохождения процесса идентификации, а затем процесса аутентификации, то есть происходит процесс авторизации.

### 4.3. Криптографическая защита удаленного доступа к локальной сети

Для защиты данных, передаваемых по указанным каналам связи, необходимо использовать соответствующие средства криптографической защиты. Криптопреобразования могут осуществляться как *на прикладном уровне* (или на уровнях между протоколами приложений и протоколом TCP/IP), так и на *сетевом* (преобразование IP-пакетов).

***На прикладном уровне***: средства криптографической защиты подключаются к прикладным программам или коммуникационной части операционной системы, что, как правило, требует больших затрат и установки соответствующих средств защиты на каждый узел локальной сети.

К решениям данного варианта относятся протоколы *SSL, S-HTTP, S/MIME, PGP/MIME*, которые обеспечивают шифрование и цифровую подпись почтовых сообщений и сообщений, передаваемых с использованием протокола HTTP.

***На сетевом уровне***: специальные криптографические средства устанавливаются в точках подключения локальных сетей и удаленных абонентов к каналам связи (сетям общего пользования), проходящим по неконтролируемой территории. При этом необходимо обеспечить требуемый уровень криптографической защиты данных и минимально возможные дополнительные задержки при их передаче, так как эти средства туннелируют передаваемый трафик (добавляют новый IP-заголовок к туннелируемому пакету) и используют различные по стойкости алгоритмы шифрования.

Средства, обеспечивающие криптопреобразования на сетевом уровне, полностью совместимы с любыми прикладными подсистемами, работающими в корпоративной информационной системе (являются «прозрачными» для приложений), поэтому они наиболее часто и применяются.

Средства криптографической защиты информации, планируемые к применению в государственных структурах, должны быть сертифицированы в России.

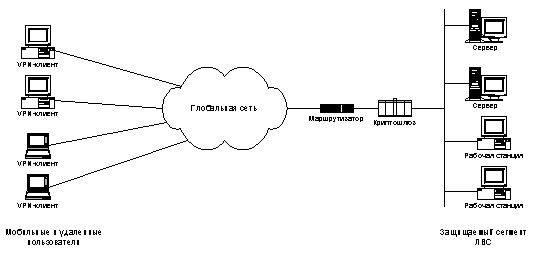
***Решения на базе сертифицированных криптошлюзов***

Для обеспечения конфиденциальности и достоверности информации, передаваемой между объектами компании по каналам связи, можно использовать сертифицированные *криптошлюзы (VPN-шлюзы),* работающие *на прикладном уровне.*Криптошлюзы позволяют осуществить защищенный доступ удаленных абонентов к ресурсам корпоративной информационной системы.

*Например,* **Континент-К, VIPNet TUNNEL, ЗАСТАВА-Офис компаний НИП «Информзащита», Инфотекс, Элвис+.**

Эти устройства обеспечивают шифрование передаваемых данных (IPпакетов) в соответствии с ГОСТ 28147-89, а также скрывают структуру локальной сети, защищают от проникновения извне, осуществляют маршрутизацию трафика и имеют сертификаты ФСТЭК и ФСБ. Доступ производится с использованием специального программного обеспечения, которое устанавливается на компьютер пользователя (VPN-клиент) для осуществления защищенного взаимодействия удаленных и мобильных пользователей с криптошлюзом.

Программное обеспечение криптошлюза (сервер доступа) проводит идентификацию и аутентификацию пользователя и осуществляет его связь с ресурсами защищаемой сети.



***Рис. 4.2.*** *Схема удаленного доступа через криптошлюз*

Криптошлюзы могут быть реализованы как *программно*, так и в виде *аппаратно-программных* комплексов.

#### Решения на базе протокола IPSec

Протокол *IPSec (IP Security)* является базовым для построения систем безопасности *сетевого уровня,* представляет собой набор открытых международных стандартов и поддерживается большинством производителей решений по защите сетевой инфраструктуры.

Протокол IPSec позволяет организовать на сетевом уровне потоки *защищенных и аутентичных данных* (IP-пакетов) между различными взаимодействующими узлами, включая компьютеры, межсетевые экраны, маршрутизаторы. Он обеспечивает:

* аутентификацию, шифрование и целостность передаваемых данных

(IP- пакетов);

* защиту от повторной передачи пакетов (replay attack);
* создание, автоматическое обновление и защищенное распространение криптографических ключей;
* использование широкого набора алгоритмов шифрования;
* аутентификацию объектов сетевого взаимодействия на базе цифровых сертификатов.

*Алгоритмы шифрования* по протоколам IPSec: **DES, 3DES, AES.** *Механизмы контроля целостности данных***: MD5, SHA-1.**

Существуют программные реализации протокола IPSec, использующие российские алгоритмы шифрования (ГОСТ 28147-89), хеширования (ГОСТ Р 34.11-94), электронной цифровой подписи (ГОСТ Р 34.10-2001). Однако большинство из сертифицированных криптошлюзов не поддерживает протокол IPSec, и поэтому они функционально несовместимы с аппаратно-программными продуктами других производителей.

Текущий набор стандартов IPSec включает в себя базовые спецификации, определенные в документах RFC (RFC 2401-2412, 2451). *Request for Comments (RFC)* – это серия документов группы *Internet Engineering Task Force (IETF)*, начатая в 1969 году и содержащая описания набора протоколов Internet.

Архитектура системы (рис. 4.3) определена в RFC 2401 «Security Architecture for Internet Protocol», а спецификации основных протоколов в следующих RFC:

* RFC 2402 «IP Authentication Header» – спецификация протокола AH, обеспечивающего целостность и аутентификацию источника передаваемых IPпакетов;
* RFC 2406 «IP Encapsulating Security Payload» – спецификация протокола ESP, обеспечивающая конфиденциальность (шифрование), целостность и аутентификацию источника передаваемых IP-пакетов;

Протокол согласования

параметров

виртуального канала и

управления ключами

)

ISAKMP

(

Протокол

аутентифицирующего

заголовка

(

АН

)

Протокол

инкапсулирующей

защиты содержимого

)

ESP

(

Алгоритмы

согласования

параметров и

управления ключами

Алгоритмы

аутентификации

Алгоритмы

шифрования

Домен

интерпретации

***Рис. 4.3.*** *Архитектура протокола IPSec*

ISAKMP – *Internet Security Association and Key Management Protocol*

AH – *Authentication header*

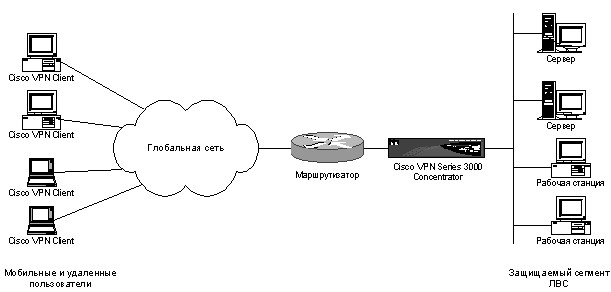
ESP – *Encapsulating Security Payload*

* RFC 2408 «Internet Security Association and Key Management Protocol» – спецификация протокола ISAKMP, обеспечивающего согласование параметров, создание, изменение, уничтожение защищенных виртуальных каналов (Security Association – SA) и управление необходимыми ключами;
* RFC 2409 «The Internet Key Exchange» – спецификация протокола IKE

(включает в себя ISAKMP), обеспечивающего согласование параметров, создание, изменение и уничтожение SA, согласование, генерацию и распространение ключевого материала, необходимого для создания SA.

Протокол IPSec для обеспечения безопасного сетевого взаимодействия использует симметричные алгоритмы шифрования и соответствующие ключи. Механизмы генерации и распространения таких ключей предоставляет *протокол IKE (Internet Key Exchange).*

*Например,* организация удаленного доступа по защищенному каналу компанией Cisco Systems с использованием VPN-концентратора на базе протокола IPSec (рис. 4.4): на компьютер удаленного пользователя устанавливается специальное программное обеспечение Cisco VPN Client. Существуют версии данного программного обеспечения для различных операционных систем – MS Windows, Linux, Solaris.



***Рис. 4.4.*** *Схема удаленного доступа через Cisco VPN-концентратор*

Cisco VPN Client взаимодействует с Cisco VPN Series 3000 Concentrator и создает защищенное соединение, которое называется *IPSec-туннелем****,*** между компьютером пользователя и локальной сетью, находящейся за VPN-концентратором.

К недостаткам такого решения можно отнести отсутствие поддержки компанией Cisco Systems российских алгоритмов шифрования, хеширования и электронной цифровой подписи.

### 4.4. СКЗИ для передачи данных в локальных сетях

#### Crypton IPMobile (компания АНКАД)

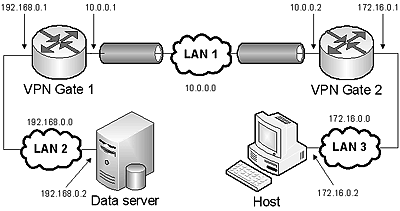
Комплекс *Crypton IPMobile* предназначен для организации виртуальной защищенной сети (VPN) c шифрованием сетевого трафика по алгоритму *ГОСТ 28147-89 и контролем целостности*.Имеет Сертификат ФСБ СФ/124-1873 от 17 июня 2012 г.

*Основные возможности Crypton IP Mobile:*

* Интегральная скорость шифрования сетевого трафика в боевом режиме работы на 200 Mbit-ном интерфейсе составляет 160 Mbit/sec.
* Фильтрация враждебного IP трафика. Встроенная защита от подмены, навязывания, компрометации, DoS-атак и переполнения буфера – работает в режиме *криптомаршрутизатора*.
* Сборка фрагментированных IP-пакетов.
* Централизованное и децентрализованное администрирование.
* Сбор и просмотр статистики по каждому туннелю. Гибкая система ведения и просмотра журналов.
* Совместимость с Microsoft VPN.
* Не требует установки соединения.
* Поддержка однонаправленных туннелей.

Crypton IP Mobile состоит из следующих компонентов:

* VPN Client – средство индивидуальной защиты пользователя
* VPN Gate – средство защиты подсетей
* VPN Server – средство индивидуальной защиты сервера  Local Admin – средство локального администрирования  ЦГК – цетр генерации ключей



***Рис.***

***4.5.***

*Схема построение туннеля*

Криптографическим ядром Crypton IPMobile является аппаратный шифратор КРИПТОН-4К/16 (в программно-аппаратной конфигурации) или программный шифратор Crypton LITE (в программной конфигурации).



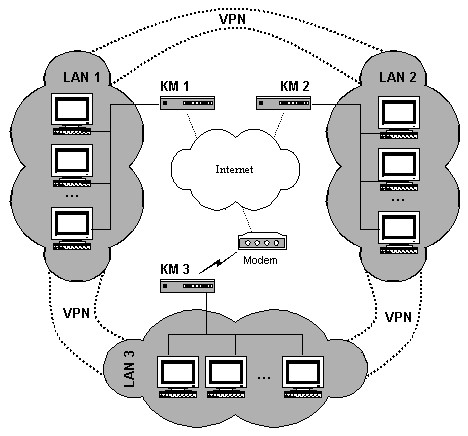
***Рис. 4.6.*** *Аппаратный шифратор КРИПТОН-4К/16*

*Таблица 4.1*

**Основные функционально-технические характеристики**

**Crypton IPMobile**

|  |  |
| --- | --- |
| Наименование характеристики | Показатель характеристики |
| Шифрование данных | Алгоритм ГОСТ 28147-89 |
| Скорость шифрования данных | 1,5 Мбайт/с |
| Электронная цифровая подпись | Алгоритм ГОСТ Р  34.10/11 |
| Производительность шифрования пакетов данных (при длине пакета 1,5 К и симметричной ключевой системе) | 1,2–1,8 Мбит/с |
| Носители ключей | Дискеты, смарт-карты |



***Рис. 4.7.*** *Схема подключения локальной компьютерной сети к глобальной сети (Internet) с использованием Crypton IPMobile (КМ)*

Криптомаршрутизатор (КМ) Crypton IPMobile выполняет функции статического фильтра пакетов данных и обеспечивает контроль прохождения всего IP-трафика. Наиболее эффективно использовать КМ в качестве шифратора на проходе, установленного в разрыве между ЛВС и маршрутизатором доступа, функции которого может выполнять и сам КМ при наличии статической IPадресации в сети.

КМ Crypton IPMobile обеспечивает прозрачное шифрование / расшифрование трафика, что позволяет использовать открытые каналы связи (в том числе сеть Internet) для организации защищенной сети.

Все КМ VPN авторизуются: каждый имеет уникальный ключ, известный другим КМ. Поскольку КМ обрабатывает и пропускает в локальную сеть только пакеты, приходящие от авторизованных КМ, то локальные сети оказываются защищенными от вторжения извне.

#### ViPNet (Компания ИнфоТеКС)

ViPNet Client (Клиент) – это программный комплекс, выполняющий на рабочем месте пользователя или сервере с прикладным ПО функции VPNклиента, персонального экрана, клиента защищенной почтовой системы, а также криптопровайдера для прикладных программ, использующих функции подписи и шифрования.

Клиент состоит из набора взаимосвязанных программных модулей:

 ViPNet [Монитор] – совместно с низкоуровневым драйвером шифрования и фильтрации трафика отвечает за реализацию функций:

* Персонального сетевого экрана; o Шифратора IP-трафика.  ViPNet [Деловая Почта]:
* Встроенные механизмы ЭЦП для подписи, в том числе множественной,

текста письма и его вложений; o Квитирование писем.

* ViPNet MFTP – обновление ключей, справочников и программного обеспечения.
* Криптопровайдер для вызова криптографических функций и их использования Web-приложениями, а также низкоуровневый С-интерфейс к криптографическим функциям для встраивания в приложения заказчика.

Криптопровайдер ViPNet выполняет следующие задачи:

* Авторизация и обеспечение подлинности документов в процессе защищенного документооборота. Для этого используются средства формирования и проверки электронной подписи в соответствии со стандартами ГОСТ P 34.11-94, ГОСТ Р 34.11-2012, ГОСТ P 34.10-2001 и ГОСТ P 34.10-2012.
* Обеспечение конфиденциальности и контроля целостности информации путем ее шифрования и имитозащиты в соответствии с ГОСТ 28147-89.
* Обеспечение аутентичности и конфиденциальности соединений TLS/SSL.

#### Hamachi (Разработчик LogMeIn Inc)

Hamachi позволяет создать собственную защищенную сеть из компьютеров, соединенных через Интернет, как будто они соединены одной физической локальной сетью.

*Таблица 4.2* ***Сравнение программных средств построения VPN-туннелей***

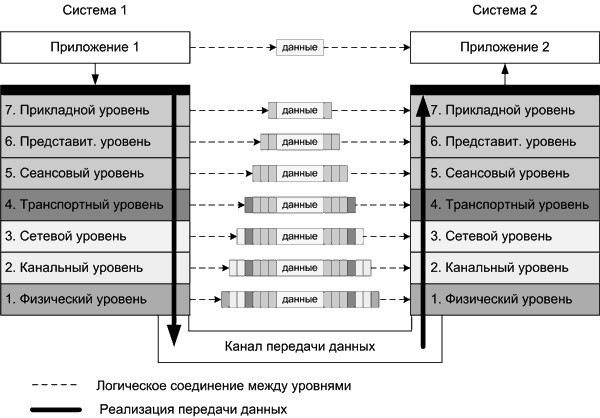
|  |  |  |  |
| --- | --- | --- | --- |
|  | ViPNet | Hamachi | Crypton IPMobile |
| Алгоритм шифрования | ГОСТ 28147  – 89 | AES-256 | ГОСТ 28147  – 89 |
| Протокол | SSL | SSL | PPTP |
| Аутентификация | TLS + ГОСТ  34.10/11 | RSA | ГОСТ 34.10/11 |
| Есть функция Firewall | + | + | + |
| Наличие сертификата ФСТЭК России | + | - | + |

### 4.5. Сетевые протоколы криптографической защиты

Сетевая модель OSI (англ. open systems interconnection basic reference model – эталонная модель взаимодействия открытых систем, сокр. ЭМВОС; 1978 г.) – модель стека сетевых протоколов OSI/ISO (рис. 4.8) (ГОСТ Р ИСО/МЭК 7498-1-99).

|  |  |  |
| --- | --- | --- |
|  | **Модель OSI** | |
| **Тип данных** | **Уровень (layer)** | **Функции** |
| Данные | 7. Прикладной (application) | Доступ к сетевым службам |
| 6. Представительный (presentation) | Представление и шифрование данных |
| 5. Сеансовый (session) | Управление сеансом связи |
| Сегменты | 4. Транспортный  (transport) | Прямая связь между конечными пунктами и надежность |
| Пакеты | 3. Сетевой (network) | Определение маршрута и логическая адресация |
| Кадры | 2. Канальный (data link) | Физическая адресация |
| Биты | 1. Физический (physical) | Работа со средой передачи, сигналами и двоичными |

***Рис. 4.8.*** *Эталонная модель взаимодействия открытых систем OSI/ISO*



***Рис. 4.9.*** *Схема работы модели OSI/ISO*

Виртуальный защищенный канал можно построить с помощью криптографических средств VPN, реализованных на разных уровнях модели OSI – *ка-*

|  |  |
| --- | --- |
| *нальном, сетевом и сеансовом*. | От выбранного рабочего уровня OSI зависит |
| *функциональность реализуемой VPN* и ее совместимость с приложениями кор- | |

поративной информационной системы, а также с другими средствами защиты.

Средства VPN, применяемые на ***канальном уровне*** модели OSI, позволяют обеспечить инкапсуляцию различных видов трафика третьего уровня (и выше) и построение виртуальных туннелей типа «точка–точка» (от маршрутизатора к маршрутизатору или от персонального компьютера к шлюзу ЛВС).

Протоколы туннелирования канального уровня модели OSI – это протоколы ***РРТР (Point-to-Poin Tunneling Protocol), L2F (Layer-2 Forwarding) и L2TP (Layer-2 Tunneling Protocol).***

Общим свойством этих протоколов является то, что они используются для организации защищенного многопротокольного удаленного доступа к ресурсам корпоративной сети через открытую сеть, например через Интернет.

Все три протокола – РРТР, L2F и L2TP – обычно относят к протоколам формирования защищенного канала, однако этому определению точно соответствует только протокол *РРТР*, который обеспечивает *туннелирование и шифрование* передаваемых данных. Протоколы L2F и L2TP поддерживают только функции туннелирования. Для защиты туннелируемых данных в этих протоколах необходимо использовать некоторый дополнительный протокол, в частности *IPSec****.***

При построении защищенных виртуальных сетей на ***сеансовом уровне*** появляется возможность криптографической защиты информационного обмена, включая аутентификацию, а также реализации ряда функций посредничества между взаимодействующими сторонами.

*Сеансовый уровень модели OSI* отвечает за установку *логических соединений* и управление этими соединениями. Поэтому существует возможность применения на этом уровне программ-посредников, проверяющих допустимость запрошенных соединений и обеспечивающих выполнение других функций защиты межсетевого взаимодействия.

