МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Пятигорский институт (филиал) СКФУ
Колледж Пятигорского института (филиал) СКФУ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

ПМ.05 ВЕБ ТЕХНОЛОГИИ И ЗАЩИТА ИНФОРМАЦИИ

МДК.05.01 ПЕРСОНАЛЬНАЯ КИБЕРБЕЗОПАСНОСТЬ

Специальность СПО
09.02.01Компьютерные системы и комплексы
Квалификация специалист по компьютерным системам

Пятигорск, 2024

Методические указания для практических занятий по дисциплине МДК.05.01 Персональная кибербезопасность составлены в соответствии с требованиями ФГОС СПО. Предназначены для студентов, обучающихся по специальности 09.02.01 Компьютерные системы и комплексы.

Пояснительная записка

Данные методические указания предназначены для закрепления теоретических знаний и приобретения необходимых практических навыков и умений по программе дисциплины «Персональная кибербезопасность» для специальности СПО 09.02.01 Компьютерные системы и комплексы.

В результате освоения учебной дисциплины обучающийся должен

знать:

- регламенты, процедуры, технические условия и нормативы;
- определения кибербезопасности и кибератак;
- требования к криптографическим системам защиты информации;
- алгоритмы шифрования;
- методы криптоанализа;
- классификацию вирусов и антивирусных программ;
- программы для защиты информации;

уметь:

- выполнять требования нормативно-технической документации;
- применять знания о кибербезопасности в решении поставленных задач;
- защищать личную информацию;
- создавать надежные пароли;
- устранять нарушения кибербезопасности;

иметь практический опыт:

- применения нормативно-технической документации;
- создания защищенных резервных копий данных;
- использования методов криптографии и алгоритмов шифрования при передачи конфиденциальной информации;
- установки и проверки устройств с помощью антивирусных программ и утилит;
- передачи конфиденциальной информации по защищенным каналам.

Практическое занятие №1. Шифры перестановки

Цель: изучить способы шифрования данных различными перестановками.

Ход работы:

Необходимо ознакомиться с теоретической частью и зашифровать свою фамилию (для первых двух шифров) или фамилию и имя (для остальных) с помощью следующих шифров:

- простой одинарной перестановки;
- блочной одинарной перестановки;
- табличной маршрутной перестановки;
- вертикальной перестановки;
- поворотной решетки;
- магический квадрат (размер квадрата - 4x4);
- двойной перестановки.

При выполнении задания необходимо привести исходное сообщение (фамилию или фамилию и имя), таблицы, ключевые слова (выбираются произвольно), маршруты вписывания и выписывания, повороты решетки и зашифрованное сообщение.

Дополнительное задание. Зашифровать свою фамилию и имя с помощью шифров:

- шифра «Перекресток»;
- шифры с использованием треугольника.

Основы шифрования

Перестановка представляет собой способ шифрования, при котором для получения шифрограммы символы исходного сообщения меняют местами. Типичным примером перестановки являются анаграммы, ставшие популярными в XVII в. **Анаграмма** (греч. ανα - «снова» и γράμμα - «запись») - литературный приём, состоящий в перестановке букв или звуков определённого слова (или словосочетания), что в результате даёт другое слово или словосочетание. Например: апельсин - спаниель, полковник - клоповник, горилка - рогалик, лепесток - телескоп.

Доподлинно не известно, когда появился шифр перестановки, но вполне возможно, что писцы в древности переставляли буквы в имени своего царя ради того, чтобы скрыть его подлинное имя или в ритуальных целях.

Все шифры перестановки делятся на два **подкласса**:

- шифры одинарной (простой) перестановки. При шифровании символы перемещаются с исходных позиций в новые один раз;
- шифры множественной (сложной) перестановки. При шифровании символы перемещаются с исходных позиций в новые несколько раз.

Шифры одинарной перестановки

В общем случае для данного класса шифров при шифровании и дешифровании используется таблица перестановок.

1	2	3	…	n
I_1	I_2	I_3	…	I_n

Рис.1. Таблица перестановок

В первой строке данной таблицы указывается позиция символа в исходном сообщении, а во второй – его позиция в шифрограмме. Таким образом, максимальное количество ключей для шифров перестановки равно **n!**, где **n** – длина сообщения.

С увеличением числа **n** значение **n!** растет очень быстро (1! = 1, 5! = 120, 10! = 3628800, 15! = 1307674368000). При больших **n** для приближенного вычисления **n!** можно воспользоваться формулой Стирлинга

$$n! \approx \sqrt{2\pi n} * \left(\frac{n}{e}\right)^n \quad (1)$$

Шифр простой одинарной перестановки. Для шифрования и дешифрования используется таблица перестановок, аналогичная показанной на рис.2.

1	2	3	4	5	6	7
2	4	1	7	6	5	3

Рис.2. Таблица перестановок

Например, если для шифрования исходного сообщения «АБРАМОВ» использовать таблицу, представленную на рис.2, то шифрограммой будет «РАВБОМА». Для использования на практике такой шифр не удобен, так как при больших значениях **n** приходится работать с длинными таблицами и для сообщений разной длины необходимо иметь свою таблицу перестановок.

Шифр блочной одинарной перестановки. При использовании этого шифра задается таблица перестановки блока символов, которая последовательно применяется до тех пор, пока исходное сообщение не закончится. Если исходное сообщение не кратно размеру блока, тогда оно при шифровании дополняется произвольными символами.

1	2	3
2	3	1

Рис.3. Таблица перестановок

Для примера выберем размер блока, равный 3, и примем таблицу перестановок, показанную на рис.4.3. Дополним исходное сообщение «АБРАМОВ» буквами **Ь** и **Э**, чтобы его длина была кратна 3. В результате шифрования получим «РАБОАМЭВЬ».

Количество ключей для данного шифра при фиксированном размере блока равно **m!**, где **m** – размер блока.

Шифры маршрутной перестановки. Широкое распространение получили шифры перестановки, использующие некоторую геометрическую фигуру (плоскую или объемную). Преобразования состоят в том, что в фигуру исходный текст вписывается по ходу одного маршрута, а выписывается по-другому.