Однако на сеансовом уровне начинается непосредственная *зависимость от приложений*, реализующих высокоуровневые протоколы. Поэтому реализация протоколов защиты информационного обмена, соответствующих этому уровню, в большинстве случаев требует внесения изменений в высокоуровневые сетевые приложения.

Для защиты информационного обмена на сеансовом уровне широкое распространение получил протокол ***SSL (Secure Sockets Layer)***. Для выполнения на сеансовом уровне функций посредничества между взаимодействующими сторонами организацией IETF (Internet Engineering Task Force) в качестве стандарта принят протокол ***SOCKS***.

Радикальное устранение уязвимостей компьютерных сетей возможно при создании системы защиты не для отдельных классов приложений, а *для сети в целом*, т.е. необходимо действовать ***на сетевом уровне*** *модели OSI*. При формировании VPN на сетевом уровне модели OSI достигается *оптимальное соотношение между прозрачностью и качеством защиты.*

На сетевом уровне работает стек протоколов ***IPSec*** *(Internet Protocol Security)* – для *аутентификации участников обмена, туннелирования трафика и шифрования* IP-пакетов.

Архитектура IPSec (рис. 4.3) *совместима с протоколом IPv4* и новой версии *IPv6*, которая постепенно внедряется в Internet.

#### Протокол PPTP

Протокол РРТР (Point-to-Point-Tunneling Protocol) представляет собой расширение протокола РРР (Point-to-Point Protocol) для создания защищенных виртуальных каналов при доступе удаленных пользователей к локальным сетям через Internet. Он предусматривает создание криптозащищенного туннеля *на канальном уровне модели* OSI.

Для передачи данных применяются IP-пакеты, содержащие инкапсулированные РРР-пакеты. Инкапсулированные РРР-пакеты в свою очередь включают *зашифрованные инкапсулированные исходные пакеты,* предназначенные для взаимодействия между компьютером удаленного пользователя и локальной сетью.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Заголовок канального уровня | Заголовок IP | Заголовок GRE | |  | | --- | | IP пакет |   РРР пакет |

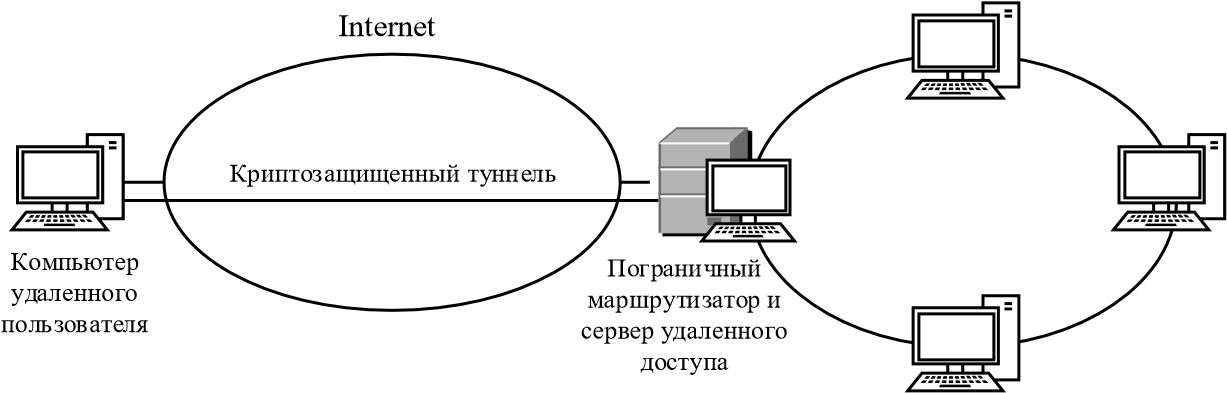
***Рис. 4.9.*** *Структура пакета РРТР*

Для аутентификации могут использоваться различные протоколы, поддерживаются протоколы аутентификации *PAP (Password Authentication Protocol* – протокол распознавания пароля) и *CHAP (Challenge-Handshaking Authentication Protocol* – протокол распознавания при рукопожатии).

При использовании *протокола РАР* идентификаторы и пароли передаются по линии связи в незашифрованном виде.

При использовании же *протокола CHAP* каждый пароль для передачи по линии связи шифруется на основе случайного числа, полученного от сервера. Такая технология обеспечивает защиту от повторного использования злоумышленником перехваченных пакетов с зашифрованным паролем.

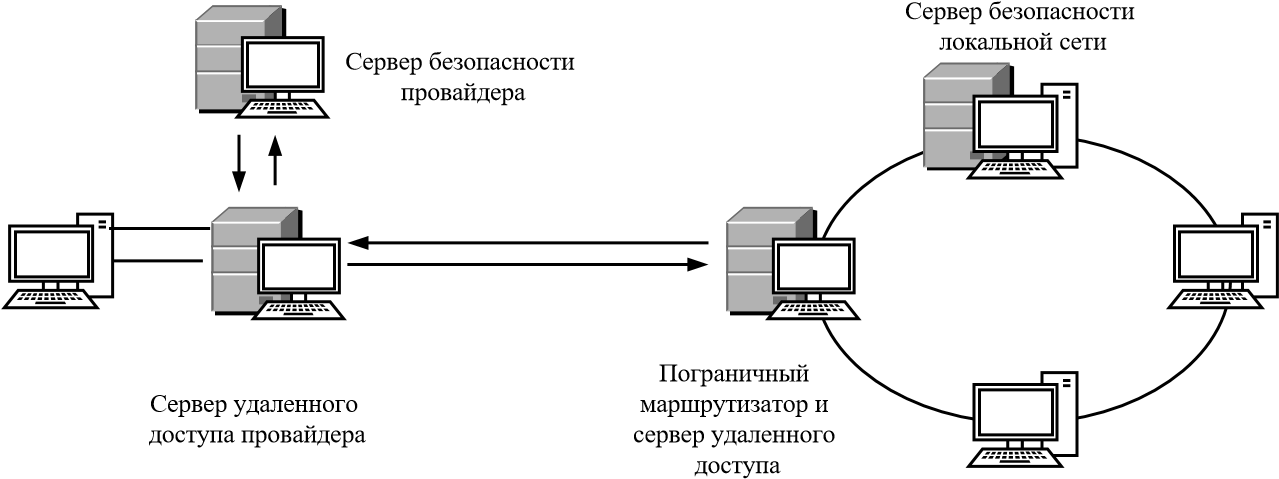
Для шифрования соединения может использоваться любой протокол шифрования, данная функция настраивается в программном обеспечении РРТР соединения.



***Рис. 4.10.*** *Схема туннелирования с помощью протокола РРТР*

#### Протокол L2F

Протокол L2F (Layer-2 Forwarding) разработан в качестве альтернативы протоколу РРТР для создания защищенных виртуальных сетей на канальном уровне модели OSI.



***Рис. 4.11.*** *Схема туннелирования с помощью протокола L2F*

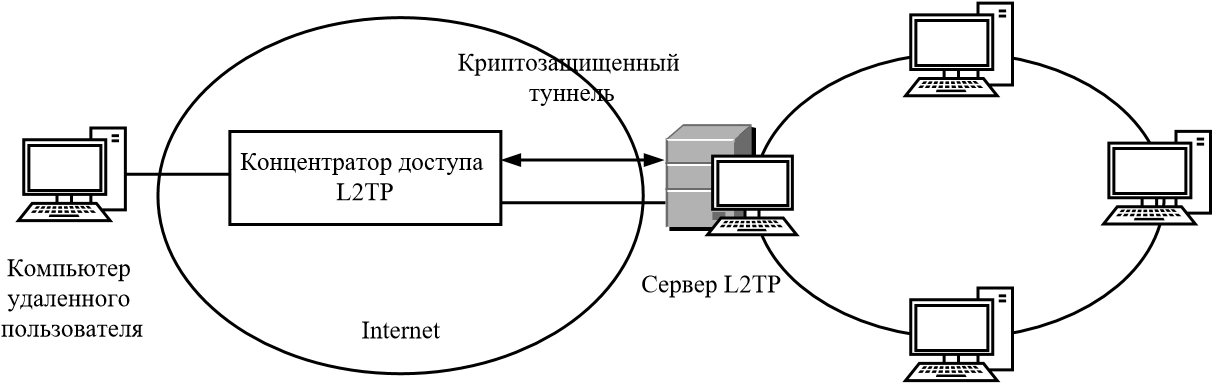
Виртуальный защищенный канал может быть создан только между сервером удаленного доступа провайдера и пограничным маршрутизатором локальной сети, а участок между компьютером удаленного пользователя и сервером провайдера остается открытым.

Различия между L2F и РРТР объясняются специализацией их разработчиков. Cisco (разрабатывало L2F) производит аппаратные маршрутизаторы для сетевой инфраструктуры, тогда как Microsoft (разработчик РРТР) выпускает операционные системы.

#### Протокол L2TP

Протокол L2TP предусматривает 3 этапа формирования защищенного канала:

1. Установление соединения с сервером удаленного доступа локальной сети.
2. Аутентификация пользователя.
3. Конфигурирование криптозащищенного туннеля.



***Рис. 4.12.*** *Схема взаимодействия по протоколу L2TP*

Сравнение практических аспектов рассмотренных протоколов туннелирования канального уровня модели OSI приведено в таблице 4.3.

*Таблица 4.3*

**Сравнение протоколов создания VPN-соединений на канальном уровне**

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр сравнения | РРТР | L2F | L2TP |
| Основан на протоколе РРР | + | + | + |
| В качестве сетевого протокола использует только IP | + | - | - |
| Необходимо специальное программное обеспечение | + | - | + |
| Используемые протоколы криптографического преобразования | Любой | IPSec | IPSec |
| Необходимо специальное оборудование | - | + | - |
| Присутствует управление потоками данных | - | - | + |
| Можно создать сразу несколько туннелей | - | - | + |

Из таблицы 4.3. следует, что протокол L2TP является самым удобным и наиболее многофункциональным в проекции практического использования. Однако и он не лишен недостатка: не предусмотрено создание криптозащищенного туннеля между точками информационного взаимодействия при использовании протокола IPv4 – это четвертая версия протокола IP (Internet Protocol), которая на сегодняшний является основной и обслуживает большую часть сети Интернет.

#### Протокол IPSec

Стандартные способы защиты информационного обмена на сетевом уровне модели OSI для IP-сети, являющейся основным видом публичных сетей, определяются протоколом *IPSec (Internet Protocol Security)*, который обеспечивает:

* аутентификацию источника данных;
* криптографическое закрытие передаваемых пакетов сообщений, проверку их целостности и подлинности после приема;  защиту от навязывания повторных сообщений;  частичную защиту от анализа трафика.

Установка соединения начинается с взаимной аутентификации сторон.

Далее происходит выбор параметров (будет ли осуществляться *аутентификация, шифрование, проверка целостности данных*) и необходимого протокола (AH (Authentication Header) или ESP (Encapsulating Security Payload) передачи данных.

После этого выбираются конкретные алгоритмы (например, шифрования, хэш-функции) из нескольких возможных схем, некоторые из которых определены стандартом (для шифрования – DES, для хэш-функций – MD5 или SHA1), другие добавляются производителями СКЗИ, использующими IPsec (например Triple DES, Blowfish, CAST).

Чтобы установить защищенное соединение, оба участника сеанса должны иметь возможность быстро согласовать параметры защиты, такие как алгоритмы аутентификации и ключи.

#### Протокол SSL

Протокол Secure Sockets Layer (SSL) является протоколом сеансового уровня модели OSI, использующим для обеспечения безопасности информационного обмена криптографические методы защиты информации.

Конфиденциальность передаваемых данных обеспечивается за счет их криптографического закрытия, а аутентификация взаимодействующих сторон, а также подлинность и целостность циркулирующей информации – за счет формирования и проверки цифровой подписи.

Ядром протокола SSL является технология комплексного использования асимметричных и симметричных криптосистем.

В качестве алгоритмов асимметричного шифрования используются такие алгоритмы, как RSA, DSA, ECDSA. Для вычисления хэш-функций могут применяться стандарты MD5 и SHA. Допустимыми алгоритмами симметричного шифрования являются RC2, RC4, IDEA, DES, Triple DES или AES, Camellia*.*

Для аутентификации взаимодействующих сторон и криптозащиты ключа симметричного шифрования применяются цифровые сертификаты открытых ключей пользователей (клиента и сервера), заверенные цифровой подписью специальных Сертификационных Центров.

Третья версия протокола SSL поддерживает три режима аутентификации:  взаимная аутентификация сторон;

 односторонняя аутентификация сервера без аутентификации клиента;  полная анонимность.

Данные, перемещающиеся между клиентом и сервером, шифруются симметричным алгоритмом, а для цифровых подписей и обмена ключами шифрования используется алгоритм с открытым ключом – асимметричные алгоритмы.

#### Протокол Kerberos

Протокол Kerberos был разработан специально для того, чтобы обеспечить надежную аутентификацию пользователей. Для этого протокола реализована аутентификация по многоразовым и по одноразовым паролям. Проблема передачи пароля решается криптографическими средствами. Вместо того, чтобы сообщать друг другу пароль, участники сеанса связи обмениваются криптографическим ключом, знание которого подтверждает личность собеседника. Для того чтобы такая технология была возможна, ключ должен обеспечивать и шифрование, и расшифровку сообщения, т.е. должны использоваться симметричные криптоалгоритмы.

По аналогии с древнегреческой

мифологией, этот протокол был

назван в честь трехголового пса, который защищал выход из царства Аида, – Цербера, или более точно – Кербера. Трем головам Цербера в протоколе соответствуют три участника безопасной связи: клиент, сервер и доверенный посредник между ними. Роль посредника здесь играет центр распределения ключей «Key distribution center», KDC.

Таким образом, в процессе аутентификации задействованы следующие основные компоненты:

* Клиент, запрашивающий доступ к службе или пытающийся осуществить аутентификацию.
* Сервер, на котором работают службы, доступ к которому требуется клиенту.
* Компьютер, которому доверяет клиент (в данном случае речь идет о контроллере домена, на котором выполняется служба KDC).
* KDC представляет собой службу, работающую на физически защищенном сервере.

KDC ведет базу учетных данных с информацией обо всех участниках своего домена. Вместе с информацией о каждом участнике в базе данных KDC сохраняется криптографический ключ, известный только этому объекту и службе KDC. Указанный ключ, который называют долговременным, используется для связи пользователя системы безопасности с центром распределения ключей. В большинстве практических реализаций протокола Kerberos долговременные ключи создаются на основе пароля пользователя.

Одним из преимуществ протокола Kerberos, обеспечивающих очень высокий уровень сетевой безопасности, является то, что во всех сетевых взаимодействиях в открытом виде не передаются ни пароли, ни значения функции хеширования паролей – все *шифруется.* На рис. 4.13 представлен один из этапов работы протокола – формирование ключа шифрования из пароля пользователя в результате хеширования.



***Рис. 4.13.*** *Схема формирования ключа шифрования из пароля пользователя в результате хеширования*

#### Протокол Radius

Протокол аутентификации RADIUS *(RemoteAuthenticationDial-inUserService)* рассматривается как механизм аутентификации и авторизации удаленных пользователей в условиях распределенной сетевой инфраструктуры (к примеру, телефонные линии, Internet), предоставляющий централизованные услуги по проверке подлинности и учету для служб удаленного доступа.

***Свойства RADIUS:***

* базируется на протоколе UDP (User Datagram Protocol – не использует прямых соединений);
* поддерживает аутентификацию по протоколам PAP и CHAP по PPP;
* использует хэширование MD5 для скрытия паролей;
* поддерживает модель AAA (от англ. *Authentication –* аутентификация*, Authorization –* авторизация, проверка полномочий, проверка уровня доступа*, Accounting –* учет, слежение за потреблением ресурсов пользователем, запись

фактов получения доступа к системе) – используется для описания процесса предоставления доступа и контроля над ним.

Протокол аутентификации RADIUSподдерживается большинством коммерческих устройств удаленного доступа.



***Рис. 4.14.*** *Аутентификация и авторизация через RADIUS-сервер*

Протокол RADIUS имеет недостаточный уровень безопасности в некоторых реализациях. В случае использования нескольких промежуточных серверов RADIUS все они имеют возможность просматривать передающиеся через них аутентификационные данные – сертификаты и пароли.

### 4.6. Персональные криптографические средства аутентификации

Персональными криптографическими средствами аутентификации являются *токены*. Токены – это компактные СКЗИ (от англ. *Token* – знак, символ; опознавательный знак; жетон), контактные смарт-карты с USB-интерфейсом, аппаратные ключи, предназначенные для безопасного хранения цифровых сертификатов, средства для многофакторной аутентификации пользователей, электронной подписи и безопасного хранения криптографических ключей.

***Рис. 4.15.*** *Смарт-карта и USB-ключ eToken PRO, eToken NG-FLASH, eToken NG-OTP, eToken PRO (Java), eToken PASS (а) и рутокены компании Актив в различных форм-факторах (б)*



а

б

Основу токена составляет *микроконтроллер*, который выполняет криптографическое преобразование данных, и *память,* в которой хранятся данные пользователя (пароли, сертификаты, ключи шифрования и т.д.).

321

eToken (от англ. *electronic* – электронный и от англ. *token* – признак, жетон) – торговая марка для линейки персональных средств аутентификации в виде USB-ключей и смарткарт, а также программные решения с их использованием. Торговая марка была создана израильской компанией Aladdin Knowledge Systems, впоследствии приобретенной SafeNet.

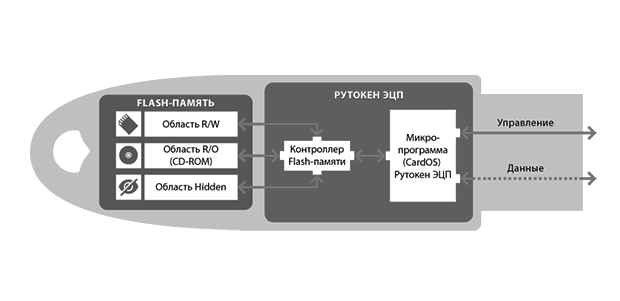
eToken PRO представляет собой защищенное устройство для строгой

аутентификации, безопасного хранения секретных данных, выполнения криптографических вычислений и работы с асимметричными ключами и цифровыми сертификатами. USB-ключ архитектурно реализован как USB-картридер с встроенной в него микросхемой (чипом) смарт-карты. Ключ выполнен в виде брелка и напрямую подключается к USB- порту компьютера, при этом не требует для своей работы каких-либо дополнительных устройств.

Рутокен (от англ. *russian* – русский и англ. *token* – признак, жетон) – программные и аппаратные средства для многофакторной аутентификации пользователей, электронной подписи и безопасного хранения криптографических ключей. Разрабатываются и выпускаются российской компанией «Актив» https://ru.wikipedia.org/wiki/%D0%A0%D1%83%D1%82%D0%BE%D0%BA%D0 %B5%D0%BD – cite\_note-1. Особенностью данных устройств является возможность аппаратного осуществления криптографических операций (в том числе по стандартам ГОСТ) и надежное хранение ключевой информации внутри микроконтроллера.

Устройство Рутокен ЭЦП Flash представляет собой криптографический USB-токен Рутокен ЭЦП, дополненный управляемой Flash-памятью объемом от 4 до 64 ГБ. Flash-память устройства можно использовать для хранения дистрибутива программного обеспечения, автоматического запуска приложений при подключении токена, доверенной загрузки операционной системы. На базе данного устройства возможно создание специализированных программно-аппаратных решений. Имеет сертификат ФСБ о соответствии требованиям, предъявляемым к СКЗИ по классу КС2 и к средствам электронной подписи в соответствии с № 63-ФЗ «Об электронной подписи».

Важной особенностью электронного идентификатора Рутокен ЭЦП Flash является наличие управляемой Flash-памяти. Она может быть разделена на разделы, доступ к которым разграничивается с помощью PIN-кодов. Предусмотрена возможность создания скрытых и CD-ROM разделов. Доступ и управление доступом к Flash-памяти осуществляется непосредственно через микропроцессор токена без использования каких-либо дополнительных электронных компонентов (хабов, контроллеров и т.п.). Для каждого раздела Flash-диска определяются индивидуальные права доступа на чтение и запись, которые могут быть изменены «на лету» прямо во время работы устройства.



***Рис. 4.16.*** *Рутокен ЭЦП Flash*

Подобные аппаратные СКЗИ широко представлены на рынке, и каждый может выбрать наиболее оптимальное для своих целей устройство. Особенность ключей состоит в том, что один и тот же ключ может позволять производить авторизацию для доступа по Wi-Fi в сеть, для установления VPN соединения и для шифрования данных на жестком диске.

Токены Рутокен получили свое название, поскольку позиционируются как отечественные решения – практически все их комплектующие российского производства, и сборка устройств производится на территории России.

На продукты Рутокен имеются сертификаты ФСБ и ФСТЭК России, поэтому устройства Рутокен являются основными ключевыми носителями в массовых российских проектах, базирующихся на технологиях электронной подписи (ЭП) и инфраструктуре открытых ключей (PKI). Продукты на базе Рутокен широко используются в коммерческих и государственных проектах.

Основные характеристики *рутокена:*

* аппаратное шифрование по ГОСТ 28147-89;
* файловая система по ISO 7816;
* 8, 16 или 32 Кбайт энергонезависимой памяти.

Общие технические характеристики:

* 2-факторная аутентификация (по факту наличия рутокена и по факту предъявления PIN-кода);
* 32-битовый уникальный серийный номер;
* поддержка стандартов ISO/IEC 7816, PC/SC, ГОСТ 28147-89, MS

CryptoAPI и MS SmartcardAPI, PKCS#11 (v.2.10+);

* собственные CryptoServiceProvider и ICC ServiceProvider со стандартными наборами интерфейсов и функций API;
* поддержка стандарта X.509 и алгоритмов RSA, DES (3DES), RC2, RC4, MD4, MD5, SHA-1;
* защищенное хранение ключей асимметричного шифрования и цифровых сертификатов;
* возможность использовать рутокен для асимметричного шифрования данных и работы с цифровыми сертификатами из любых smartcard-приложений стандарта PC/SC.

PKCS относится к группе *Public Key Cryptography Standarts* (Стандарты криптографии с открытым ключом), разработанным и опубликованным RSA

Laboratories.

Ключевые носители eToken предоставляют для криптопровайдеров надежный отчуждаемый носитель для защищенного хранения ключевой информации.

*Недостаток:* PIN-код вводится не с собственной клавиатуры устройства, а с клавиатуры терминала, к которому устройство подключено, таким образом, с помощью троянской программы злоумышленник может перехватить PIN-код и произвести неоднократное несанкционированное подписывание или шифрование любой информации от имени владельца устройства.

*Таблица 4.4* **Сравнение eToken и Рутокен**

|  |  |  |
| --- | --- | --- |
|  | eToken Pro | Рутокен S |
| Производитель | SafeNet (Аладдин) | Актив |
| Поддержка операционных систем | MS Windows | MS Windows |
|  | 2000/XP/2003/Vista/2008/7  /2012/8 (32/64-бит), Mac  OS X, GNU/Linux | 2000/XP/2003/Vista/2008/7/201  2/8 (32/64-бит), Mac OS X,  GNU/Linux |
| Объем защищенной памяти | 72 КБ (из них 47 КБ доступно пользователю) | 32, 64, 128 КБ |
| Количество контейнеров КриптоПро CSP | до 10 | от 7 до 31 |
| Сертификация | НДВ4 | НДВ3, НДВ4, ФСБ |
| Радиометки (RFID) | встраивание возможно | встраивание возможно |
| Форм-факторы устройства |  USB-ключ  смарт-карты |  USB-ключи  micro-токены |
| Брендирование | заказной цвет устройства,  тампопечать | заказной цвет устройства, гра-  вировка, тампопечать, брендированная упаковка |

НДВ – недекларированные возможности – функциональные возможности средств вычислительной техники, не описанные или не соответствующие описанным в документации, при использовании которых возможно нарушение конфиденциальности, доступности или целостности обрабатываемой информации. Сертификацию по НДВ проводит ФСТЭК.

#### Вопросы и задания для самостоятельной работы

1. Какая технология лежит в основе СКЗИ сетевого взаимодействия?
2. В чем состоит принцип туннелирования в компьютерных сетях?
3. Чем отличается криптографическая защита на прикладном уровне от криптозащиты на сетевом уровне? Какая защита надежней?
4. На каких уровнях модели сетевого взаимодействия строится криптографическая защита удаленного доступа к локальной сети?
5. Чем отличается криптографическая защита на базе криптошлюзов от защиты по протоколу IPSek?
6. Какие преимущества имеют сертифицированные криптошлюзы при защите удаленного доступа?
7. Приведите пример сертифицированных криптошлюзов для защиты удаленного доступа.
8. Какие алгоритмы шифрования поддерживает протокол IPSek?
9. Как обеспечивается аутентификация источника передаваемых IPпакетов по протоколу IPSek?
10. Какие ограничения по применению для защиты удаленного доступа имеют СКЗИ компании CiscoSystems?
11. Приведите пример сертифицированных СКЗИ для передачи данных в локальных сетях.
12. Каким образом в СКЗИ VPN может быть реализована защита от DoSатак в процессе сетевого взаимодействия? Приведите пример СКЗИ с защитой от DoS-атак.
13. Какой функциональный блок является ядром СКЗИ VPN? Приведите пример.
14. Как происходит авторизация криптомаршрутизаторов в VPN-сети?
15. От чего зависит функциональность реализуемой VPN-сети?
16. Назовите криптографические протоколы канального уровня. В чем их сходство и отличие?
17. Какие возможности дает построение виртуальных сетей на сеансовом уровне? С какими ограничениями связана криптозащита на этом уровне?
18. На каком уровне модели OSI можно достичь оптимального соотношения между прозрачностью и качеством защиты?
19. Какие функции защиты обеспечивает протокол IPSek?
20. Какие режимы аутентификации поддерживает протокол SSL, и какие алгоритмы для этого используются?
21. Какие особенности протокола аутентификации Kerberos обеспечивают высокий уровень сетевой безопасности?
22. В чем преимущества и недостатки СКЗИ-токенов?

# Приложения

*Приложение 1*

**Пример зашифрования текста различными шифрами.**

В рассматриваемых способах шифрования в качестве примера используется следующая пословица:

## Выпущенное слово и камень не имеют возврата

### *Шифр простой замены АТБАШ*

Одним из первых шифров был ***шифр простой замены*** «**АТБАШ**». Алгоритм шифрования представлял следующее: каждой букве алфавита соответствовала буква, цифра или комбинация.

**Yi = X(n-i +1)**

Где:

Y – исходный (открытый) текст

X(n-i +1) – зашифрованный текст

i – порядковый номер буквы открытого текста, i=(1…n) n – количество букв в выбранном алфавите.

То есть букве Б соответствовала буква Я при n=33 и i=2

Таким образом, рассматриваемая пословица преобразуется в зашифрованный текст с помощью следующей таблицы:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Y** | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р |
| **X** | - | Я | Ю | Э | Ь | Ы | Ъ | Щ | Ш | Ч | Ц | Х | Ф | У | Т | С | Р |
| **i** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Y** | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | - |
| **X** | П | О | Н | М | Л | К | Й | И | З | Ж | Е | Д | Г | В | Б | А |
| **i** | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 |

Полученный зашифрованный текст выгладит следующим образом:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Ю** | **Е** | **С** | **Н** | **З** | **Ы** | **У** | **У** | **Т** | **Ы** | **А** | **П** | **Х** | **Т** | **Ю** | **Т** | **А** | **Ш** | **А** | **Ц** | **-** | **Ф** | **Ы** | **У** | **Д** | **А** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **У** | **Ы** | **А** | **Ш** | **Ф** | **Ы** | **В** | **О** | **А** | **Ю** | **Т** | **Щ** | **Ю** | **Р** | **-** | **О** | **-** |

### *Шифр циклических подстановок Цезаря*

Криптология – древняя наука и обычно это подчеркивают рассказом о Юлии Цезаре (100–44 гг. до н.э.), переписка которого с Цицероном (106–43 гг. до н.э.) и другими «абонентами» в Древнем Риме шифровалась. **Шифр Цезаря**, иначе шифр циклических подстановок, состоит в замене каждой буквы в сообщении буквой алфавита, отстоящей от нее на фиксированное число букв. Алфавит считается циклическим, то есть после Z следует A. Цезарь заменял букву буквой, отстоящей от исходной на три.

#### Yi = Xi +3 mod n

Где:

Y – исходный (открытый) текст Xi +1 – зашифрованный текст

i – порядковый номер буквы открытого текста, i=(1…n) n – количество букв в выбранном алфавите. То есть букве Б соответствовала буква Д при i=2

Таким образом, рассматриваемая пословица преобразуется в зашифрованный текст с помощью следующей таблицы:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Y** | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р |
| **X** | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У |
| **i** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | - |
| Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | - | А | Б | В |
| 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 |

Полученная в результате шифрограмма выглядит следующим образом:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Е | Ю | Т | Ц | Ь | И | Р | Р | С | И | В | Ф | О |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| В | Р | И | В | Л | П | И | А | Х | В | Е | С | К | Е | У | Г | Х | Г |

### *Шифр маршрутной перестановки с числовым ключом*

В **шифре маршрутной перестановки** буквы открытого текста не замещаются на другие, а меняется сам порядок их следования. Таким образом ключом данного шифра является маршрут – определенное число или количество букв в ключевом слове. В шифре маршрутной перестановки исходный (открытый) текст записывается построчно в таблицу, запись осуществляется слева направо или справа налево, или и тем и другим способом поочередно. При этом число букв исходного текста, размещаемых в строке таблицы, фиксировано, то есть равно определенному выбранному числу или количеству букв в выбранном ключевом слове.

Например, ключом является число 8, тогда число букв в строке таблицы равно 8, рассматриваемая пословица записывается в таблицу слева направо, потом справа налево и т.д., в оставшиеся пустые клетки таблицы записываются любые буквы. Таблица выглядит следующим образом:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| В | Ы | П | У | Щ | Е | Н | Н |
| И | О | В | О | Л | С | Е | О |
| К | А | М | Е | Н | Ь | Н | Е |
| З | О | В | Т | Ю | Е | М | И |
| В | Р | А | Т | А | Й | Ц | У |

Далее из полученной таблицы выписываются столбцы – буквы первого столбца записываются в очередности сверху вниз, буквы второго столбца – в очередности снизу вверх, и так далее. При этом можно выписывать по 4 буквы, по 5 букв, по 6 букв и т.д. В данном случае выпишем буквы по их количеству в столбце сверху вниз, зашифрованный текст будет иметь следующий вид:

**В И К З В Ы О А О Р П В М В А У О Е Т Т**

**Щ Л Н Ю А Е С Ь Е Й Н Е Н М Ц Н О Е И У**

***Шифр маршрутной перестановки с ключевым словом***

Если ключом является слово, то буквам этого слова присваиваются номера их очередности в алфавите и столбцы таблицы выписываются в полученной очередности. Например, пусть ключевым словом является «ВЛАДИМИР», очередность букв получается следующая:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **В** | **Л** | **А** | **Д** | **И** | **М** | **И** | **Р** |
| 2 | 6 | 1 | 3 | 4 | 7 | 5 | 8 |

Таблица с исходным текстом имеет вид:

**В Л А Д И М И Р**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| В | Ы | П | У | Щ | Е | Н | Н |
| И | О | В | О | Л | С | Е | О |
| К | А | М | Е | Н | Ь | Н | Е |
| З | О | В | Т | Ю | Е | М | И |
| В | Р | А | Т | А | Й | Ц | У |

Выпишем буквы аналогично предыдущему примеру, и шифрограмма пословицы будет выглядеть так:

**П В М В А В И К З В У О Е Т Т Щ Л Н Ю А**

**Н Е Н М Ц Ы О А О Р Е С Ь Е Й Н О Е И У**

### *Поворотная решетка Кардано*

Интересным шифром маршрутной перестановки является **решетка Кардано**. Выбирается квадрат, в котором количество клеток равно количеству букв исходного текста. Если в исходном тексте нечетное количество букв, то последняя буква отсекается. В квадрате определенным образом заштриховывается (отмечается) некоторое количество клеток, получаемое следующим образом: общее количество клеток квадрата делится на 4, так как решетка Кардано предусматривает только 4 перемещения – поворота. Ключом для данного шифра является расположение вырезанных клеток.

Для исходного текста пословицы решетка Кардано представляет собой квадрат:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

В данный квадрат в отмеченные клетки вписывается исходный текст:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **В** |  |  |  | **Ы** |  |
|  | **П** |  |  |  |  |
|  |  | **У** |  | **Щ** |  |
|  |  |  |  |  | **Е** |
| **Н** |  | **Н** |  |  |  |
|  |  |  | **О** |  |  |

Далее решетка Кардано поворачивается относительно своей вертикальной оси, и продолжаем записывать исходный текст в отмеченные клетки:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  | В | Е |  |  | Ы | С |
|  |  |  |  |  |  |  | П |  |  | Л |  |
|  |  |  |  |  |  |  | О | У | В | Щ |  |
|  |  |  |  |  |  | О |  |  |  |  | Е |
|  |  |  |  |  |  | Н |  | Н | И |  | К |
|  |  |  |  |  |  |  |  | А | О |  |  |

Во второй раз решетка поворачивается относительно горизонтальной оси:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  | В | Е | М |  | Ы | С |
|  |  |  |  |  |  |  | П |  | Е | Л | Н |
|  |  |  |  |  |  | Ь | О | У | В | Щ |  |
|  |  |  |  |  |  | О | Н |  | Е |  | Е |
|  |  |  |  |  |  | Н |  | Н | И | И | К |
|  |  |  |  |  |  |  | М | А | О |  | Е |

В третий раз решетка снова поворачивается относительно вертикальной оси, текст записывается в оставшиеся клетки и шифрограмма принимает окончательный вид:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  | В | Е | М | Ю | Ы | С |
|  |  |  |  |  |  | Т | П | В | Е | Л | Н |
|  |  |  |  |  |  | Ь | О | У | В | Щ | О |
|  |  |  |  |  |  | О | Н | З | Е | В | Е |
|  |  |  |  |  |  | Н | Р | Н | И | И | К |
|  |  |  |  |  |  | А | М | А | О | Т | Е |

### *Шифр биграммной замены Квадрат Полибия*

Шифр замены, называемый **Квадратом Полибия** был изобретен во 2 веке до нашей эры, он использовался для предупреждения об опасности. Текст алфавита помещается в таблицу, где каждой букве соответствуют два числа – номер строки i и номер столбца j.

**1 2 3 4 5 6**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **А** | **Б** | **В** | **Г** | **Д** | **Е** |
| **Ж** | **З** | **И** | **Й** | **К** | **Л** |
| **М** | **Н** | **О** | **П** | **Р** | **С** |
| **Т** | **У** | **Ф** | **Х** | **Ц** | **Ч** |
| **Ш** | **Щ** | **Ъ** | **Ы** | **Ь** | **Э** |
| **Ю** | **Я** | **-** |  |  |  |

**1**

**2**

**3**

**4**

**5**

**6**

Шифрование производится по формуле:

#### Yij = ij

Где:

Y – исходный (открытый) текст ij – зашифрованный текст

Например, букве «И» соответствует число 23, где 2 – номер строки, 3 – номер столбца.

Таким образом, рассматриваемая половица в зашифрованном виде представляет собой:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1**  **3** | **5**  **4** | **3**  **4** | **4**  **2** | **5**  **2** | **1**  **6** | **3**  **2** | **3**  **2** | **3**  **3** | **1**  **6** | **6**  **3** | **3**  **6** | **2**  **6** | **3**  **3** | **1**  **3** | **3**  **3** | **6**  **3** | **2**  **3** | **6**  **3** |

**25 11 31 16 32 55 63 32 16 63 23 31 16 61 41 63**

**13 33 22 13 35 11 41 11**

### *Многоалфавитный шифр с ключевым словом*

Шифрование по **Многоалфавитному шифру** с ключевым словом (паролем) представляет собой квадрат, в котором количество строк и столбцов соответствуют количеству букв в алфавите. Таким образом, каждая строка (начиная со второй) имеет сдвиг влево на одну букву, а вытесненные буквы переходят в конец строки.

Далее выбирается слово пароль, например, «Ноябрь». В соответствии с алфавитом, записанным в левом столбце и соответствующим количеству строк таблицы, на основе букв ключевого слова, определяются строки, которые будут использоваться для шифрования.



По первой вспомогательной строке таблицы с алфавитом выбирается первая буква исходного текста, в нашем случае, это буква «В», далее по первой букве ключевого слова выбирается строка, в которой букве «В» соответствует определенная буква зашифрованного текста – буква «П». Затем по горизонтали (первая вспомогательная строка таблицы) выбирается вторая буква исходного текста «Ы» и по вертикали (вспомогательный левый столбец) определяется строка по второй букве ключевого слова, в данной строке букве «Ы» соответствует буква «Й» зашифрованного текста и т.д.

Шифрограмма пословицы «Выпущенное слово и камень не имеют возврата» имеет вид:



### *Шифр гаммирования*

**Шифр «Гаммирование»** был изобретен в XIX веке. Он представляет собой запись текста в цифровом виде, которая получается в следующем порядке. Выбирается некоторое число гамма, например 5129. Алфавит представляется в виде двухзначных цифр.