Шифр табличной маршрутной перестановки. Наибольшее распространение получили шифры маршрутной перестановки, основанные на таблицах. При шифровании в такую таблицу вписывают исходное сообщение по определенному маршруту, а выписывают (получают шифрограмму) - по-другому. Для данного шифра маршруты вписывания и выписывания, а также размеры таблицы являются ключом.

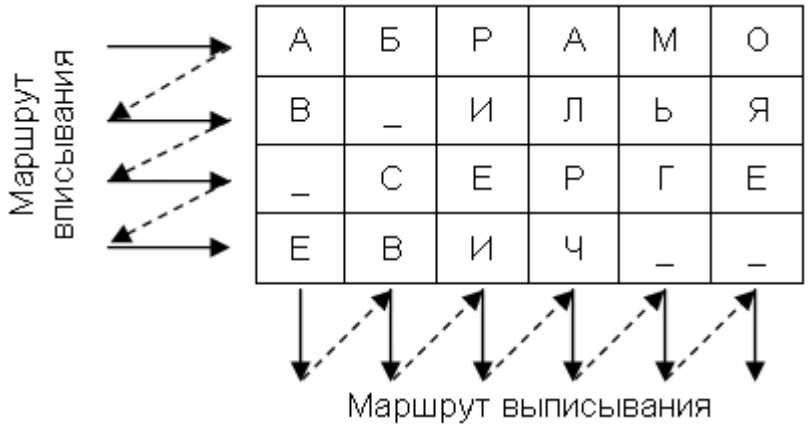


Рис.4. Пример использования шифра маршрутной перестановки

Например, исходное сообщения «АБРАМОВ ИЛЬЯ СЕРГЕЕВИЧ» вписывается в прямоугольную таблицу размерами 4х6, маршрут вписывания – слева-направо сверху-вниз, маршрут выписывания – сверху-вниз слева-направо. Шифрограмма в этом случае выглядит «АВ_ЕБ_СВРИЕИАЛР ЧМЬГ_ОЯЕ_».

Шифр вертикальной перестановки. Является разновидностью предыдущего шифра. К особенностям шифра можно отнести следующие:

- количество столбцов в таблице фиксируется и определяется длиной ключа;
- маршрут вписывания - строго слева-направо сверху-вниз;
- шифрограмма выписывается по столбцам в соответствии с их нумерацией (ключом).

Ключ	Д	Я	Д	И	Н	А
	2	6	3	4	5	1
Текст	А	Б	Р	А	М	О
	В	_	И	Л	Ь	Я
	_	С	Е	Р	Г	Е
	Е	В	И	Ч	_	_

Рис.5. Пример использования шифра вертикальной перестановки

В качестве ключа можно использовать слово или фразу. Тогда порядок выписывания столбцов соответствует алфавитному порядку букв в ключе.

Например, если ключевым словом будет «ДЯДИНА», то присутствующая в нем буква **А** получает номер 1, **Д** – 2 и т.д. Если какая-то буква входит в слово несколько раз, то ее появления нумеруются последовательно слева направо. В примере первая буква **Д** получает номер 2, вторая **Д** – 3.

При шифровании сообщения «АБРАМОВ ИЛЬЯ СЕРГЕЕВИЧ» результат будет «ОЯЕ_АВ_ЕРИЕИАЛРЧМЫГ_Б_СВ».

Шифр «Перекресток». Для перемешивания букв могут использоваться фигуры специального вида. Один из таких способов носит название «перекресток». В приведенном ниже примере рисуют крестообразные фигуры в количестве, достаточном, чтобы разместить в них все буквы сообщения. Открытый текст записывают вокруг этих фигур заранее оговоренным способом - в нашем случае по часовой стрелке. Таким образом, сообщение «АБРАМОВ ИЛЬЯ СЕРГЕЕВИЧ» может выглядеть следующим образом:

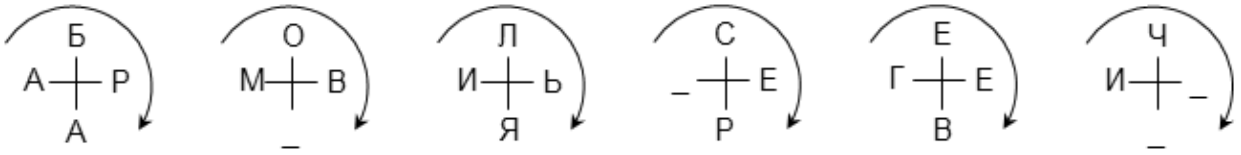


Рис.6. Пример размещения открытого текста в шифре «Перекресток»

Буквы берутся построчно. Вначале берется оговоренное количество букв (N) из первой строки, затем удвоенное количество букв (2N) из второй и снова N букв из третьей строки. Например, при N = 3 шифрограмма будет выглядеть «БОЛАРМВИЬА_ЯСЕЧ_ЕГЕИ_РВ_».

Шифры с использованием треугольников и трапеций. Помочь выполнить перестановки могут как треугольники, так и трапеции. Открытый текст вписывается в эти фигуры в соответствии с количеством слов и формой выбранной фигуры, которая может быть растянута или сжата, чтобы в ней поместилось сообщение. Для первой фигуры, треугольника, открытый текст записывается построчно от вершины до основания.

А

Б Р

А М О

В _ И Л

Ь Я _ С Е

Р Г Е Е В И

Д Я Д И Н А Д Я Д И Н

2 10 3 6 8 1 4 11 5 7 9

Рис.7. Пример использования шифра перестановки при вписывании в треугольник

Ниже записывается ключевое слово. Поскольку основание широкое, ключевое слово повторяется. Буквы строки с ключевым словом нумеруются последовательно согласно их алфавитному порядку. Зашифрованное сообщение выписывается по столбцам согласно выполненной нумерации. Таким образом, для открытого текста «АБРАМОВ ИЛЬЯ СЕРГЕЕВИ» и ключевого слова «ДЯДИНА» шифрограмма будет выглядеть «АМ_РВГРИЕЛВАЯЕБ_ЕИЬРС».

Шифр «Поворотная решетка». В 1550 г. итальянский математик Джероламо Кардано, состоящий на службе у папы Римского, в книге «О тонкостях» предложил новую технику шифрования - **решётку Кардано**.

Изначально решетка Кардано представляла собой трафарет с прорезанными в нем отверстиями. В этих отверстиях на листе бумаги, который клали под решетку, записывались буквы, слоги и слова сообщения. Далее трафарет снимался, и свободное пространство заполнялось более или менее осмысленным текстом для маскировки секретного послания. Такой метод сокрытия информации относится к стеганографии.

Позднее был предложен шифр «поворотная решетка» или, как его еще называют, «решетка для вьющихся растений», поскольку она напоминала отверстия в деревянных решетках садовых строений. Этот шифр считают первым **транспозиционным** (геометрическим) шифром.

Несмотря на то, что между изначальным предложением Кардано и шифром «поворотная решетка» большая разница, методы сокрытия информации, основанные на использовании трафаретов, принято называть «решетками Кардано».

Для шифрования и дешифрования с помощью данного шифра изготовляется прямоугольный трафарет с четным количеством строк и столбцов. В трафарете вырезаются клетки таким образом, чтобы при наложении его на таблицу того же размера четырьмя возможными способами, его вырезы полностью покрывали все ячейки таблицы ровно по одному разу.

При шифровании трафарет накладывается на таблицу. В видимые ячейки таблицы выписываются буквы исходного текста слева-направо сверху-вниз. Далее трафарет поворачивается и вписывается следующая часть букв. Эта операция повторяется еще два раза. Шифрограмму выписывают из итоговой таблицы по определенному маршруту.

Таким образом, ключом при шифровании является трафарет, порядок его поворотов и маршрут выписывания.

Пример шифрования сообщения «АБРАМОВ+ДЯДИНА» показан на рис.8. Результат шифрования – «АДВ_МНРДБЯ+_ОААИ».

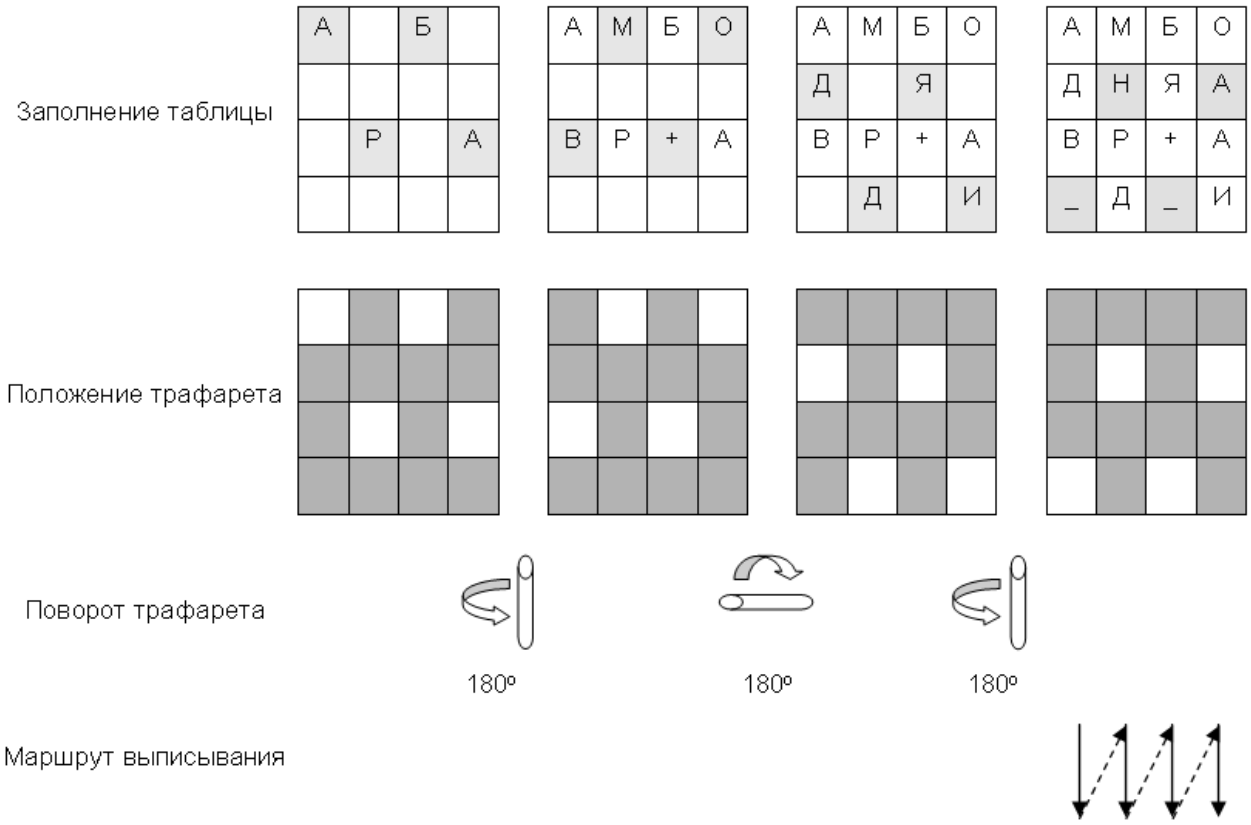


Рис.8. Пример использования шифра «поворотная решетка»

Данный метод шифрования применялся нидерландскими правителями для секретных посланий в 1740-х гг. Он также использовался в армии кайзера Вильгельма в Первую мировую войну. Для шифрования немцы использовали решетки разных размеров, которым французские криптоаналитики дали собственные кодовые имена: Анна (25 букв), Берта (36 букв), Дора (64 буквы) и Эмиль (81 буква). Однако использовались решетки очень недолго (всего четыре месяца) к огромному разочарованию французов, которые только-только начали подбирать к ним ключи.

Магические квадраты. Магическими [нормальными] квадратами называются квадратные таблицы со вписанными в их ячейки последовательными натуральными числами начиная с 1, которые в сумме по каждому столбцу, каждой строке и главным диагоналям дают одно и то же число.

Впервые эти квадраты появились в Китае, где им и была приписана некоторая «магическая сила». По преданию, описанному в одной из пяти канонических книг Древнего Китая - Шу-Цзин (Книге записанных преданий), в 2200 году до н.э. из реки Ло вышла огромная черепаха (по другой версии - дракон), символ вечности. На ее панцире были видны пятна, образовывавшие удивительный рисунок.