А Б В Г Д Е Ж З И Й К Л М Н О П Р

01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17

С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я -

18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33

Исходный текст записывается в виде цифр по 2 буквы и к нему прибавляется число-гамма. Например:

П – 16

И – 09

1609 +

5129

6738

Шифрограмма рассматриваемой пословицы представляет собой следующие комбинации чисел:

**5457 6749 7735 6543 6635 8447 6344 5444**

**8438 8440 5242 5743 8062 6535 8438 6435**

**8248 8432 6637 5446 5248 5130**

### *Криптосистема без передачи ключа*

**Симметричное шифрование,** то есть шифрование с секретным ключом, основано на том, что отправитель и получатель информации используют один и тот же ключ. Этот ключ должен храниться в тайне и передаваться способом, исключающим его перехват.

Обмен информацией осуществляется в 3 этапа:

* отправитель передает получателю ключ (в случае сети с несколькими абонентами у каждой пары абонентов должен быть свой ключ, отличный от ключей других пар);
* отправитель, используя ключ, зашифровывает сообщение, которое пересылается получателю;
* получатель получает сообщение и расшифровывает его.

Если для каждого дня и для каждого сеанса связи будет использоваться уникальный ключ, это повысит защищенность системы.

Рассмотрим алгоритм симметричного шифрования.

Два абонента условились организовать переписку, для этого они выбирают большое простое число Р – такое, что Р-1 легко раскладывалось на не очень большие простые числа-сомножители. Для примера возьмем небольшое число Р=33.

Далее решаем функцию Эйлера, так как число 33 не является простым и раскладывается на множители 3 и 11. То есть мы получаем Р1 = 3, Р2 = 11. Функция Эйлера:

Y (Р1, Р2) = Y (Р1) х Y (P2) = (3-1) х (11-1) = 20

Таким образом получаем модуль для сравнения первой степени, и при выбранном ключе 3 (взаимно-простое число с полученным модулем 20) имеем следующее сравнение:

3 \* Х = 1 (mod 20)

Для решения данного сравнения вычисляется НОД (Наибольший общий делитель)

20/3=6 остаток 2

3/2=1 остаток 1

2/1=2 остаток 0

Результаты заносятся в таблицу

M 1 2 3 qm 6 1 2

Pm 1 6 7 20

Где m – число делителей, qm – делители, Pm неполное частное деление

Рm = gm \* Pm-1 + Pm-2

Заполнив таблицу согласно данной формуле, можем вычислить Х:

Х = Рm-1 = 7

Решение сравнения примет вид:

3 \* 7 = 1 (mod 20)

Зашифруем текст рассматриваемой пословицы выбранным ключом 3 по формуле:

Y1=YA(mod P)

Где:

Y1 – зашифрованный текст

Y – исходный текст

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **А** | **Б** | **В** | **Г** | **Д** | **Е** | **Ж** | **З** | **И** | **Й** | **К** | **Л** | **М** | **Н** | **О** | **П** | **Р** | **С** | **Т** | **У** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **1** | **8** | **27** | **31** | **26** | **18** | **13** | **17** | **3** | **10** | **11** | **12** | **19** | **5** | **9** | **4** | **29** | **24** | **28** | **14** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Ф** | **Х** | **Ц** | **Ч** | **Ш** | **Щ** | **Ъ** | **Ы** | **Ь** | **Э** | **Ю** | **Я** | **-** |
| **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** | **31** | **32** | **33** |
| **21** | **22** | **23** | **30** | **16** | **20** | **15** | **7** | **2** | **6** | **25** | **32** | **0** |

В результате получается шифрограмма:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **27 7** | **4** | **14 20 18 5** | **5** | **9 18 0** | **24 12 9 27 9 0** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **10 0** | **11 1 19 18 5** | **2 0** | **5** | **18 0** | **10 19 18 25 28 0** |

**27 9 17 27 29 1 28 1**

*Приложение 2*

**Надежность криптосистем**

#### Ненадежность криптосистем

В современном программном обеспечении *(ПО)* криптоалгоритмы широко применяются не только для задач шифрования данных, но и для аутентификации и проверки целостности. На сегодняшний день существуют хорошо известные и апробированные криптоалгоритмы (как с симметричными, так и несимметричными ключами), криптостойкость которых либо доказана математически, либо основана на необходимости решения математически сложной задачи (факторизации, дискретного логарифмирования и т.п.). К наиболее известным из них относятся *DES, ГОСТ, RSA.* Таким образом, они не могут быть вскрыты иначе, чем полным перебором или решением указанной задачи.

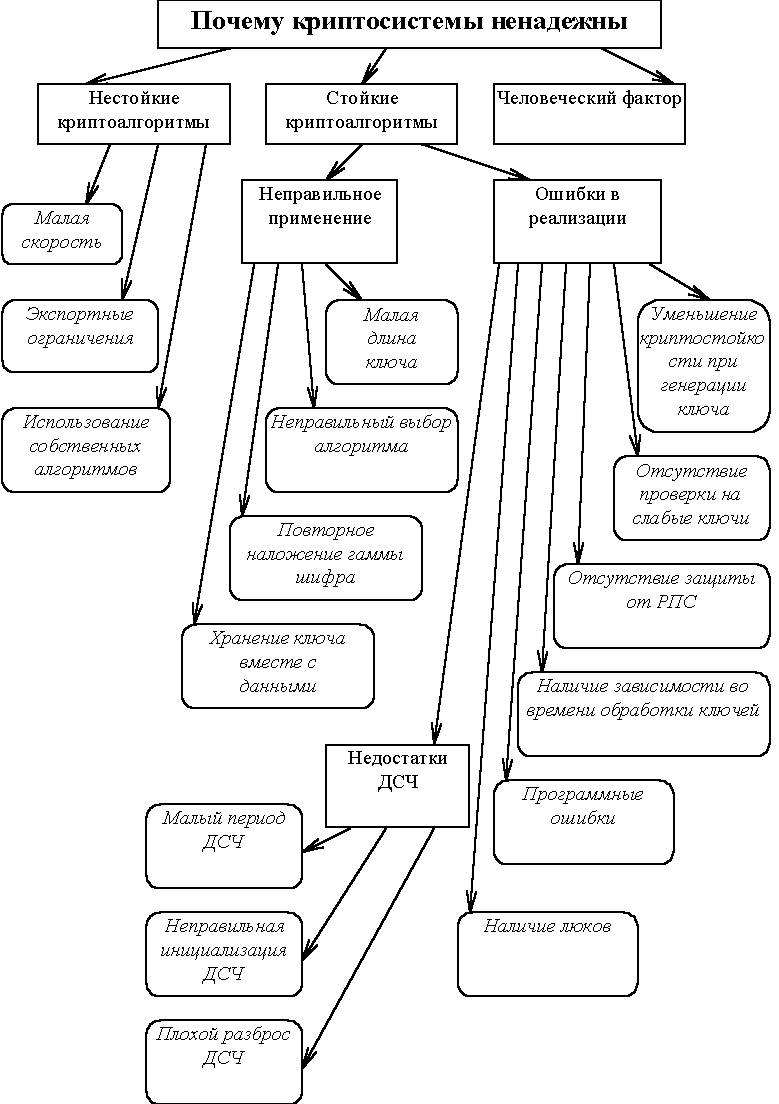
С другой стороны, в компьютерном и околокомпьютерном мире все время появляется информация об ошибках или «дырах» в той или иной программе (в т.ч. применяющей криптоалгоритмы), или о том, что она была взломана (*cracked*). Это создает недоверие как к конкретным программам, так и к возможности вообще защитить что-либо криптографичеcкими методами не только от спецслужб, но и от простых хакеров.

Поэтому знание истории атак и «дыр» в криптосистемах, а также понимание причин, по которым они имели место, является одним из необходимых условий разработки защищенных систем. Перспективным направлением исследований в этой области является анализ успешно проведенных атак или выявленных уязвимостей в криптосистемах с целью их обобщения, классификации и выявления причин и закономерностей их появления и существования. Это и будет являться задачей данной статьи.

По аналогии с таксономией причин нарушения безопасности ВС, выделим следующие причины ненадежности криптографических программ (см. рис. 1):

1. Невозможность применения стойких криптоалгоритмов.
2. Ошибки в реализации криптоалгоритмов.
3. Неправильное применение криптоалгоритмов.
4. Человеческий фактор.

Отметим сразу, что рассматриваемые ниже причины покрывают только два вида потенциально возможных угроз: *раскрытия* и *целостности*, оставляя в стороне угрозу *отказа в обслуживании*, которая приобретает все большее значение по мере развития распределенных криптосистем.



***Рис. 1.*** *Причины ненадежности криптосистем*

.

#### Малая скорость стойких криптоалгоритмов

Это основной фактор, затрудняющий применение хороших алгоритмов в, например, системах «тотального» шифрования или шифрования «на лету». В частности, программаNorton DiskReet, хотя и имеет реализацию *DES*, при смене пользователем ключа может не перешифровывать весь диск, т.к. это займет слишком много времени. Аналогично, программа компрессии «на лету» Stackerфирмы *Stac Electronics* имеет опцию закрытия паролем компрессируемых данных. Однако она не имеет физической возможности зашифровать этим паролем свой файл, обычно имеющий размеры в несколько сот мегабайт, поэтому она ограничивается очень слабым алгоритмом и хранит хэш-функцию от пароля вместе с защищаемыми данными. Величина криптостойкости этой функции была исследована и оказалась равной 28, т.е. пароль может быть вскрыт тривиально.

#### Экспортные ограничения

Это причина, связанная с экспортом криптоалгоритмов или с необходимостью приобретать патент или права на них. В частности, из США запрещен экспорт криптоалгоритмов с длиной ключа более 40 бит. Очевидно, что такая криптостойкость не может считаться надежной при современных вычислительных мощностях, и даже на персональном компьютере, положив скорость перебора в 50 000 паролей/сек, получим время перебора в среднем порядка 4 месяцев.

Известные примеры программ, подверженных экспортным ограничениям,

– это последние версии *броузеров (browser)* Интернета, в частности Netscape Navigator фирмы *Netscape Communications* и Internet Explorer фирмы *Microsoft*. Они предоставляют шифрование со 128-битным ключом для пользователей внутри США и с 40-битным ключом для всех остальных.

Также в эту группу попадает последняя версия архиватора ARJ 2.60, известного своим слабым алгоритмом шифрования архивов. Теперь пользователи внутри США могут использовать криптостойкий алгоритм ГОСТ. Комизм ситуации в том, что хотя этот алгоритм является российским, даже россияне по законам США все равно не могут воспользоваться им в программе ARJ.

#### Неправильная реализация криптоалгоритмов

Несмотря на то, что в этом случае применяются криптостойкие или сертифицированные алгоритмы, эта группа причин приводит к нарушениям безопасности криптосистем из-за их неправильной реализации.

.

#### Отсутствие проверки на слабые ключи

Некоторые криптоалгоритмы (в частности, DES, IDEA) при шифровании со специфическими ключами не могут обеспечить должный уровень криптостойкости. Такие ключи называют слабыми *(weak).* Для DES известно 4 слабых и 12 полуслабых *(semi-weak)* ключей. И хотя вероятность попасть в них равняется ~2х10-16, для серьезных криптографических систем пренебрегать ею нельзя.

Мощность множества слабых ключей IDEA составляет ни много ни мало – 251 (впрочем, из-за того, что всего ключей 2128, вероятность попасть в него в 3Ї107 раз меньше, чем у DES).

#### Недостаточная защищенность от РПС

РПС (разрушающие программные средства) – это компьютерные вирусы, троянских кони, программные закладки и т.п. программы, способные перехватить секретный ключ или сами нешифрованные данные, а также просто подменить алгоритм на некриптостойкий. В случае, если программист не предусмотрел достаточных способов защиты от РПС, они легко способны нарушить безопасность криптосистемы. Особенно это актуально для операционных систем, не имеющих встроенных средств защиты или средств разграничения доступа – типа MS DOS или Windows 95:

1. *Перехват пароля.* Как пример можно привести самый старый способ похищения пароля, известный еще со времен больших ЭВМ, когда программа«фантом» эмулирует приглашение ОС, предлагая ввести имя пользователя и пароль, запоминает его в некотором файле и прекращает работу с сообщением *«Invalid password»*. Для MS DOS и Windows существует множество закладок для чтения и сохранения паролей, набираемых на клавиатуре (через перехват соответствующего прерывания), например, при работе утилиты Diskreet v.

6.0.

1. *Подмена криптоалгоритма.* Примером реализации этого случая является закладка, маскируемая под прикладную программу-«ускоритель» типа *Turbo Krypton*. Эта закладка заменяет алгоритм шифрования ГОСТ 28147-89, реализуемой платой *«Krypton-3»* (демонстрационный вариант), другим, простым и легко дешифруемым алгоритмом.
2. *Троянский конь в электронной почте.* Последним примером служит имевшие место в июне 1998 года попытки проникновения троянского коня через электронную почту. В письмо были вложены порнографическая картинка и EXE-файл FREECD.EXE, который за то время, пока пользователь развлекался с письмом, расшифровывал пароли на соединение с провайдером (Dial-Up) и отправлял их на адрес ispp@usa.net.

#### Малая длина ключа

Это самая очевидная причина. Возникает вопрос: как стойкие криптоалгоритмы могут иметь малую длину ключа? Видимо, вследствие двух факторов.

1. Некоторые алгоритмы могут работать с переменной длиной ключа, обеспечивая разную криптостойкость, и именно задача разработчика выбрать необходимую длину, исходя из желаемой криптостойкости и эффективности.

Иногда на это желание накладываются и иные обстоятельства – такие, как экспортные ограничения.

1. Некоторые алгоритмы разрабатывались весьма давно, когда длина используемого в них ключа считалась более чем достаточной для соблюдения нужного уровня защиты.

С резким скачком производительности вычислительной техники сначала столкнулся алгоритм RSA, для вскрытия которого необходимо решать задачу факторизации. В марте 1994 года была закончена длившаяся в течение 8 месяцев факторизация числа из 129 цифр (428 бит). Для этого было задействовано 600 добровольцев и 1600 машин, связанных посредством электронной почты. Затраченное машинное время было эквивалентно примерно 5000 MIPS-лет.

Прогресс в решении проблемы факторизации во многом связан не только с ростом вычислительных мощностей, но и с появлением в последнее время новых эффективных алгоритмов. (На факторизацию следующего числа из 130 цифр ушло всего 500 MIPS-лет.) На сегодняшний день в принципе реально факторизовать 512-битные числа. Если вспомнить, что такие числа еще недавно использовались в программе PGP, то можно утверждать, что это самая быстро развивающаяся область криптографии и теории чисел.

29 января 1997 года фирмой RSA Labs был объявлен конкурс на вскрытие симметричного алгоритма RC5. 40-битный ключ был вскрыт через 3,5 часа после начала конкурса! (Для этого даже не потребовалась связывать компьютеры через Интернет – хватило локальной сети из 250 машин в Берклевском университете.) Через 313 часов был вскрыт и 48-битный ключ. Таким образом, всем стало очевидно, что длина ключа, удовлетворяющая экспортным ограничениям, не может обеспечить даже минимальной надежности.

Параллельно со вскрытием RC5 был дан вызов и столпу американской криптографии – алгоритму DES, имеющему ключ в 56 бит. И он пал 17 июня 1997 года, через 140 дней после начала конкурса (при этом было протестировано около 25% всех возможных ключей и затрачено примерно 450 MIPS-лет). Это было безусловно выдающееся достижение, которое означало фактическую смерть DES как стандарта шифрования. И действительно, когда в начала 1998 года следующее соревнование по нахождению ключа DES привело к успеху всего за 39 дней, национальный институт стандартов США (NIST) объявил конкурс на утверждение нового стандарта AES (Advanced Encryption Standard).

AES должен быть полностью открытым симметричным алгоритмом с ключом размером 128, 192, 256 бит и блоком шифрования размером 128 бит.

#### Повторное наложение гаммы шифра

Уже классическим примером стала уязвимость в Windows 3.x и первых версиях Windows 95, связанная с шифрованием. В этом случае программисты фирмы Microsoft, хорошо известные своими знаниями в области безопасности, применяли алгоритм RC4 (представляющем собой не что иное, как шифрование гаммированием), не меняя гаммы, несколько раз к разным данным – сетевым ресурсам, хранящимся в файлах типа .pwl.

Оказалось, что один из наборов данных файла .pwl представлял из себя более чем специфичный текст – 20-символьное имя пользователя (в верхнем регистре) и набор указателей на ресурсы (см. рис. 2). Таким образом, угадав имя пользователя (которое в большинстве случаев к тому же совпадает с именем файла), можно вычислить по крайней мере 20 байт гаммы. Т.к. гамма не меняется при шифровании других ресурсов (в этом состоит основная ошибка применения RC4 в этом случае), могут быть вычислены первые 20 байт всех ресурсов, в которые входит длина каждого из них. Вычислив длину, можно найти значения указателей и тем самым прибавить еще несколько десятков байт к угаданной гамме. Этот алгоритм реализован в известной программе glide.



***Рис. 2.*** *Формат файла .PWL*

#### Хранение ключа вместе с данными

Эта причина приводит к тому, что данные, зашифрованные с помощью криптостойкого и корректно реализованного алгоритма, могут быть легко дешифрованы. Это связано со спецификой решаемой задачи, при которой невозможно вводить ключ извне и он хранится где-то внутри в практически незашифрованном виде. Иначе говоря, здесь наиболее уязвимым будет алгоритм шифрования не ключом, а ключа (с помощью некоего вторичного ключа). Но так как (что опять-таки очевидно следует из специфики задачи) этот вторичный ключ хранить извне нельзя, то основные данные рано или поздно будут расшифрованы без использования методов перебора.

Типичным примером здесь будут все WWW-, ftp-, e-mail-клиенты. Дело в том, что для базовой (наиболее часто встречающейся) аутентификации в этих протоколах пароль должен передаваться серверу в открытом виде. Поэтому клиентские программы вынуждены шифровать (а не хэшировать) пароль, причем с фиксированным ключом, чтобы не надоедать пользователю постоянными вопросами. Отсюда следует, что где-то внутри любого броузера, почтового или ftp-клиента (будь то Netscape Communicator, Eudora, Outlook, FAR и т.п.) лежат все ваши пароли в практически открытом виде, и что расшифровать их не представляет труда. (Чаще всего, кстати, пароль в таких программах даже не шифруется, а кодируется алгоритмом типа base-64.)

#### Человеческий фактор

В любой критической системе ошибки человека-оператора являются чуть ли не самыми дорогостоящими и распространенными. В случае криптосистем непрофессиональные действия пользователя сводят на нет самый стойкий криптоалгоритм и самую корректную его реализацию и применение.

В первую очередь это связано с выбором паролей. Очевидно, что короткие или осмысленные пароли легко запоминаются человеком, но они гораздо проще для вскрытия. Использование длинных и бессмысленных паролей безусловно лучше с точки зрения криптостойкости, но человек обычно не может их запомнить и записывает на бумажке, которая потом либо теряется, либо попадает в руки злоумышленнику.

В последние годы много внимания уделяется разрешению этого противоречия, но рекомендации по выбору хороших паролей выходят за рамки этой статьи.