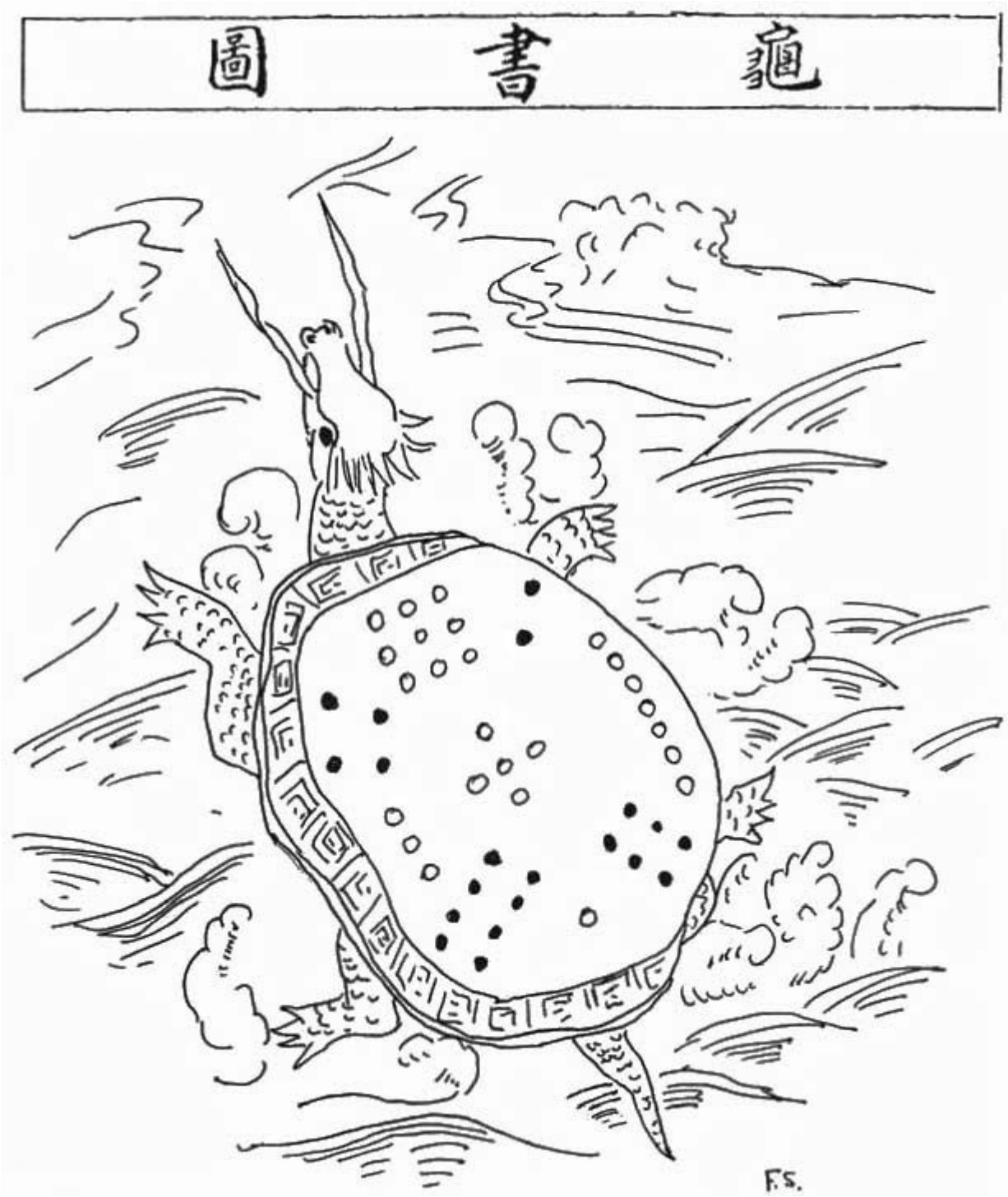


Рис.9. Магический квадрат Ло Шу

Когда черепаха вышла из воды, высыхали лужи после недавнего ливня. Великий Юй взял эту черепаху и рассмотрел странный узор на ее панцире. Этот узор вдохновил его на создание трактата под названием «Хун Фань» («Великий план»), в котором говорилось о физике, астрологии, предсказаниях, морали, политике и религии.

Магические квадраты широко применялись для передачи секретной информации. При шифровании исходное сообщение вписывалось в квадрат по приведенной в них нумерации, после чего шифрограмма выписывалась по строкам. Количество возможных магических квадратов (ключей) быстро

возрастает с увеличением их размера. Так, существует лишь один магический квадрат размером 3x3, если не принимать во внимание его повороты. Магических квадратов 4x4 насчитывается уже 880, а число магических квадратов размером 5x5 около 250000. Поэтому магические квадраты больших размеров могли быть хорошей основой для надежной системы шифрования того времени, потому что ручной перебор всех вариантов ключа для этого шифра был немыслим.

Рассмотрим квадрат размером 4x4. В него вписываются числа от 1 до 16. Его магия состоит в том, что сумма чисел по строкам, столбцам и полным диагоналям равняется одному и тому же числу - 34.

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

Рис.10. Магический квадрат 4x4

Шифрование по магическому квадрату производилось следующим образом. Например, требуется зашифровать фразу: «АБРАМОВ+ДЯДИНА..». Буквы этой фразы вписываются последовательно в квадрат согласно записанным в ячейках числам. В пустые клетки ставится точка или любая буква.

16 .	3 Р	2 Б	13 Н
5 М	10 Я	11 Д	8 +
9 Д	6 О	7 В	12 И
4 А	15 .	14 А	1 А

Рис.11. Пример шифрования с помощью магического квадрата

После этого шифрованный текст записывается в строку (считывание производится слева-направо сверху-вниз, построчно) – «.РБНМЯД+ДОВИА.АА».

Шифры множественной перестановки

В данном подклассе шифров используется идея повторного шифрования уже зашифрованного сообщения или многократной перестановки символов исходного сообщения перед попаданием в итоговую шифрограмму.

Шифр двойной перестановки. В таблицу по определенному маршруту записывается текст сообщения, затем переставляются столбцы, а потом переставляются строки. Шифрограмма выписывается по определенному маршруту.

Пример шифрования сообщения «АБРАМОВ+ДЯДИНА» показан на следующем рисунке. Результат шифрования – «ОАБЯ+_АИВ_РДМНАД».

Рис.12. Пример использования шифра двойной перестановки

Ключом к шифру являются размеры таблицы, маршруты вписывания и выписывания, а также порядки перестановки столбцов и строк. Если маршруты являются фиксированными величинами, то количество ключей равно **n! *m!**, **n** и **m** – количество столбцов и строк в таблице.

Несмотря на многоступенчатую процедуру шифрования, включая двойную перестановку, данный шифр может быть эквивалентно заменен шифром простой одинарной перестановки. На следующем рисунке приведена таблица эквивалентных одинарных перестановок для примера шифрования, приведенного на рис. 12.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
А	Б	Р	А	М	О	В	+	Д	Я	Д	И	Н	А	_	_
15	3	11	7	13	1	9	5	16	4	12	8	14	2	10	6
О	А	Б	Я	+	_	А	И	В	_	Р	Д	М	Н	А	Д

Рис.13. Таблица эквивалентных одинарных перестановок

Аналогичную замену на шифр простой одинарной перестановки можно выполнить и для других шифров: табличная маршрутная перестановка, «поворотная решетка», «магический квадрат» и др.

Вопросы для самоконтроля

1. Основные понятия кибербезопасности.
2. Кибербезопасность и информационная безопасность.
3. Составляющие информационной безопасности.
4. Уровни критической инфраструктуры.

Практическое занятие №2. Шифры замены

Цель: изучить способы шифрования данных различными заменами.

Ход работы:

Необходимо ознакомиться с теоретической частью и зашифровать свою фамилию с помощью следующих шифров:

- шифра Цезаря;
- лозунгового шифра;
- полибианского квадрата;
- шифрующей системы Трисемуса;
- шифра Playfair;
- системы омофонов (допускается для каждой буквы алфавита привести всего по две шифрозамены, т.е. принять, что все буквы имеют одинаковую вероятность появления в текстах);
- шифра Виженера.

При выполнении задания необходимо привести исходное сообщение (фамилию), таблицу шифрозамен, ключ (если таблица шифрозамен не является ключом) и зашифрованное сообщение.

Дополнительное задание. Зашифровать свою фамилию с помощью шифров:

- шифра масонов;
- биграммного шифра Порты;
- шифра Хилла;
- вариантного шифра;
- шифра Тени;
- совмещенного шифра.

Основы шифрования

Сущность шифрования методом замены заключается в следующем. Пусть шифруются сообщения на русском языке и замене подлежит каждая буква этих сообщений. Тогда, букве **А** исходного алфавита сопоставляется некоторое множество символов (шифрозамен) $\mathbf{M_А}$, **Б** – $\mathbf{M_Б}$, **…**, **Я** – $\mathbf{M_Я}$. Шифрозамены выбираются таким образом, чтобы любые два множества ($\mathbf{M_I}$ и $\mathbf{M_J}$, $\mathbf{i \neq j}$) не содержали одинаковых элементов ($\mathbf{M_I \cap M_J = Ø}$).

Таблица, приведенная на рис.1, является ключом шифра замены. Зная ее, можно осуществить как шифрование, так и расшифрование.

А	Б	...	Я
$M_А$	$M_Б$	...	$M_Я$

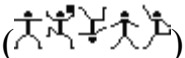
Рис.1. Таблица шифрозамен

При шифровании каждая буква **А** открытого сообщения заменяется любым символом из множества $\mathbf{M_А}$. Если в сообщении содержится несколько букв **А**, то каждая из них заменяется на любой символ из $\mathbf{M_А}$. За счет этого с помощью

одного ключа можно получить различные варианты шифрограммы для одного и того же открытого сообщения. Так как множества $\mathbf{M_А}$, $\mathbf{M_Б}$, ..., $\mathbf{M_Я}$ попарно не пересекаются, то по каждому символу шифрограммы можно однозначно определить, какому множеству он принадлежит, и, следовательно, какую букву открытого сообщения он заменяет. Поэтому расшифрование возможно и открытое сообщение определяется единственным образом.

Приведенное выше описание сущности шифров замены относится ко всем их разновидностям за исключением полиалфавитных шифров, в которых для зашифрования разных символов исходного алфавита могут использоваться одинаковые шифрозамены (т.е. $\mathbf{M_I \cap M_J \neq Ø}$, $\mathbf{i \neq j}$).

Метод замены часто реализуется многими пользователями при работе на компьютере. Если по забывчивости не переключить на клавиатуре набор символов с латиницы на кириллицу, то вместо букв русского алфавита при вводе текста будут печататься буквы латинского алфавита («шифрозамены»).

Для записи исходных и зашифрованных сообщений используются строго определенные алфавиты. Алфавиты для записи исходных и зашифрованных сообщений могут отличаться. Символы обоих алфавитов могут быть представлены буквами, их сочетаниями, числами, рисунками, звуками, жестами и т.п. В качестве примера можно привести пляшущих человечков из рассказа А. Конан Дойла () и рукопись рунического письма () из романа Ж. Верна «Путешествие к центру Земли».

Шифры замены по особенностям процедур преобразования сообщения можно разделить на следующие **подклассы** (типы, разновидности).