Именно из-за того, что неискушенные пользователи обычно выбирают либо короткие, либо осмысленные пароли, существуют два метода их вскрытия: *атака полным перебором и атака по словарю*.

С связи с резким ростом вычислительных мощностей атаки полным перебором имеют гораздо больше шансов на успех, чем раньше (см. также «*Малая длина ключа*»). Если для системы UNIX функция crypt(), которая отвечает за хэширование паролей, была реализована так, что выполнялась почти 1 секунду на машину класса PDP, то за двадцать лет скорость ее вычисления увеличилась в 15000 раз (!). Поэтому если раньше хакеры (и разработчики, которые ограничили длину пароля 8 символами) и представить себе не могли полный перебор, то сегодня такая атака в среднем приведет к успеху за 80 дней. Скорость перебора паролей для различных криптосистем приведена в табл. 1.

Однако вернемся на несколько лет назад, когда вычислительной мощности для полного перебора всех паролей не хватало. Тем не менее, хакерами был придуман остроумный метод, основанный на том, что в качестве пароля человеком выбирается существующее слово или какая-либо информация о себе или своих знакомых (имя, дата рождения и т.п.). Ну, а поскольку в любом языке не более 100 000 слов, то их перебор займет весьма небольшое время, и от 40 до 80% существующих паролей может быть угадано с помощью такой простой схемы, называемой «атакой по словарю».

*Таблица 1*

**Скорость полного перебора на компьютере класса Pentium/166**

|  |  |
| --- | --- |
|  | |
| Криптосистема | Скорость, паролей/cек |
| ARJ 2.50 | 350 000 |
| RC5 – 56 бит | 150 000 |
| LM-хэш | 50 000 |
| Novell Netware 3.x | 25 000 |
| MS Office 97 | 15 000 |
| UNIX – crypt() | 15 000 |
| RAR 2.0 | 1 000 |
| UNIX -MD5 | 500 |

(Кстати, до 80% этих паролей может быть угадано с использованием словаря размером всего 1000 слов!) Даже вирус Морриса (в 1988 г.!) применял такой способ, тем более что в UNIX «под рукой» часто оказывается файл-словарь, обычно используемый программами-корректорами. Что же касается «собственных» паролей, то файл /etc/passwd может дать немало информации о пользователе: его входное имя, имя и фамилию, домашний каталог. Вирус Морриса с успехом пользовался следующими предположениями:

* в качестве пароля берется входное имя пользователя;
* пароль представляет собой двойной повтор имени пользователя;
* то же, но прочитанное справа налево;  имя или фамилия пользователя;  то же, но в нижнем регистре.

Пусть сегодня пользователи уже понимают, что выбирать такие пароли нельзя, но до тех пор, пока с компьютером работает человек, эксперты по компьютерной безопасности не дождутся использования таких простых и радующих душу паролей, как *34jXs5U@bTa!6*. Поэтому даже искушенный пользователь хитрит и выбирает такие пароли, как *hope1, user1997, pAsSwOrD, toor, roottoor, parol, gfhjkm, asxz*. Видно, что все они, как правило, базируются на осмысленном слове и некотором простом правиле его преобразования: прибавить цифру, прибавить год, перевести через букву в другой регистр, записать слово наоборот, прибавить записанное наоборот слово, записать русское слово латинскими буквами, набрать русское слово на клавиатуре с латинской раскладкой, составить пароль из рядом расположенных на клавиатуре клавиш и т.п.

Поэтому не надо удивляться, если такой «хитрый» пароль будет вскрыт хакерами – они не глупее самих пользователей, и уже вставили в свои программы те правила, по которым может идти преобразование слов. В самых продвинутых программах (John The Ripper*,* Password Cracking Library) эти правила могут быть программируемыми и задаваться с помощью специального языка самим хакером.

Приведем пример эффективности такой стратегии перебора. Во многих книгах по безопасности предлагается выбирать в качестве надежного пароля два осмысленных слова, разделенных некоторым знаком, например *«good!password»*. Подсчитаем, за сколько времени в среднем будут сломаны такие пароли, если такое правило включено в набор программы-взломщика (пусть словарь 10000 слов, разделительными знаками могут быть 10 цифр и 32 знака препинания и специальных символа, машина класса Pentium со скоростью 15000 crypt/сек): =140 000 секунд или менее 1,5 дней!

*Приложение 3*

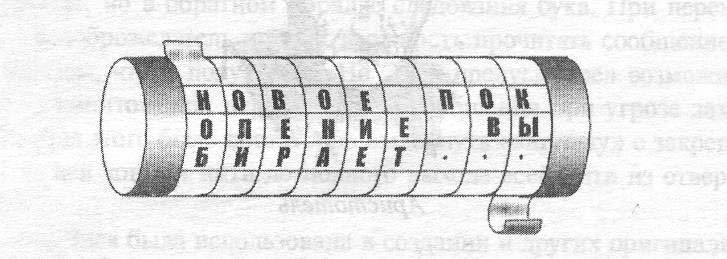
**ШИФРОВАЛЬНЫЕ МАШИНЫ**

**ШИФРОВАЛЬНЫЕ МАШИНЫ ДРЕВНОСТИ**

#### ● Сцитала

Одним из первых физических приборов, реализующих шифр перестановки, является прибор Сцитала. Он был изобретен в древней Спарте во времена Ликурга (5 век до нашей эры). Для зашифрования текста использовался цилиндр обусловленного диаметра. На цилиндр наматывался тонкий ремень из пергамента, и текст выписывался построчно по образующей цилиндра (вдоль его оси). Затем ремень сматывался и отправлялся получателю сообщения. Получатель наматывал его на цилиндр того же диаметра и читал текст по оси цилиндра.

Изобретение дешифровального устройства – Антисцитала – приписывается Аристотелю. Он предложил использовать конусообразное копье, на которое наматывается перехваченный ремень. Этот ремень передвигается по оси до того положения, пока не появляется открытый текст.



#### ●Диск Энея

Были и другие способы защиты информации, разработанные в античные времена.

В области тайнописи два изобретения принадлежат Энею.

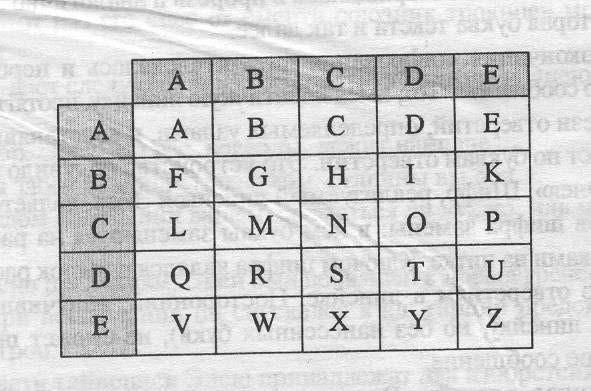
Первое из них – диск Энея. Его принцип был прост:

На диске диаметром 10–15 см и толщиной 1–2 см высверливались отверстия по числу букв алфавита. В центре диска помещалась катушка с намотанной на ней ниткой достаточной длины. При зашифровании нитка вытягивалась с катушки и последовательно протягивалась через отверстия в соответствии с буквами шифруемого текста. Диск и являлся посланием. Получатель послания последовательно вытягивал нитку из отверстий, что позволяло ему получать передаваемое сообщение, но в обратном порядке следования букв. При перехвате диска недоброжелатель имел возможность прочитать сообщение тем же образом, что и получатель. Но Эней предусмотрел возможность легкого уничтожения сообщения при угрозе захвата диска. Для этого было достаточно выдернуть катушку с закрепленным на ней концом нити до полного выхода всей нити из отверстий диска.

#### ●Квадрат Полибия

Еще одно изобретение древних греков – так называемый квадрат Полибия (Полибий – греческий государственный деятель, 3 век до нашей эры). Применительно к современному латинскому языку из 26 букв шифрование по этому квадрату заключалось в следующем. В квадрат размером 5 на 5 клеток выписываются все буквы алфавита, при этом буквы I, j учитываются за одну.

Шифруемая буква заменялась на координаты квадрата, в котором она записана.



Усложненный вариант квадрата Полибия заключается в записи букв в квадрат в произвольном порядке. Этот порядок является ключом.

##### ШИФРОВАЛЬНЫЕ МАШИНЫ ХХ ВЕКА

После окончания Гражданской войны Советское правительство продолжило наращивание усилий по становлению и развитию шифровальной службы как отдельной структуры во всех ведомствах и звеньях государства. Основные усилия были направлены на организацию скрытого управления войсками. С этой целью 15 ноября 1929 года были созданы «Курсы усовершенствования командного состава РККА», на которых готовились офицерские кадры шифровальной службы. Данные только в армии и на флоте, но так же готовили специалистов для органов НКВД, дипломатии и других ведомств. Подготовка велась серьезно и основательно. Их стен школы выходили высокопрофессиональные специалисты своего дела, обладающие всеми необходимыми навыками по организации и осуществлению скрытой связи на порученных участках работы. Именно эти Курсы впоследствии послужили основой создания среднего, а затем и высшего военного училища по подготовке офицеров-шифровальщиков.

В это время были впервые разработаны и внедрены в войска ручные шифры высокой степени стойкости, обеспечивающие необходимую скрытность любых сообщений.

Аналогичным образом развивалось шифровальное дело и в странах противоположного лагеря.

К началу 30-х годов прошлого столетия уже стало ясно, что имеющиеся ручные системы и способы шифрования и кодирования, сколько бы их ни совершенствовали и модернизировали, не в состоянии справиться со все возрастающими потоками информации в силу слабой скорости ее обработки. Встал вопрос о механизации данного процесса.

Попытки создания механических и электромеханических шифровальных и кодировочных машин в мире уже были. Это, в первую очередь, знаменитая шифрмашина германского производства «Энигма» и менее известные машины подобного класса – «Крига», «Фуллера и Бурга». Но вся эта шифровальная техника обладала малой криптографической стойкостью в силу того, что периодичность применяемого ключа составляла от 5 до 10 суток. Например, телеграммы германского правительства 30-х годов, зашифрованные с помощью «Энигмы», без особых сложностей читались дешифровальщиками Англии и США. Аналогичным образом обстояло дело и со знаменитой «Пурпурной кодировочной машиной» Японии. Кодограммы, получаемые посредством применения «пурпурного» кода, свободно декодировались американскими и советскими специалистами в конце 30-х – начале 40-х годов.

В СССР же к тому времени уже была создана довольно устойчивая система машинного шифрования, радикально отличающаяся от западных образцов. Теоретическую основу создания шифровальной техники такого типа впервые предложил на заседании научного совета РККА 29.6.1930 инженер-конструктор Волосок Иван Павлович. Впоследствии именно он стал ведущим конструктором многих образцов шифровальной и кодировочной техники довоенного и послевоенного периодов. За основу был взят принцип наложения комбинаций так называемой гаммы бесконечного ключа на комбинации знаков открытого текста. В итоге получалась нечитаемая криптограмма с гарантированной стойкостью. В качестве носителя знаков гаммы использовалась перфолента, изготовленная с помощью специального устройства, получившего название «X». Это было оригинальным и удачным изобретением, позволяющим получать шифрленты с гаммой случайной последовательности, что обеспечивало необходимую гарантированную стойкость шифрования любых сообщений.

###### ●Шифровальная машина М-100

В 1931 году при 8-м отделе Главного штаба РККА (а именно с этого времени шифровальная служба получила условное наименование – «8-й отдел») была создана техническая лаборатория. И уже в 1932 году под руководством инженера И.П. Волоска был создан первый опытный образец советской шифровальной машины, получившей название «ШМВ-1». В том же году были разработаны опытные образцы механических шифрующих приспособлений к телеграфным аппаратам, которые, впрочем, не получили дальнейшего развития. В силу своей громоздкости и механической ненадежности «ШМВ-1» не пошла в серию, однако послужила прототипом для последующего создания новых серийных образцов. И вскоре, 31 января 1934 года, начальник Главного штаба РККА маршал А.И. Егоров утверждает «Акт о принятии на вооружение электромеханической шифровальной машины «В-4», получившей впоследствии индекс «М-100». Именно с этого образца инженерно-криптографической мысли начался отсчет создания серии шифровальных машин, обеспечивающих и поныне гарантированную стойкость применяемых шифров.

Шифровальная машина М-100 состояла из трех основных узлов (клавиатура с контактными группами, лентопротяжный механизм с трансмиттером и приспособление, устанавливаемое на клавиатуру пишущей машинки) и семи дополнительных. Общий вес комплекта достигал 141 килограмма.

Только одни аккумуляторы для автономного питания электрической части машины весили 32 килограмма. Тем не менее данная техника выпускалась серийно и в 1938 году была успешно испытана в боевых условиях на Хасане, в 1939 году – на Халхин-Голе, в Испании и в 1939–1940 годах – во время финской войны. Шифрованная связь в этих конфликтах осуществлялась в звене Генеральный штаб – штаб армии. Непосредственное руководство эксплуатацией техники осуществлял И.П. Волосок.

Полученный боевой опыт осуществления скрытого управления войсками показал, что для успешного применения машинного шифрования необходима обособленная работа шифровальных органов РККА. Кроме того, необходимо было обеспечить конспирацию в работе шифровальщиков и их достаточную мобильность при передислокации войск. Для этих целей в 1939 году в США были закуплены 100 автобусов «Студебеккер» и переоборудованы под спецаппараты-шифрорганы. Стало возможным зашифровывать и расшифровывать телеграммы не только во время остановок, но и во время движения колонн.

К этому времени лаборатория 8-го отдела Генерального штаба РККА переросла в довольно мощное конструкторское бюро. Сам 8-й отдел возглавлял с 1938 года Петр Николаевич Белюсов. Это был талантливейший руководитель шифровальной службы, великолепный администратор, тонкий психолог, прослуживший на своем посту до 1961 года. Под его началом находилась сильная команда конструкторов первых советских шифровальных и кодировочных машин – Волосок И.П., Судаков П.А., Рытов В.Н., Строителев П.И., Гусев Н.И., Шарыгин Н.М., Козлов М.С.

#### ●Шифровальная машина М-101 («Изумруд»)

Козлов М.С. в тридцатые годы был первым офицером-шифровальщиком в СССР с высшим военным образованием. Закончив в 1937 году Военную академию связи в Ленинграде (по специальности «Военный инженер-электрик»), он находился на отдыхе в черноморском санатории Саки. Здесь он и познакомился с Волоском И.П. Тот, увидев в молодом офицере талант конструктора, предложил продолжить службу в лаборатории 8-го отдела ГШ РККА. Идея конструирования и предстоящие изобретения необычного захватили Михаила, и вся последующая жизнь его была связана с созданием и внедрением аппаратуры скрытой связи. Он принимал непосредственное участие в создании нового образца – шифрмашины М-101, пришедшей на смену морально устаревшей М100. Данная аппаратура шифрования состояла уже из двух основных узлов и по габаритам была уменьшена более чем в 6 раз, а по весу – более чем в 2 раза. Некоторые узлы и механизмы данной техники были разработаны и изобретены лично офицером Козловым М.С. Ведущим конструктором этой модели был Шарыгин Николай Михайлович. Общее руководство проектом осуществлял бессменный Волосок И.П. За создание и внедрение шифровальной машины М101 «Изумруд» в 1943 год Волоску И.П., Судакову П.А. и Рытову В.Н. были присуждены государственные премии. Орденами были награждены Шарыгин Н.М., Козлов М.С., Строителев П.И. и Гусев Н.И. Кроме того, Волоску И.П. была присвоена ученая степень «кандидат технических наук» (без защиты диссертации). В этом же году в войска было отправлено свыше 90 комплектов М101.

#### ●Шифровальная машина К-37 («Кристалл»)

К тому времени в войсках уже несколько лет успешно эксплуатировалась малогабаритная дисковая кодировочная машина К-37 «Кристалл», призванная собой заменить документы ручного кодирования в оперативном звене управления (армия – корпус – дивизия). Это было довольно компактное устройство, состоящее из одной упаковки весом 19 килограммов.

Только за 1940 год было выпущено 100 комплектов К-37. А всего к началу Великой Отечественной войны было принято на вооружение шифрорганов СССР свыше 150 комплектов К-37 и 96 комплектов М-100. К июню 1941 года всего на шифровальной службе насчитывалось 1857 человек. Шифровальная техника позволила в 5-6 раз по сравнению с ручным способом повысить скорость обработки телеграмм, сохраняя при этом гарантированную стойкость передаваемых сообщений.

В конце 1945 года были подведены итоги эксплуатации шифровальнокодировочной техники в действующей армии. В это же время проводится исследовательская работа по дальнейшему повышению криптостойкости применяемой техники, и намечаются пути и направления по созданию новых образцов аппаратуры с гарантированной стойкостью. Наряду с шифром гаммирования все большее применение находит шифр колонной замены, техническая реализация которого была заложена еще в 30-е годы в дисковой кодировочной машине К-37.

#### ●Шифровальная машина KRYHA

Механическое шифровальное устройство «KRYHA», созданное в 1924 году, активно использовалось немецкими дипломатами в годы Второй мировой войны, не знавшими того, что этот шифр был раскрыт американцами. Специальный шифровальный диск приводился в действие при помощи пружинного двигателя.

#### ●Шифровальная машина М-94

Механическое устройство М-94 использовалось в американской армии с 1924 по 1943 год. В основу его работы был положен принцип действия шифровального приспособления XVIII века, состоявшего из нескольких вращающихся дисков, по окружности которых были выгравированы буквы и цифры.

#### ●Шифровальное колесо Болтона

В основу этого устройства был положен шифровальный диск Леона Батиста Альберти, итальянского ученого и криптографа XV века. Это устройство представляло собой типичную конструкцию конца XIX века и производило простую замену одной буквы другой.

#### ●Шифровальная машина Хеберна

Эдвард Хеберн (1869–1952) был американским изобретателем-самоучкой. С 1909 года он разработал целую серию электромеханических шифровальных машин с вращающимися дисками. Шифровальные машины Хеберна предназначались для защиты секретной переписки между различными компаниями от возможного перехвата конкурентами. В 1915 году Э. Хеберн предложил конструкцию из двух пишущих машинок, соединенных проводами с центральным диском. Для своего времени это было весьма оригинальное решение. Впоследствии оно было использовано японцами при создании шифровальных машин для защиты дипломатической переписки.

Поначалу руководство американского военно-морского флота с большим вниманием отнеслось к достижениям Э. Хеберна, однако во время испытания стойкости предложенного шифра выдающемуся американскому криптографу Уильяму Фридману (1891–1969) удалось «взломать» его. Но неутомимый Эдвард разработал новую машину Mark II (SIGABA), которая оказалась самой надежной американской шифровальной системой в годы Второй мировой войны.