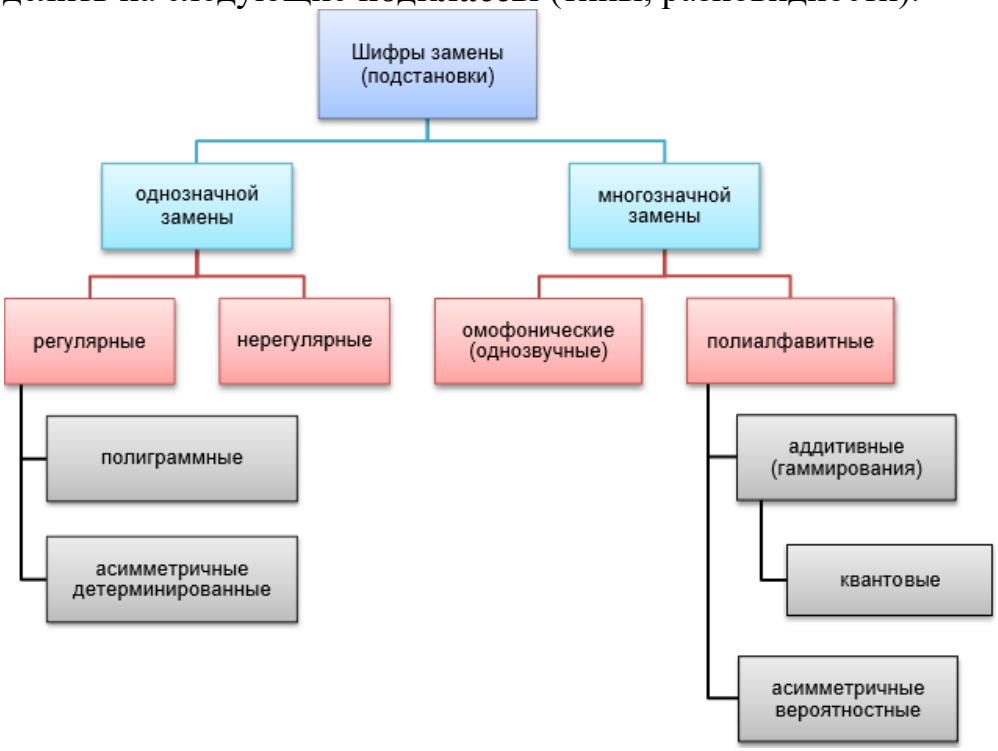


Рис.2. Классификация шифров замены

На рисунке серым фоном представлены подгуппы шифров, которые не образуют полную вышележащую группу шифров, но получившие широкое применение на практике и описаны в этой или отдельных лекциях.

В следующий таблице приведена краткая характеристика типов шифров замены.

Таблица 1. Типы шифров замены

Тип шифра			Краткая характеристика	Примеры шифров
однозначной замены (моноалфавитные, простые подстановочные)			Количество шифрозамен **M** для каждого символа или блока символов исходного алфавита равно 1 ($\|\mathbf{M}_i\| = 1$ для одного i-го символа или блока символов).	
однозначной замены	регулярные		Шифрозамены состоят из одинакового количества символов или отделяются друг от друга разделителем (пробелом, точкой, тире и т.п.).	шифр Цезаря, лозунговый шифр, тюремный шифр
	регулярные	полиграммные	Шифрозамене соответствует блок символов исходного алфавита ($\|\mathbf{M}_i\| = 1$ для одного i-ого блока символов).	биграммный шифр Порты, шифр Хилла
		асимметричные детерминированные	При зашифровании одного и того же открытого сообщения одним	RSA, шифр на основе задачи об укладке

			и тем же открытым ключом всегда будет получаться одна и та же шифрограмма. Т.е. для заданного открытого ключа один и тот же символ (блок символов) открытого сообщения всегда будет представляться одной и той же шифрозаменой.	ранца
	нерегулярные		Шифрозамены состоят из разного количество символов, записываемых без разделителей.	совмещенный шифр
многозначной замены			Количество шифрозамен **M** для отдельных символов или блока символов исходного алфавита больше 1 ($\|\mathbf{M}_i\| \geq 1$ для одного i-го символа или блока символов).	
многозначной замены	омофонические (однозвучные)		Шифрозамены для разных символов или блоков символов исходного алфавита не повторяются ($\mathbf{M}_I \cap \mathbf{M}_J = \mathbf{Ø}$ для двух разных i-го и	система омофонов, книжный шифр

			j-го символов или блоков символов).	
	полиалфавитные (многоалфавитные)		Исходному алфавиту для записи открытых сообщений соответствует несколько алфавитов шифрозамен. Выбор варианта алфавита шифрозамен для зашифрования отдельного символа или блока символов зависит от особенностей шифра. Одна и та же шифрозамена может использоваться для разных символов или блоков символов исходного алфавита ($\mathbf{M_I} \cap \mathbf{M_J} \neq \mathbf{Ø}$ для двух разных i-го и j-го символов или блоков символов).	диск Альберти, система Виженера
	полиалфавитные	аддитивные (гаммирования)	При зашифровании символы исходного алфавита в открытом сообщении заменяются числами, к которым добавляются	шифрование сложением по модулю N, шифр Вернама

			числа секретной случайной числовой последовательности (гаммы), после чего берется остаток от деления по модулю (операция **mod**).	
		квантовые	Являются разновидностью шифров гаммирования, где в качестве носителей информации используются элементарные частицы (пучки элементарных частиц).	
		асимметричные вероятностные	При зашифровании одного и того же открытого сообщения одним и тем же открытым ключом могут получаться разные шифрограммы. Т.е. для заданного открытого ключа один и тот же символ (блок символов) открытого сообщения может представляться разными шифрозаменами. Это достигается	схема Эль-Гамаля, шифр на основе эллиптических кривых

			за счет использования случайной величины при зашифровании символа (блока символов), что эквивалентно переключению алфавитов шифрозамен.	

Регулярные шифры однозначной замены

Максимальное количество ключей для любого шифра этого типа не превышает **n!**, где **n** – количество символов в алфавите. При больших **n** для приближенного вычисления **n!** можно воспользоваться формулой Стирлинга.

Шифр Цезаря. Согласно описаниям историка Светония в книге «Жизнь двенадцати цезарей» данный шифр использовался Гаем Юлием Цезарем для секретной переписке со своими генералами (I век до н.э.). Применительно к русскому языку суть его состоит в следующем. Выписывается исходный алфавит (**А, Б, ..., Я**), затем под ним выписывается тот же алфавит, но с циклическим сдвигом на 3 буквы влево.

А	Б	В	Г	Д	Е	Ё	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
Г	Д	Е	Ё	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	А	Б	В

Рис.3. Таблица шифрозамен для шифра Цезаря

При зашифровке буква **А** заменяется буквой **Г**, **Б** - на **Д** и т. д. Так, например, исходное сообщение «АБРАМОВ» после шифрования будет выглядеть «ГДУГПСЕ». Получатель сообщения «ГДУГПСЕ» ищет эти буквы в нижней строке и по буквам над ними восстанавливает исходное сообщение «АБРАМОВ».

Ключом в шифре Цезаря является величина сдвига нижней строки алфавита. Количество ключей для всех модификаций данного шифра применительно к алфавиту русского языка равно 33. Существуют различные модификации шифра Цезаря, в частности атбаш и лозунговый шифр.