#### ●Шифровальная машина Конвертер М-209



В 1934 году по заданию французских спецслужб шведский криптограф Борис Хагелин (1892–1983) разработал оригинальную шифровальную машину. На ее основе ему удалось создать знаменитый Конвертер М-209. Эта шифровальная машина, выпущенная серией более 140 тысяч экземпляров, использовалась американской армией во время Второй мировой войны. Она представляла собой портативное устройство с набором специальных роторов для зашифровки и расшифровки секретных сообщений. Шифруемое с помощью М-209 сообщение распечатывалось на бумажной ленте в виде пятизначных групп. Затем оно передавалось по радио и расшифровывалось на приемном конце при помощи другой такой же машины.

#### ●Шифровальное устройство CD-57

Разработано Б. Хагелином для французской секретной полиции. Оно было столь миниатюрным, что легко умещалось в кармане и приводилось действие одним большим пальцем, оставляя вторую руку свободной для записи сообщений.

#### ●Шифровальная машина Enigma



Основой стратегии фашистской Германии во Второй мировой войне была ставка на внезапность и высокую мобильность всех видов вооруженных сил в ходе военных действий. Разумеется, это возможно только при наличии надежных быстродействующих каналов связи, хорошо защищенных от перехвата противником.

Первая версия шифровальной машины Enigma предназначалась для защиты коммерческих секретов от промышленного шпионажа. Затем появились модели для использования в армии, службе безопасности и разведке. Уже в ходе войны Enigma подвергалась постоянной модернизации. Только в 1943 году, используя электронно-вычислительную технику, удалось раскрыть применявшийся в ней шифр. По мнению историков, этот факт сыграл решающую роль в победе союзников над нацистами во Второй мировой войне. Созданная в 1923 году Enigma представляла собой электромеханическое устройство для зашифровки и расшифровки текстовой информации. Каждая буква сообщения зашифровывалась самостоятельно при помощи целого набора механических роторов и электрических разъемов.

##### Вскрытие шифра Enigm'ы

В 1939 году англичане начали работы по вскрытию шифра немецкой шифровальной машины Enigma. Для этого неподалеку от Лондона в местечке Блечли Парк была организована «шифровальная школа» (Code and Cipher School at Bletchly Park) – исследовательское подразделение, где собрали математиков, лингвистов и других специалистов в области криптографии для расшифровки кодов Enigm'ы, а затем и других шифров.

Ученые столкнулись с двумя сложными проблемами: раскрытием начальной установки шифрующих элементов, производимой в начале каждого рабочего дня, и конструктивными отличиями шифровальных машин Enigma, применявшихся в различных ведомствах Германии.

Перед войной польским криптографам удалось расшифровать некоторые сообщения, выполненные с помощью Enigm'ы. Но, увы, немцы постоянно совершенствовали свою шифровальную технику, делая Enigm'у все сложнее. И хотя англичанам удалось развить достижение поляков, они поначалу несколько не поспевали за процессом модернизации Enigm'ы. Наконец, молодому и очень талантливому математику Алану Терингу (1912–1954) удалось сконструировать специальную релейную схему, которая существенно ускоряла процесс дешифровки. Уже в 24 года он изложил в своей работе некоторые принципы действия современных компьютеров. Работая в Блечли Парк, он всерьез увлекся использованием появившихся тогда первых электронно-вычислительных машин для вскрытия шифров. Именно с их помощью удалось взломать шифр даже такой сложной шифровальной машины, как немецкая «Geheimschreiber».

В ходе работ по вскрытию шифров английские криптографы провели громадную аналитическую работу. Основываясь на догадках и предположениях, строя всевозможные гипотезы, много экспериментируя, они пытались по структуре буквенных сочетаний в зашифрованных сообщениях распознать начальную установку немецких шифровальных машин. Так, например, математик Гордон Уэлчман разработал своеобразный сетевой анализ, помогавший отслеживать, из каких именно организаций противника поступали зашифрованные сообщения. Это позволяло идентифицировать модель Enigm'ы, использовавшейся для зашифровки, и экономить массу времени при вскрытии ее шифра.

Следует отметить, что сообщения немецкого военно-морского флота расшифровывались англичанами на всем протяжении 1941–1945 годов в основном благодаря захваченным образцам шифровальных машин и документам, содержавшим важную информацию об их работе.

#### ●Шифровальная машина «Geheimschreiber»

Электромеханическое устройство с таким зубодробительным названием, которое по-немецки означает всего-навсего «некто, пишущий что-то секретное», обладало 10 или даже 12 шифровальными колесами-роторами. По понятным причинам «взломать» шифр Geheimshreiber'а было исключительно трудно. Эта шифровальная машина была очень громоздкой и устанавливалась только в основных центрах связи Германии на территориях, контролируемых немецкими властями.

#### ● «Пурпурная» шифровальная машина

В этом японском шифровальном устройстве две электрические пишущие машинки были соединены при помощи двух специальных переключающих устройств. В то время как исходный текст печатался на первой машинке, на второй появлялось зашифрованное сообщение.

##### «Пурпурный код»

В 1939 году японцы начали использовать для передачи дипломатической корреспонденции новую шифровальную машину «Типа 97». В Соединенных Штатах она получила романтическое кодовое название «Пурпурной». Тип 97 был модернизацией предыдущей модели, так называемой «Красной» машины. К тому времени американцы уже «взломали» ее шифр, и специалисты соответствующего подразделения радиоконтрразведки начали работу по разгадыванию нового шифра. Ведущую роль в этих работах играл Уильям Фридман (1891– 1969). Эмигрант из Советского Союза, он к этому времени был уже автором нескольких оригинальных работ, где излагались основные принципы современной криптографии. Его жена Элизабет также была криптографом, и иногда они работали вместе. После войны успешная работа Фридмана привела к созданию в США Агентства Национальной безопасности, и по сей день отвечающего за безопасность государственных каналов связи, зашифровку своих и расшифровку чужих сообщений.

Но это случилось в далеком 1952 году, а пока американцы продолжали перехват сообщений, зашифрованных «Пурпурным» и «Красным» кодами. Это было единственной информацией, которая могла бы помочь создать свой аналог «Пурпурной» машины. Прорыв состоялся, когда криптографы попытались использовать шаговые искатели, применяемые в телефонии. По счастливому совпадению они работали, основываясь на том же принципе, что и переключатели «Пурпурной» машины. В конце 1940 года У. Фридман и его команда из военно-морской контрразведки были в состоянии создать свой вариант «Пурпурной» машины. Он оказался столь эффективным, что текст официального объявления войны Японией, направленный ею в свое посольство в Вашингтоне за день до атаки на Перл Харбор (чтобы посольские шифровальщики успели расшифровать и распечатать сообщение), оказался на столе американской разведки еще до официального вручения текста японцами.

*Приложение 4*

## СТАНДАРТЫ ШИФРОВАНИЯ Стандарт шифрования DES

### *История*

В 1972 году, после проведения исследования потребностей правительства США в компьютерной безопасности, американское НБС (Национальное Бюро Стандартов) – теперь переименовано НИСТ (Национальный Институт Стандартов и Технологий) – определило необходимость в общеправительственном стандарте шифрования некритичной информации. 15 мая 1973 года после консультации с УНБ (Управлением национальной безопасности) НБС объявило конкурс на шифр, который удовлетворит строгим критериям проекта, но ни один конкурсант не обеспечивал выполнение всех требований. Второй конкурс был начат 27 августа 1974 года. На сей раз, шифр представленный IBM и развитый в течение периода 1973–1974, сочли приемлемым, он был основан на более раннем алгоритме Хорста Фейстеля.

17 марта 1975 года предложенный алгоритм DES был издан в Федеральном Регистре. В следующем году было проведено 2 открытых симпозиума по обсуждению этого стандарта, где подверглись жесткой критике изменения, внесенные в алгоритм УНБ: уменьшение первоначальной длины ключа и таинственные S-перестановки. УНБ подозревалось в сознательном ослаблении алгоритма с целью, чтобы УНБ могло легко просматривать зашифрованные сообщения. После чего сенатом США была проведена проверка действий УНБ, результатом которой стало заявление, опубликованное в 1978-м, в котором говорилось о том, что в процессе разработки DES УНБ убедило IBM, что уменьшенной длины ключа более чем достаточно для всех коммерческих приложений, использующих DES, косвенно помогало в разработке S-перестановок, а также, что окончательный алгоритм DES был лучшим, по их мнению, алгоритмом шифрования и был лишен статистической или математической слабости. Также было обнаружено, что УНБ никогда не вмешивалось в разработку этого алгоритма.

Часть подозрений в скрытой слабости S-перестановок была снята в 1990-м, когда были опубликованы результаты независимых исследований Эли Бьяма (Eli Biham) и Ади Шамира (Adi Shamir) по дифференциальному криптоанализу – основному методу взлома блочных алгоритмов шифрования с симметричным ключом. S-перестановки алгоритма DES оказались намного более устойчивыми к атакам, чем если бы их выбрали случайно. Это означает, что IBM была известна такая техника анализа еще в 70-х годах XX века.

DES (англ. Data Encryption Standart) – симметричный алгоритм шифрования, в котором один ключ используется как для зашифровывания, так и для расшифровывания сообщений. Также известен как алгоритм шифрования данных DEA (англ. Data Encryption Algorithm). Алгоритм, который с 1977 года является федеральным стандартом для шифрования в США. Создан в исследовательской лаборатории корпорации IBM (1972–1975 годы) под руководством дра У. Тачмена. Длинна ключа 56 бит (64, но 8 из них не используются). Размер блока 64 бит. Число раундов – 16. Является симметричным алгоритмом. С использованием современных вычислительных систем вскрывается за время около 1-2 дней. Именно по причине недостаточной криптостойкости (на сегодняшний день) был объявлен конкурс на новый стандарт (AES). Существует крипточип (Deep Crack), который позволяет перебирать ключи для DES со скоростью 88 миллиардов ключей в секунду. Этот чип взламывает DES за время, не превышающее 3 дня. Современные распределенные вычислительные системы и суперкомпьютеры вскрывают зашифрованные по алгоритму DES сообщения за еще меньшее время.

Один из шагов алгоритма DES заключается в следующем.

Входной блок данных делится пополам на левую (L’) и правую (R’) части. После этого формируется выходной массив так, что его левая часть L’’ представлена правой частью R’ входного слова, из 32-битового слова R’ с помощью битовых перестановок формируется 48-битовое слово. Полученное 48-битовое слово «XOR’ится» с 48-битовым раундовым ключом. Результирующее 48битовое слово разбивается на восемь 6-битовых групп, каждая 6-битовая группа посредством соответствующего S-блока заменяется на 4-битовую группу, и из полученных восьми 4-битовых групп составляется 32-битовое слово. Полученное слово «XOR’ится» с L’, в результате получается R’’. Можно убедиться, что все произведенные операции могут быть обращены и расшифровывание осуществляется за число операций, линейно зависящее от размера блока. После нескольких таких перемешиваний можно считать, что каждый бит выходного блока шифровки может зависеть от каждого бита сообщения.

Поскольку текст, зашифрованный двойным алгоритмом DES, оказывается нестойким при криптографической атаке, то текст шифруется три раза. Таким образом, длина ключа возрастает до 168 бит (56×3).

Типы тройного шифрования DES:

DES-EEE3: шифруется три раза с тремя различными ключами.

DES-EDE3: три DES-операции «шифровка-расшифровка-шифровка» с тремя различными ключами.

DES-EEE2 и DES-EDE2: как и предыдущие, за исключением того, что первая и третья операции используют одинаковый ключ.

**Стандарт шифрования данных (DES)**

Выражение Data Encryption Standard дословно переводится как «стандарт шифрования данных». И, наверное, подобное название вполне оправданно. Ведь это самый известный в мире алгоритм шифрования информации, на котором основываются множество других вариантов кодирования данных. В период с 1977 по 2001 год DES был федеральным стандартом Соединенных Штатов Америки. За это время он приобрел такую популярность, что стал основой многих разработок в области блочных шифров, а также был использован в большом количестве коммерческих продуктов.

По своей сути DES – симметричный блочный алгоритм шифрования. Его архитектуру можно классифицировать как классическую сбалансированную сеть Фейстеля с начальной и конечной битовыми перестановками общего вида.

Основные параметры DES таковы: pазмер блока – 64 бита; размер ключа – 64 бита; число раундов – 16;

pазмер ключевого элемента – 48 бит; число ключевых элементов – 16.

**Экскурс в историю**

Стандарт DES подвергался многим и, надо признаться, вполне справедливым нападкам. Самый главный недостаток этого алгоритма шифрования лежит наверху. Речь идет о чрезвычайно малой длине ключа. 64 бита явно недостаточно для обеспечения надежной защиты. А если учесть, что реально используются в процессе шифрования только 56 бит, а остальные служат только для контроля четности, то становится понятно, почему алгоритм DES нельзя применять сегодня для защиты важной информации.

Но почему же была выбрана столь малая длина ключа? Многие люди считают, что это произошло по настоянию Агентства национальной безопасности США (АНБ). Эта спецслужба Америки всегда была самой хорошо оснащенной в компьютерном плане. Считается, что ее возможности позволяли вскрывать информацию, зашифрованную алгоритмом DES, за вполне приемлемое время. Если же использовать алгоритм с ключом большей длины, то АНБ в то время не могло декодировать интересующие ее данные. Правда, подобная теория больше похожа на домыслы журналистов, чем на правду. Скорее всего, дело обстояло совсем не так. Компьютеры в 1977 году, когда разрабатывался DES, мягко говоря, очень сильно отличались по мощности от современных машин. Для них шифрование информации и с таким ключом – процесс весьма длительный. Вообще, в 70–80-х годах большую роль в разрабатываемых алгоритмах играла не надежность, а именно скорость.

Хотя, если говорить откровенно, скорость DES при шифровании информации также оставляет желать лучшего. Давайте для примера сравним его с российским алгоритмом, описанным в ГОСТе 28147-89. Кстати, уважаемые читатели, обратите внимание, что, несмотря на 1989 год принятия этого стандарта, разработан он был примерно в то же время, что и DES. ГОСТ 28147-89 предполагает использование ключа длиной 128 бит. Естественно, это обеспечивает большую надежность. Кроме того, согласно исследованиям, процесс шифрования информации на обычном компьютере с процессором Intel x86 с использованием российского алгоритма осуществляется в 3–5 раз быстрее, нежели с применением DES. Причина этого кроется в архитектуре шифра, которая сразу же разрабатывалась с учетом как аппаратных, так и программных реализаций.

И все-таки, несмотря на очевидные недостатки, стандарт DES прослужил до 2001 года. И за это время не было найдено никаких «обходных путей», облегчающих его взлом. Специалисты предложили несколько способов немного ускорить взлом способом полного перебора всех вариантов ключа. Так, например, в 80-х годах проводились исследования, которые закончились удачной расшифровкой информации. Правда, на этот процесс ушло больше 50 дней. Естественно, подобный срок вряд ли можно считать приемлемым. Но с тех пор возможности вычислительной техники выросли очень сильно. Так, например, в феврале 1998 года алгоритм DES был взломан за 39 дней. Правда, для этого потребовалась работа общенациональной сети компьютеров. Небольшой успех. Но уже в июле того же года организация Electronic Frontiers Foundation объявила об успешном декодировании текста всего за 56 часов. Подобное время является вполне приемлемым. Тем более что работа велась с использованием всего одного обычного персонального компьютера. Конечно, этот ПК был переоборудован. Стоимость дополнительно установленной аппаратуры составила около 200 тысяч долларов. На первый взгляд эта цифра просто астрономическая. Но, во-первых, сегодня подобное оборудование гораздо дешевле, а во-вторых, государственные и коммерческие тайны зачастую стоят в десятки раз дороже. Так что алгоритм DES на сегодняшний день не может считаться действительно надежным.

### Режимы шифрования

Стандарт DES предусматривает наличие нескольких режимов работы алгоритма шифрования. Первый из них – ECB (электронная кодовая книга). Это наиболее простой вариант: каждый блок исходного текста (64 бита) кодируется с помощью одного и того же 56-битового ключа. Второй режим, CBC, несколько сложнее. Каждый блок исходного текста перед обработкой суммируется с предыдущим, уже зашифрованным. Таким образом, один и тот же отрывок информации в закодированном виде будет выглядеть по-разному в зависимости от его положения. CBC – самый часто используемый режим шифрования информации по алгоритму DES.

Остальные режимы алгоритма DES используются гораздо реже первых двух. Режим FSB – шифрование с обратной связью. Он позволяет использовать алгоритм DES с блоками исходной информации менее 64 бит. Режим OFB фактически превращает начальный стандарт в потоковый шифр.

#### Разновидности DES

За более чем двадцатилетний срок действия стандарта DES специалисты предлагали различные «улучшенные» варианты этого алгоритма шифрования. Первым из них был G-DES. Создатели этого варианта утверждали, что смогли добиться значительного повышения скорости шифрования данных при оставшейся на прежнем уровне надежности. Это якобы достигалось за счет использования DES-основанного шифра с большим размером блока, но при тех же требованиях к вычислительным ресурсам. Правда, независимые эксперты вскоре доказали, что G-DES довольно быстро взламывается.

Во второй разновидности DES использовались субключи. То есть перед началом процесса шифрования на основе 56-битового секретного ключа пользователя алгоритм получает шестнадцать 48-битовых субключей для использования в каждом из 16 раундов. К сожалению, оказалось, что это замедляет процесс кодирования и при этом практически не повышает надежность защиты. Конечно, взлом способом полного поиска ключа сильно усложняется. Но на эффективность других атак, например дифференциального или линейного криптоанализа, использование субключей влияет не сильно.

Следующая разновидность стандарта DES – DESX. Она представляет собой усиленный вариант алгоритма, поддерживаемый инструментарием RSA Security. В этом варианте шифра каждый бит входного открытого текста логически суммируется с 64 битами дополнительного ключа. Такой способ практически ничего не добавляет к защите от атак с помощью дифференциального или линейного криптоанализа, но зато существенно увеличивает стойкость шифра к полному перебору всех вариантов ключей. Причем он очень прост в реализации, а также не отличается от стандартного DES по скорости работы. Поэтому DESX получил некоторое распространение в различных коммерческих продуктах.

Triple DES – наиболее удачный улучшенный вариант стандарта DES. Его суть заключается в последовательном трехкратном применении шифрования к исходному тексту. При этом ANSI (Американский национальный институт станда́ртов – American National Standards Institute, ANSI – объединение американских промышленных и деловых групп, разрабатывающее торговые и коммуникационные стандарты) X9.52 предусматривает три различных варианта использования Triple DES. Первый – это применение к тексту алгоритма шифрования с тремя различными ключами. Второй вариант – когда первый и третий ключи одинаковые, а средний – отличается от них. Ну и последний случай – все три ключа абсолютно идентичны. Такой способ шифрования информации получил достаточно большую популярность. Многие компания, занимающиеся разработкой коммерческих продуктов, выпустили даже специальные версии своих программ, в которые включили Triple DES.

## Стандарт шифрования ГОСТ 28147-89

### *История*

ГОСТ 28147-89 – советский и российский стандарт симметричного шифрования, введенный в 1990 году. Полное название «ГОСТ 28147-89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования». Американцы называют его «русский аналог DES». Но в отличие от DES алгоритм ГОСТ 28147-89 он значительно более криптостоек и сложен.

Разработан в одном из институтов КГБ в конце 1970-х годов. В 1989 году принял статус официального стандарта шифрования СССР, а после распада СССР – стандарта Российской Федерации на шифрование и имитозащиту данных. Авторы, само собой, неизвестны. Алгоритм использует несколько различных этапов на каждой итерации.

С момента опубликования ГОСТа на нем стоял ограничительный гриф «Для служебного пользования», и формально шифр был объявлен «полностью открытым» только в мае 1994 года. К сожалению, история создания шифра и критерии его проектирования до сих пор неизвестны.

**Описание**

ГОСТ 28147-89 – блочный шифр с 256-битным ключом и 32 циклами преобразования, оперирующий 64-битными блоками. Основа алгоритма шифра – сеть Фейстеля. Базовым режимом шифрования по ГОСТ 28147-89 является режим простой замены, хотя он имеет три режима работы (простая замена, гаммирование и гаммирование с обратной связью) и один режим выработки имитовставки. Первый из режимов шифрования предназначен для шифрования ключевой информации и не может использоваться для шифрования других данных; для этого предусмотрены два других режима шифрования. Режим выработки ИМИТОВСТАВКИ (криптографической контрольной комбинации) предназначен для ИМИТОЗАЩИТЫ шифруемых данных, то есть для их защиты от случайных или преднамеренных несанкционированных изменений.

При разработке этого криптоалгоритма были учтены недостатки и недоработки DES. Так, он оптимизирован на программную реализацию, использует вдвое больше раундов шифрования с гораздо более простыми операциями и в 5 раз более длинный ключ. Кроме того, S-блоки в ГОСТ не являются фиксированными и представляют собой долговременный ключ.

Рекомендован к применению в системах защиты информации.

Для зашифрования в этом режиме открытый текст сначала разбивается на левую и правую половины L и R. На i-м цикле используется подключ ki:

Ri + 1 = Li

= xor)



(



Для генерации подключей исходный 256-битный ключ разбивается на восемь 32-битных блоков: K1…K8.

Расшифрование выполняется так же, как и зашифрование, но инвертируется порядок подключей Ki.

Функция f(Li,Ki) вычисляется следующим образом:

Ri-1 и Ki складываются по модулю 232.

Результат разбивается на восемь 4-битовых подпоследовательностей, каждая из которых поступает на вход своего S-блока. Общее количество S-блоков ГОСТа – восемь, т.е. столько же, сколько и подпоследовательностей. Каждый S-блок представляет собой перестановку чисел от 0 до 15. Первая 4-битная подпоследовательность попадает на вход первого S-блока, вторая – на вход второго и т.д.

Если S-блок выглядит так:

1, 15, 13, 0, 5, 7, 10, 4, 9, 2, 3, 14, 6, 11, 8, 12

и на входе S-блока 0, то на выходе будет 1, если 5, то на выходе будет 7 и т.д.

Выходы всех восьми S-блоков объединяются в 32-битное слово, затем все слово циклически сдвигается влево на 11 бит.

Все восемь S-блоков могут быть различными. Фактически они могут являтся дополнительным ключевым материалом, но чаще являются параметром схемы, общим для определенной группы пользователей, например, по некоторым данным. Центральный Банк использует следующие S-блоки:

№ S-блока Значение

1. 4 10 9 2 13 8 0 14 6 11 1 12 7 15 5 3
2. 14 11 4 12 6 13 15 10 2 3 8 1 0 7 5 9
3. 5 8 1 13 10 3 4 2 14 15 12 7 6 0 9 11
4. 7 13 10 1 0 8 9 15 14 4 6 12 11 2 5 3
5. 6 12 7 1 5 15 13 8 4 10 9 14 0 3 11 2
6. 4 11 10 0 7 2 1 13 3 6 8 5 9 12 15 14
7. 13 11 4 1 3 15 5 9 0 10 14 7 6 8 2 12
8. 1 15 13 0 5 7 10 4 9 2 3 14 6 11 8 12

Алгоритм построен по тому же принципу, что и DES, – классический блочный шифр с секретным ключом, однако он отличается от DES большей длиной ключа, большим количеством раундов и более простой схемой построения самих раундов. Ниже приведены его основные параметры, для удобства – в сравнении с параметрами DES.

В силу намного большей длины ключа ГОСТ гораздо устойчивей, чем DES, к вскрытию путем полного перебора по множеству возможных значений ключа.

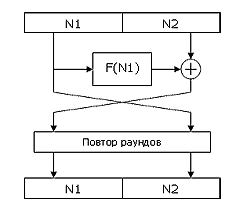
Функция шифрования ГОСТ гораздо проще функции шифрования DES, она не содержит операций битовых перестановок, которыми изобилует DES, и которые крайне неэффективно реализуются на современных универсальных процессорах (хотя их очень просто реализовать аппаратно – путем разводки проводников в кристалле или на плате). Таким образом, при вдвое большем количестве раундов (32 против 16) программная реализация ГОСТ на процессорах Intel x86 в среднем более чем в два раза превосходит по быстродействию реализацию DES.

На сегодняшний день существует огромное множество различных алгоритмов шифрования. Надежность некоторых из них подтверждена исследованиями экспертов в области криптографии, другие не могут этим похвастать. А еще есть целая группа алгоритмов, принятых в качестве стандартов в тех или иных странах. Естественно, это говорит об их высоком качестве: надежности, скорости обработки данных, универсальности и т.п. Еще одним преимуществом этой группы является то, что по мере устаревания алгоритма в качестве национального стандарта принимается новая, более современная разработка. Так было, например, с американским стандартом шифрования. Первоначально им был алгоритм DES, а в конце прошлого века ему на смену пришла более современная разработка AES.

В нашей стране в качестве стандарта используется технология, описанная в ГОСТе 28147-89 «Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования». Как видно из названия, этот ГОСТ был принят в 1989 году и с тех пор не изменялся. Но мало кто знает, что этот алгоритм шифрования разработали в КГБ еще в конце 70-х годов. Но стоит ли доверять такому старому алгоритму? Не подведет ли он? И не лучше ли обратить свое внимание на более современные разработки? Что ж, давайте попробуем разобраться в этом вопросе.

Во-первых, нужно учитывать, что алгоритм, описанный в ГОСТе 28147-89, создавался с достаточно большим «запасом прочности». По этому параметру он на порядок превосходил американский DES, который сначала заменили на тройной DES (то есть одну и ту же информацию «прогоняли» три раза), а потом на AES. Таким образом, и на сегодняшний день криптостойкость российского стандарта вполне удовлетворяет всем современным требованиям. Вторая причина большого распространения ГОСТа 28147-89 – наше законодательство. Так, например, государственные организации и многие коммерческие структуры обязаны использовать для защиты данных сертифицированные средства защиты. Однако получение сертификата возможно только в том случае, если «в указанных криптосредствах реализованы криптографические алгоритмы, объявленные государственными или отраслевыми стандартами Российской Федерации». Таким образом, у многих компаний просто-напросто нет выбора. Однако, как мы уже говорили, ГОСТ 28147-89 удовлетворяет всем современным требованиям, а поэтому вряд ли кто-то будет расстраиваться из-за необходимости его использования.

Алгоритм, описанный в ГОСТе 28147-89, является типичным представителем класса симметричных. В его основе лежит так называемая сеть Фейстеля. Принцип ее работы заключается в следующем. Первоначальный блок данных (в ГОСТе 28147-89 это 64 бита) разбивается на несколько подблоков. Часть этих подблоков преобразовывается по определенным законам, а потом накладывается на необработанные подблоки. Затем они меняются местами и снова обрабатываются. И все эти действия повторяются определенное число раз.



На основе сети Фейстеля построен целый ряд различных алгоритмов. Вот только, несмотря на внешнее сходство, их криптостойкость и скорость работы очень сильно различаются. Все зависит от действий, которые выполняются над подблоками. Именно поэтому они и называются «основным криптографическим преобразованием». В ГОСТе 28147-89 используются относительно простые для реализации, быстрые для исполнения и устойчивые к взлому операции. Таким образом, этот алгоритм отличается в лучшую сторону от большинства своих «собратьев».

Алгоритм, описанный в ГОСТе 28147-89, может работать в трех различных режимах. Первый из них – простая замена. В этом случае блоки шифруются независимо друг от друга и от положения в массиве исходной информации. То есть две одинаковые последовательности размером в 64 бита и после обработки останутся одинаковыми. Нужно ли говорить, что это недопустимо для качественного и надежного алгоритма шифрования. Кроме того, в режиме простой замены очень желательно, чтобы объем исходного массива данных был кратен 64 битам. В противном случае наблюдается снижение криптоустойчивости алгоритма к некоторым видам атак.

На первый взгляд кажется, что режим простой замены абсолютно непригоден для практического использования. Однако на самом деле это не совсем так. Дело в том, что есть информация, которая практически всегда кратна 64 битам и не имеет в себе повторяющихся блоков этого размера. Речь идет, конечно же, о ключах шифрования. И действительно, задача их кодирования достаточно часто встает перед пользователями. И именно для ее решения подходит режим простой замены. Тем более что работает он быстрее других способов шифрования.

Вторым режимом работы алгоритма, описанного в ГОСТе 28147-89, является гаммирование. Этим словом обозначают «наложение (снятие) на открытые (зашифрованные) данные криптографической гаммы, то есть последовательности элементов данных, вырабатываемых с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных». В рассматриваемом алгоритме для этой цели используется операция побитового сложения по модулю 2, поскольку она является обратной самой себе и к тому же наиболее просто реализуется аппаратно.

Таким образом, использование гаммирования решает обе проблемы простой замены. Во-первых, применение различных гамм для шифрования одинаковых последовательностей приводит к тому, что в закодированном тексте они отличаются друг от друга. Ну и, во-вторых, никто не может помешать «обрезать» гамму до нужного размера для шифрования блоков, отличных от стандартных. Таким образом, режим гаммирования избавлен от всех недостатков простой замены и может использоваться для надежного шифрования любой информации.

Третий и последний режим работы алгоритма, описанный в ГОСТе 2814789, – это гаммирование с обратной связью. В принципе, он очень похож на предыдущий. Единственное отличие заключается в том, что очередной элемент гаммы вырабатывается с помощью определенных преобразований предыдущей последовательности информации. То есть фактически получается, что результат кодирования блока данных зависит от ключа шифрования и предыдущего уже зашифрованного блока. Таким образом, мы имеем связь всего закодированного текста. И если кто-то поменяет в нем какой-либо бит, то при расшифровке искаженным окажется не один, а два блока данных. В принципе, сильного влияния на криптостойкость обратная связь не оказывает. Однако она может послужить дополнительной защитой от некоторых типов атак.

В ГОСТе 28147-89 описана еще одна очень полезная функция алгоритма шифрования. Речь идет о возможности создания имитовставки. Что это? Имитовставка – контрольная комбинация, зависящая от открытых данных и секретной ключевой информации. Она нужна для того, чтобы обнаружить все случайные или преднамеренные изменения в зашифрованных данных. На первый взгляд кажется, что имитовставка не нужна. Однако на самом деле это не так. Конечно, обнаружить внесенные в текст изменения несложно. Тем более что полноценно заменить одни слова другими без знания ключа нельзя. Но если злоумышленник внесет изменения в файл, полученный путем шифрования бинарной, звуковой, графической или видеоинформации, то при декодировании искажения перейдут и на исходные данные. И заметить их зачастую простонапросто нереально. Именно тогда имитовставка и оказывается незаменимой. Да и в случае с текстом ее использование позволяет точно узнать, вносились изменения в зашифрованный файл или нет.

Итак, как мы видим, алгоритм, принятый в качестве государственного стандарта РФ и описанный в ГОСТе 28147-89, сохраняет свою актуальность и сегодня. Он обеспечивает надежную защиту любых данных и не может быть взломан с помощью известных типов криптографических атак. Хотя, конечно, очень многое зависит от конкретной реализации. Часто бывает так, что на базе надежного алгоритма из-за ошибок при разработке создаются системы шифрования, не удовлетворяющие современным требованиям. А поэтому лучше всегда отдавать предпочтение хорошо зарекомендовавшим себя сертифицированным продуктам.

### *Достоинства ГОСТа*

* бесперспективность силовой атаки (XSL-атаки в учет не берутся, т.к.

их эффективность на данный момент полностью не доказана);

* эффективность реализации и соответственно высокое быстродействие на современных компьютерах.

### *Недостатки ГОСТ*

Стандарт не специфицирует алгоритм генерации S-блоков (таблицы замен). С одной стороны, это может являться дополнительной секретной информацией (помимо ключа), а с другой – поднимает ряд проблем:

* нельзя определить криптостойкость алгоритма, не зная таблицы замен

(таблицы бывают «удачные» и «неудачные»);

* реализации алгоритма от различных производителей могут использовать разные таблицы замен и могут быть несовместимы между собой;
* генерацией и проверкой S-блоков для реализаций алгоритма занимался ФАПСИ (при лицензировании алгоритма) – Федеральное агентство правительственной связи и информации. Не было уверенности, что эта организация не выдаст «специальных» блоков. В настоящее время упразднено. Его функции распределены между ФСБ и Минобороны РФ.

## Сравнение стандартов шифрования

|  |  |  |
| --- | --- | --- |
| Параметры | DES | ГОСТ 28147-89 |
| Начальное шифрование | Перестановка | Разбивается на 2 блока и в каждом осуществляются перестановки |
| Размер ключа | 56 | 8\*32=256 |
| Количество циклов | 16 | 32 |
| Размер блока | 64 | 64 |

|  |  |  |
| --- | --- | --- |
| Режим работы | Блочный шифр, самосинхронизирующийся. Электронная кодовая книга, цепочный режим, обратная связь по выводу, обратная связь по шифратору | Самосинхронизирующийся. Режим простой замены, гаммирования, гаммирования с обратной связью, выработки имитовставки |
| Блок подстановки | Нельзя изменять | Можно изменять |
| Заполнение узлов замены | Явно указано в стандарте | Является секретным элементом |
| Скорость | Быстрее | Медленнее |
| Криптостойкость | Ниже | Выше |
| Преобразование | Модулярное | Модулярное |
| Перестановка | По таблице | --------- |
| Конечная перестановка | Обратная по отношению к первоначальной | --------- |
| Недостатки | Небольшое количество ключей | Нельзя оценить стойкость системы, требует много ресурсов  системы |
| Основание | 1977 г. | 1989 г. |

## Стандарт шифрования ГОСТ 34.12-2015

Стандарт представляет два базовых блочных шифра с длинами блоков *n* = 128 бит и *n* = 64 бит и длинами ключей *k* = 256 бит:

* шифр с длиной блока *n* = 128 бит – блочный шифр «Кузнечик»

(«Kuznyechik») – *SP-сеть*;

* шифр с длиной блока *n* = 64 бит – блочный шифр «Магма» («Magma» – *сеть Фейстеля*. Преобразования

Шифрование основано на последовательном применении нескольких однотипных раундов, каждый из которых содержит три преобразования: сложение с раундовым ключом, преобразование блоком подстановок и линейное преобразование.

128-битный входной вектор очередного раунда складывается побитно с раундовым ключом:

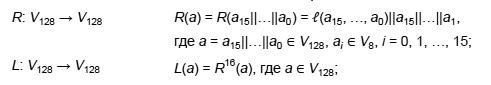


Нелинейное преобразование представляет собой применение к каждому 8битному подвектору 128-битного входного вектора фиксированной подстановки:

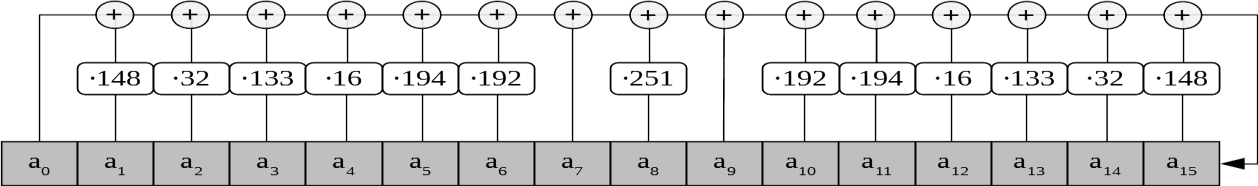


В «Кузнечике» используется та же подстановка, что и в хэш-функции «Стрибог».

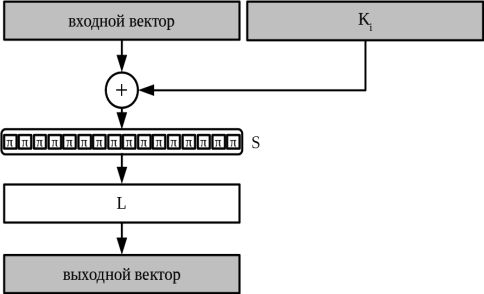
Линейное преобразование может быть реализовано не только как обычно в блочных шифрах – матрицей, но и с помощью РСЛОС – линейного регистра сдвига с обратной связью, который движется 16 раз.