Атбаш. В Ветхом Завете существует несколько фрагментов из священных текстов, которые зашифрованы с помощью шифра замены, называемого атбаш. Этот шифр состоит в замене каждой буквы другой буквой, которая находится в алфавите на таком же расстоянии от конца алфавита, как оригинальная буква - от начала. Например, в русском алфавите буква **А** заменяется на **Я**, буква **Б** - на **Ю** и т.д. В оригинальном Ветхом Завете использовались буквы еврейского алфавита. Так, в книге пророка Иеремии (25:26) слово «Бабель» (Вавилон) зашифровано как «Шешах».

Лозунговый шифр. Для данного шифра построение таблицы шифрозамен основано на лозунге (ключе) – легко запоминаемом слове. Вторая строка таблицы шифрозамен заполняется сначала словом-лозунгом (причем повторяющиеся буквы отбрасываются), а затем остальными буквами, не вошедшими в слово-лозунг, в алфавитном порядке. Например, если выбрано слово-лозунг «ДЯДИНА», то таблица имеет следующий вид.

А	Б	В	Г	Д	Е	Ё	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
Д	Я	И	Н	А	Б	В	Г	Е	Ё	Ж	З	Й	К	Л	М	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю

Рис.4. Таблица шифрозамен для лозунгового шифра

При шифровании исходного сообщения «АБРАМОВ» по приведенному выше ключу шифрограмма будет выглядеть «ДЯПДКМИ».

В качестве лозунга рекомендуется выбирать фразу, в которой содержаться конечные буквы алфавита. В общем случае, количество вариантов нижней строки (применительно к русскому языку) составляет 33! ($\geq 10^{35}$).

Полибианский квадрат. Шифр изобретен греческим государственным деятелем, полководцем и историком Полибием (203-120 гг. до н.э.). Применительно к русскому алфавиту и индийским (арабским) цифрам суть шифрования заключалась в следующем. В квадрат 6х6 выписываются буквы (необязательно в алфавитном порядке).

	1	2	3	4	5	6
1	А	Б	В	Г	Д	Е
2	Ё	Ж	З	И	Й	К
3	Л	М	Н	О	П	Р
4	С	Т	У	Ф	Х	Ц
5	Ч	Ш	Щ	Ъ	Ы	Ь
6	Э	Ю	Я	-	-	-

Рис.5. Таблица шифрозамен для полибианского квадрата

Шифруемая буква заменяется на координаты квадрата (строка-столбец), в котором она записана. Например, если исходное сообщение «АБРАМОВ», то шифрограмма – «11 12 36 11 32 34 13». В Древней Греции сообщения передавались с помощью оптического телеграфа (с помощью факелов). Для каждой буквы сообщения вначале поднималось количество факелов, соответствующее номеру строки буквы, а затем номеру столбца.

Тюремный шифр. Эта звуковая разновидность полибианского квадрата была разработана заключенными. Система состояла из нескольких ударов, обозначающих строки и столбцы в таблице с буквами алфавита. Один удар, а потом еще два соответствовали строке 1 и столбцу 2, т.е. букве **Б**. Пауза служила

разделителем между строками и столбцами. Таким образом, зашифровать исходное сообщение «АБРАМОВ» можно следующим образом.

А	тук ___ тук
Б	тук ___ тук, тук
Р	тук, тук, тук ___ тук, тук, тук, тук, тук, тук
А	тук ___ тук
М	тук, тук, тук ___ тук, тук
О	тук, тук, тук ___ тук, тук, тук, тук
В	тук ___ тук, тук, тук

Рис.6. Пример использования тюремного шифра

Шифрующая система Трисемуса (Тритемия). В 1508 г. аббат из Германии Иоганн Трисемус написал печатную работу по криптологии под названием «Полиграфия». В этой книге он впервые систематически описал применение шифрующих таблиц, заполненных алфавитом в случайном порядке. Для получения такого шифра замены обычно использовались таблица для записи букв алфавита и ключевое слово (или фраза). В таблицу сначала вписывалось по строкам ключевое слово, причем повторяющиеся буквы отбрасывались. Затем эта таблица дополнялась не вошедшими в нее буквами алфавита по порядку. На рис.7 изображена таблица с ключевым словом «ДЯДИНА».

Д	Я	И	Н	А	Б
В	Г	Е	Ё	Ж	З
Й	К	Л	М	О	П
Р	С	Т	У	Ф	Х
Ц	Ч	Ш	Щ	Ъ	Ы
Ь	Э	Ю	-	-	-

Рис.7. Таблица шифрозамен для шифра Трисемуса

Каждая буква открытого сообщения заменяется буквой, расположенной под ней в том же столбце. Если буква находится в последней строке таблицы, то для ее шифрования берут самую верхнюю букву столбца. Например, исходное сообщение «АБРАМОВ», зашифрованное – «ЖЗЦЖУФЙ».

Шифр масонов. В XVIII в. масоны создали шифр, чтобы скрыть от общественности свои коммерческие сделки. Как поведали те, кто прежде состоял в рядах этого общества, масоны пользовались способом засекречивания, весьма похожим на шифр розенкрейцеров. В «решетке» и в углах находятся точки, которыми заменяются буквы:

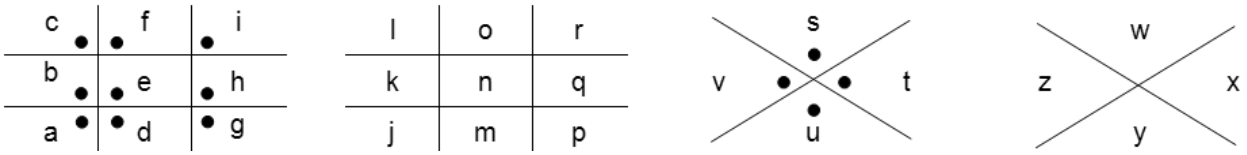


Рис.8. Шифрозамены шифра масонов

Так как клятвы хранить тайну нарушались не раз, большинство Великих лож масонов в США больше не пользуются письменными шифрами, предпочитая передавать устные инструкции во время закрытых ритуалов.

С помощью шифра масонов можно легко расшифровать следующую фразу.