Сам регистр реализуется над полем Галуа по модулю неприводимого многочлена степени 8: :

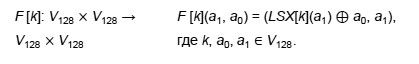


Раундовое преобразование можно изобразить следующим образом:

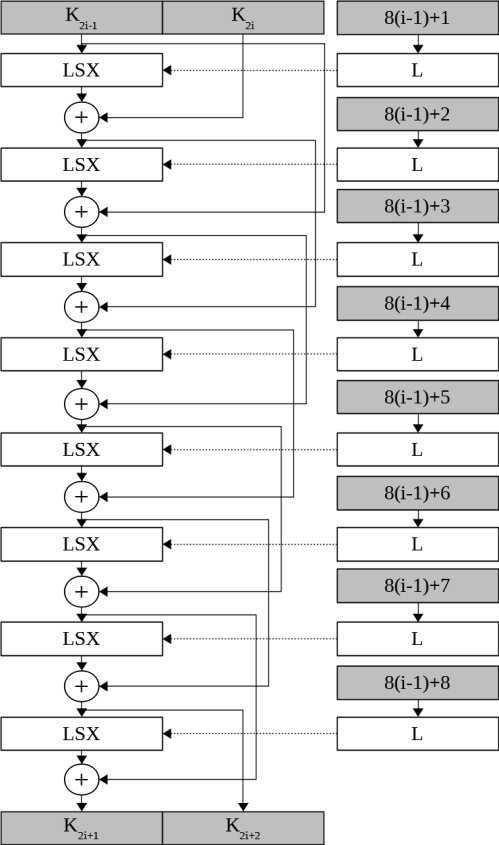


Выработка раундовых ключей

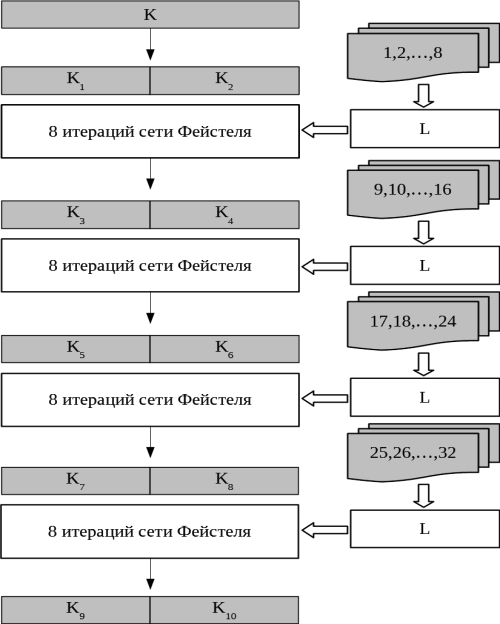
Рассмотрим процедуру генерации раундовых ключей из мастер-ключа. Первые два получаются разбиением мастер-ключа пополам. Далее для выработки очередной пары раундовых ключей используется 8 итераций сети Фейстеля, где, в свою очередь, в качестве раундовых ключей используется счетчиковая последовательность, прошедшая через линейное преобразование алгоритма:



Раунд ключевой развертки можно представить следующим образом:



а всю процедуру выработки раундовых ключей так:

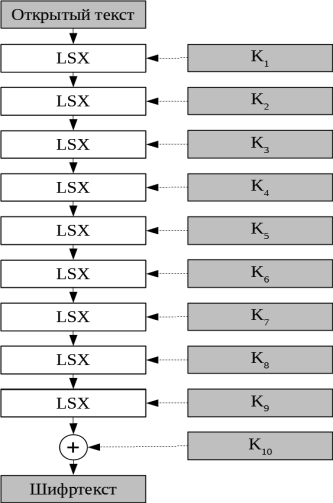


Шифрование и расшифрование

В результате шифрование одного 128-битного входного блока описывается следующим уравнением:



А в виде блок-схемы может быть представлено так:



Расшифрование реализуется обращением базовых преобразований и применением их в обратном порядке:



## Литература

1. Бутакова Н.Г., Семененко В.А., Федоров Н.В. Криптографическая защита информации: учебное пособие для вузов. – М.: Изд-во МГИУ, 2011. – 316 с.
2. Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В. Основы криптографии: Учебное пособие. – М.: Гелиос АРВ, 2002. – 480 с.
3. Зубов А.Ю. Совершенные шифры. – М.: Гелиос АРВ, 2003. – 162 с.
4. Словарь криптографических терминов / Под редакцией Б.А.Погорелова и

В.Н.Сачкова. – М.: МЦНМО, 2006.

1. Введение в криптографию / Под общей редакцией В.В.Ященко. – 3-е изд., доп. – М.: МЦНМО: «ЧеРо», 2000. – 288 с.
2. Нечаев В.И. Элементы криптографии (Основы теории защиты информации): Учеб. пособие для ун-тов и пед. Вузов / Под ред. В.А.Садовничего – М.:

Высш.шк., 1999, – 109 с.

1. Виноградов И.М. Основы теории чисел. – 10-е изд., стер. – СПб.: Издательство «Лань», 2004. – 176 с.
2. Жданов О.Н., Чалкин В.А. Эллиптические кривые: Основы теории и криптографические приложения. – М.: Книжный дом «ЛИБРОКОМ», 2013. –

200 с.

9. Масленников М. Практическая криптография. – СПб.: БХВ-Петербург,

2003. – 464 с.

1. Коутинхо С.. Введение в теорию чисел. Алгоритм RSA. – М.: Постмаркет, 2001. – 328 с.
2. Бабаш А.В., Шанкин Г.П. История криптографии. Ч.1. – М.: Гелиос АРВ, 2002. – 240 с.
3. Грушо А.А. Тимонина Е.Е. Теоретические основы защиты информации. – М.: Издательство Агентства «Яхтсмен», 1996. – 192 с.
4. Бутакова Н.Г.Основы защиты конфиденциальных документов: Учебное пособие. – М.:МОСУ, 2005. – 246 с.
5. Семененко В.А., Бутакова Н.Г. Основы защиты информации в компьютерных системах: Учебное пособие. – М.: МОСУ, 2004. – 238 с.
6. Закон «Об информации, информационных технологиях и защите информации» от 27 июля 2006 года N 149-ФЗ.
7. Федеральный Закон «Об электронной цифровой подписи» от 10 января 2002 г. N 1-ФЗ.
8. ГОСТ 28147-89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. – М., Госстандарт, 1990.
9. ГОСТ Р 34.10-2001. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи. – М., Госстандарт, 2001.
10. ГОСТ Р 34.11-94. Функция хеширования. – М., Госстандарт, 1994.
11. ГОСТ Р 34.10-2012. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи. – М., Стандартинформ, 2012.
12. ГОСТ Р 34.11-2012. Информационная технология. Криптографическая защита информации. Функция хэширования. – М., Стандартинформ, 2012.
13. ГОСТ Р 34.12-2015. Информационная технология. Криптографическая защита информации. Блочные шифры. – М., Стандартинформ, 2015.
14. ГОСТ Р 34.13-2015. Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров. – М., Стандартинформ, 2015.
15. Соболева Т.А. История шифровального дела в России. – М.: ОЛМА-ПРЕССОбразование, 2002.
16. КомпьютерПресс: Безопасность, 2003, №3 (март).
17. Романец Ю.В., Тимофеев П.А., Шаньгин В.Ф. Защита информации в компьютерных системах и сетях. – М.: Радио и связь, 2001. – 376 с.
18. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – М.: ТРИУМФ, 2002 – 816 с.
19. Баричев С.Г., Гончаров В.В., Серов Р.Е. Основы современной криптографии. – М.: Горячая линия – Телеком, 2001.
20. Анохин М.И., Варновский Н.П., Сидельников В.М., Ященко В.В. «Криптография в банковском деле». – М.: Изд-во МИФИ, 1997.
21. Крысин А.В. Информационная безопасность: Практическое руководство. – М.: «СПАРРК», К. «ВЕК+», 2003. – 320 с.
22. Жельников В. Криптография от папируса до компьютера. – М., 1996.
23. Теория и практика обеспечения информационной безопасности. Под редакцией П.Д. Зегжды. – М.: Яхтсмен, 1996.
24. Кочер П. Временной анализ реализаций Диффи–Хеллмана, RSA, DSS и других систем.
25. Mark W. Eichin, Jon A. Rochils. With Microscope and Tweezers: An Analysis of the Internet virus of November 1988.
26. Шнайер Б., Мюдж П. Криптоанализ протокола PPTP от Microsoft.
27. Eli Biham, Lars R. Knudsen. Cryptanalysis of the ANSI X9.52 CBCM Mode. Proceedings of Eurocrypt'98.
28. КомпьютерПресс 3'2002.
29. Винокуров А.Ю. ГОСТ не прост, а очень прост. – М.: Монитор, 1995. – N1.
30. Винокуров А.Ю. Еще раз про ГОСТ. – М.: Монитор, 1995. – N5.
31. Смид М.Э., Бранстед Д.К. Стандарт шифрования данных: прошлое и будущее. /пер. с англ. – М.: Мир, ТИИЭР, 1988. –т.76. –N5.
32. Березин Б.В., Дорошкевич П.В. Цифровая подпись на основе традиционной криптографии//Защита информации, вып. 2. – М.: МП «Ирбис-II», 1992.
33. W.Diffie,M.E.Hellman. New Directions in cryptography// IEEE Trans. Inform. Theory, IT-22, vol 6 (Nov. 1976), pp. 644-654. 43. Диффи У. Первые десять лет криптографии с открытым ключом /пер. с англ. – М.: Мир, ТИИЭР, 1988. – т.76. –N5.
34. Водолазкий В. Стандарт шифрования ДЕС, Монитор, 03-04 1992 г.
35. Воробьев С. Защита информации в персональных ЗВМ. – М.: Мир, 1993.
36. Ковалевский В. Криптографические методы. – Компьютер Пресс 05.93.
37. Мафтик С. Механизмы защиты в сетях ЭВМ. – М.: Мир, 1993. – 216 с.
38. Фролов Г. Тайна тайнописи. – М., 1992.
39. Зелинский Ф. Сказочная древность Эллады. – М., 1993.
40. Жельников В. Криптография от папируса до компьютера. М., 1996.
41. Горбовский А. Загадки древнейшей истории. – М., 1971.
42. Пазин С.В. Основы защиты информации в компьютерных системах. – М.: ТВП/ОпиПМ, 2003.
43. Стрельцов А.А. Обеспечение информационной безопасности России. Теоретические и методические основы / Под ред. В.А. Садовничего и В.П. Шерстюка. – М.: МЦНМО, 2002.
44. Информационная безопасность и защита информации. Учебное пособие /Степанов Е.А. и др. – М.: Инфра-М, 2002.
45. Информационная безопасность. Учебное пособие. Ярочкин В.И. – М.: Международные отношения, 2000.
46. Малюк А.А., Пазизин С.В., Погожин Н.С. Введение в защиту информации в автоматизированных системах: Учебное пособие. – М.: Горячая линия – Телеком, 2001.
47. Завгородний В.И. Комплексная защита информации в компьютерных системах: Учебное пособие. – М.: Логос, 2001.
48. Зегжда Д.П., Ивашко А.М. Основы безопасности информационных систем. – М.: Горячая линия – Телеком, 2000.
49. Девянин П.М., Михальский Д.И., Правиков Д.И., Щербаков А.Ю. Теоретические основы компьютерной безопасности: Учеб. пособие для вузов. – М.: Радио и связь, 2000.
50. Артехин Б.В. Стеганография // Защита информации. Конфидент. – 1996. – №4.
51. Барсуков В.С. Обеспечение информационной безопасности. – М.: ТЭК, 1996.
52. Галатенко В.. Трифаленков И. Информационная безопасность в Интранет: концепции и решения // Jet Info. №23/24, 1996.
53. Герасименко В.А. Защита информации в автоматизированных системах обработки данных: В 2 кн. – М.: Энергоатомиздат, 1994.
54. Гостехкомиссия РФ. Руководящий документ. Защита информации. Специальные защитные знаки. – М.: Jet Info, 1997.
55. Гостехкомиссия РФ. Руководящий документ. Средства вычислительной техники. Межсетевые экраны. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации. – М.: Jet Info, 1997.
56. Гостехкомиссия РФ. Руководящий документ. Концепция защиты средств вычислительной техники и автоматизированных систем от несанкционированного доступа к информации. – М.: Jet Info, №2, 1996.
57. Груздев С.Л., Хачатурова О.Л. Электронные ключи «YASP» компании «ALADDIN». Новые технологии в маркетинге программного обеспечения //Защита информации. Конфидент. – 1996. – №6.
58. Защита программного обеспечения: Пер. с англ./ Д. Гроувер, Р. Сатер, Дж-Фипс и др.; Под ред. Д. Гроувера. – М.: Мир. 1992. – 286 с.
59. Алешин И. и др. Информационно-безопасные системы. Анализ проблемы: Учеб. пособие / Под ред. В.Н. Козлова – СПб.: Издательство С.Петербургского гос. техн. университета. 1996. – 69 с.
60. Лебедев А.Н. Криптография с «открытым ключом» и возможности его практического применения // Защита информации. Конфидент. – 1992. – №2.
61. Медведовский И.Д., Семьянов П.В., Платонов В.В. Атака через INTERNET /Под научной редакцией проф. П.Д. Зегжды – СПб.: Мир и семья, 1997. –

296 с. 72. Мельников В.В. Защита информации в компьютерных системах. – М.: Финансы и статистика; Электронинформ, 1997. – 368 с.

1. Михайлов С.Ф., Петров В.А., Тимофеев Ю.А. Информационная безопасность. Защита информации в автоматизированных системах. Основные концепции: Учебное пособие. – М.: МИФИ, 1995. – 112 с.
2. Протоколы и методы управления в сетях передачи данных / Пер. с англ.: Под ред. Ф.Ф. Куо. – М.: Радио и связь, 1985.
3. Стенг Д.И., Мун С. Секреты безопасности сетей. – К.: Диалектика, 1996.
4. Хоффман Л.Дж. Современные методы защиты информации / Пер. с англ. – М.: Сов. радио, 1980.
5. Шеннон К. Работы по теории информации и кибернетике. – М.: ИИЛ, 1963. – 830 с.
6. Шрейдер Ю.А. О семантических аспектах теории информации. Информация и кибернетика. – М.: Сов. радио, 1967.
7. Хоффман Л. Дж. Современные методы защиты информации, М.: Сов..
8. Оре О. Приглашение в теорию чисел. – Пер. с англ. Изд. 2-е, стер. – М.: Едиториал УРСС, 2003. – 128 с.
9. Digital signature standart (DSS). FIPS PUB 190, 1994, MAY 19.
10. Massey J. Cryptography: Fundamentals and Applications. Advanced Technology Seminars, 1994.
11. Menzes A., Oorschot P., Vanstone S. Hadbook of Applied Cryptography, CRC Press, 1997.
12. Meyer C., Matyas S. Cryptography: a New Dimension in Computer Data Security. John Wiley & Sons, 1982.
13. Salomaa A. Public-key cryptography, Springer-Verlas, Berlin. Heidelbers New York, London, Paris, Tokyo, Hong Kong, Barselona, 1990.
14. Schneier B. Applied Cryptography Second Edition: protocols, algorithms and source code in C. John Wiley & Sons Inc., 1996.
15. Амамия М., Танака Ю. Архитектура ЭВМ и искусственный интеллект. М.: Мир, 1993.
16. Гилл А. Линейные последовательные машины. Анализ, синтез и применение. М.: Наука, 1974.
17. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М., 1982.
18. Колчин В.Ф., Севастьянов Б.А., Чистяков В.П. Случайные размещения.

М.: Наука, 1976.

1. Рейнгольд Э., Нивергельт Ю., Део Н. Комбинаторные алгоритмы. Теория и практика. М.: Мир, 1980.
2. Biham E., Shamir A. Differential Cryptanalysis of the Data Encryption Standard/ Springer- Verlag, 1993.
3. Biham E. Cryptanalysis of Multiple Modes of Operation/ Proc. ASIACRYPT’94, Springer- Verlag, 1994, с. 278–291.
4. Biham E., Knudsen L. Cryptanalysis of the ANSI X.9.52 CBCM Mode /Technion – Computer Science Department – Technical Report CS 0928, 1998.
5. Golic J. Intrinsic Statistical Weakness of Keystream generators //Advances in Cryptology – ASIACRYPT’94, Lecture Notes in Computer Science, v. 917, Springer- Verlag, 1995, pp. 91-103.
6. Gollmann D., Chambers W. Clock Controlled Shift Registers: a Review //IEEE J. Scl. Ar. Commun., 1989, v. 7, N 4, pp. 525–533.
7. Matsui M. Linear Cryptanalysis Method for DES Cipher//Pre-Proceedings of Eurocrypt’93, 1993, pp. 112–133.
8. Meier W., Staffelbach O. Fast Correlation Attacks on Certain Stream Ciphers //

J. Cryptology, v. 1, N 3, 1989, pp. 159–176.

1. Rueppel R. Good Stream Ciphers are Hard to Design. ICCST, Zurich, 1989, pp. 163–173.
2. Schnorr C. On the Construction of Random Number Generators and Random Function Generators//Advances in Cryptology – Eurocrypt’88, 1988, pp. 225–232. 101. http://stavkombez.ru/method/PASOIB/html/content/lect\_2.html 102. http://project5gym6.narod.ru/5/22/p19aa1.html.
3. Черезов К. Средства криптографической защиты информации (СКЗИ) – http://www.itsec.ru/articles2/Oborandteh/sredstva\_kriptograf zasch inform sthash.cnLT6gvY.dpuf.
4. Винокуров А. Стандарты криптографической защиты информации России и США – http://www.bre.ru/security/18952.html
5. В ГОСТе сидел «Кузнечик» – https://habrahabr.ru/post/266359/
6. Шудрова К.Е. Криптографическая защита персональных данных – http://iecp.ru/articles/item/392140.
7. Защита информации в персональных компьютерах – http://bezopasnik.org/article/2.htm.
8. Рутокен ЭЦП Flash – http://www.rutoken.ru/
9. Microsoft TechNet. – http://technet.microsoft.com/ru-ru/library/
10. Безмалый Владимир. Эволюция шифрования в документах Microsoft Office. Август 2012. – http://www.microsoft.com/ru-ru/business/smb/blog/postview.aspx?id=175.
11. PGP Whole Disk Encryption software documentation. – http://www.symbuy.ru/symantec-drive-encryption.
12. TrueCrypt software documentation. – http://truecrypt.org.ua/docs. 113. WinZip software documentation.
13. Средство криптографической защиты информации Crypton Disk. Фирма «АНКАД» – http://ancud.ru/crdisk.html.
14. Secret Disk 4 – система защиты конфиденциальной информации. – http://www.aladdin-rd.ru/catalog/secret\_disk/personal/.
15. Комплекс криптографических решений для защиты данных Zecurion

Zserver. – http://www.zecurion.ru/products/zserver/.

1. Средство обеспечения защиты конфиденциальной информации КриптоПро EFS. – https://www.cryptopro.ru/products/other/efs
2. Принципы аутентификации по протоколу Kerberos. – http://itband.ru/2010/12/kerberos1/
3. Доступ повышенной проходимости. – https://habrahabr.ru/post/5401/
4. Adaptec maxCrypto – превосходное шифрование данных в режиме реального времени со скоростью передачи по каналу связи https://www.adaptec.com/nr/pdfs/maxcrypto\_techbrief\_ru.pdf.
5. Чмора А.П. Современная прикладная криптография. 2-е изд., стер. – М.: Гелиос АРВ, 2002. – 256 с.: ил.
6. Бабаш А.В. Криптографические и теоретико-автоматные аспекты современной защиты информации. Том 1. – М.: Изд. центр ЕАОИ, 2009. – 414 с. 123. Е.С. Селмер. Монография: «Линейная рекурсия в конечном поле». – Университет Бергена, Норвегия, 1966.
7. Н. Зиерлер и Дж. Бриллхарт. О примитивных трехчленах по модулю 2, журнал Информация и Контроль, издание 13, №6 декабрь 1968, стр. 541–544.
8. К.Х. Мейер и У.Л. Тучман. Псевдослучайные коды могут быть взломаны, журнал «Электроник Дизайн», № 23, ноябрь 1972.
9. Дж.Л. Массей. Обобщение регистров сдвига и дешифровка кода БозеЧоудхури-Хоквингема, IEEE Transactions on Information Theory, v. IT-15, n. 1, январь 1969, стр. 122–127.
10. С.У. Голомб. Последовательности регистров сдвига, Сан-Франциско, Голден-Дей, 1967 (переиздано Аигеан Парк Пресс, 1982).