Рис.9. Пример использования шифра масонов

Это первый уровень, на котором находятся все впервые вступившие в общество члены: Blue Lodge (рус. «Голубая (Синяя) ложа»).

Одним из существенных **недостатков шифров однозначной замены** является их легкая вскрываемость. При вскрытии шифрограмм используются различные приемы, которые даже при отсутствии мощных вычислительных средств позволяют добиться положительного результата. Один из таких приемов базируется на том, что в шифрограммах остается информация о частоте встречаемости букв исходного текста. Если в открытом сообщении часто встречается какая-либо буква, то в шифрованном сообщении также часто будет встречаться соответствующий ей символ. Еще в 1412 г. Шихаба ал-Калкашанди в своем труде «Субх ал-Ааша» привел таблицу частоты появления арабских букв в тексте на основе анализа текста Корана. Для разных языков мира существуют подобные таблицы. Так, например, для букв русского алфавита по данным "Национального корпуса русского языка" такая таблица выглядит следующим образом.

Таблица 2. Частота появления букв русского языка в текстах

№ п/п	Буква	Частота, %	№ п/п	Буква	Частота, %
1	О	10.97	18	Ь	1.74
2	Е	8.45	19	Г	1.70
3	А	8.01	20	З	1.65
4	И	7.35	21	Б	1.59
5	Н	6.70	22	Ч	1.44
6	Т	6.26	23	Й	1.21
7	С	5.47	24	Х	0.97
8	Р	4.73	25	Ж	0.94
9	В	4.54	26	Ш	0.73

10	Л	4.40	27	Ю	0.64
11	К	3.49	28	Ц	0.48
12	М	3.21	29	Щ	0.36
13	Д	2.98	30	Э	0.32
14	П	2.81	31	Ф	0.26
15	У	2.62	32	Ъ	0.04
16	Я	2.01	33	Ё	0.04
17	Ы	1.90			

Существуют подобные таблицы для пар букв (биграмм). Например, часто встречаемыми биграммами являются «то», «но», «ст», «по», «ен» и т.д. Другой прием вскрытия шифрограмм основан на исключении возможных сочетаний букв. Например, в текстах (если они написаны без орфографических ошибок) нельзя встретить сочетания «чя», «щы», «ьъ» и т.п.

Для усложнения задачи вскрытия шифров однозначной замены еще в древности перед шифрованием из исходных сообщений исключали пробелы и/или гласные буквы. Другим способом, затрудняющим вскрытие, является шифрование **биграммами** (парами букв).

Полиграммные шифры

Полиграммные шифры - шифры, в которых одна шифрозамена соответствует сразу нескольким символам исходного сообщения.

Биграммный шифр Порты. Шифр Порты, представленный им в виде таблицы, является первым известным биграммным шифром. Размер его таблицы составлял 20 х 20 ячеек; наверху горизонтально и слева вертикально записывался стандартный алфавит (в нем не было букв J, K, U, W, X и Z). В ячейках таблицы могли быть записаны любые числа, буквы или символы - сам Джованни Порта пользовался символами - при условии, что содержимое ни одной из ячеек не повторялось. Применительно к русскому языку таблица шифрозамен может выглядеть следующим образом.

	А	Б	В	Г	Д	Е (Ё)	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
А	001	002	003	004	005	006	007	008	009	010	011	012	013	014	015	016	017	018	019	020	021	022	023	024	025	026	027	028	029	030	031
Б	032	033	034	035	036	037	038	039	040	041	042	043	044	045	046	047	048	049	050	051	052	053	054	055	056	057	058	059	060	061	062

В	063	064	065	066	067	068	069	070	071	072	073	074	075	076	077	078	079	080	081	082	083	084	085	086	087	088	089	090	091	092	093
Г	094	095	096	097	098	099	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124
Д	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155
Е (Ё)	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186
Ж	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217
З	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248
И (Й)	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279
К	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310
Л	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341
М	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372
Н	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403
О	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434

П	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465
Р	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496
С	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527
Т	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558
У	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589
Ф	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620
Х	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651
Ц	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682
Ч	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713
Ш	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744
Щ	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775
Ъ	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806
Ы	80	80	80	81	81	81	81	81	81	81	81	81	81	81	82	82	82	82	82	82	82	82	82	82	83	83	83	83	83	83	83

	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7
Ь	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868
Э	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899
Ю	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930
Я	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961

Рис.10. Таблица шифрозамен для шифра Порты

Шифрование выполняется парами букв исходного сообщения. Первая буква пары указывает на строку шифрозамены, вторая - на столбец. В случае нечетного количества букв в исходном сообщении к нему добавляется вспомогательный символ («пустой знак»). Например, исходное сообщение «АБ РА МО В», зашифрованное – «002 466 355 093». В качестве вспомогательного символа использована буква «Я».

Шифр Playfair (англ. «Честная игра»). В начале 1850-х гг. Чарлз Уитстон придумал так называемый «прямоугольный шифр». Леон Плейфер, близкий друг Уитстона, рассказал об этом шифре во время официального обеда в 1854 г. министру внутренних дел лорду Пальмерстону и принцу Альберту. А поскольку Плейфер был хорошо известен в военных и дипломатических кругах, то за творением Уитстона навечно закрепилось название «шифр Плейфера».

Данный шифр стал первым буквенным биграммным шифром (в биграммной таблице Порты использовались символы, а не буквы). Он был предназначен для обеспечения секретности телеграфной связи и применялся британскими войсками в Англо-бурской и Первой мировой войнах. Им пользовалась также австралийская служба береговой охраны островов во время Второй мировой войны.

Шифр предусматривает шифрование пар символов (биграмм). Таким образом, этот шифр более устойчив к взлому по сравнению с шифром простой замены, так как затрудняется частотный анализ. Он может быть проведен, но не для 26 возможных символов (латинский алфавит), а для 26 х 26 = 676 возможных биграмм. Анализ частоты биграмм возможен, но является значительно более трудным и требует намного большего объема зашифрованного текста.

Для шифрования сообщения необходимо разбить его на биграммы (группы из двух символов), при этом, если в биграмме встретятся два одинаковых символа, то между ними добавляется заранее оговоренный вспомогательный символ (в оригинале – **X**, для русского алфавита - **Я**). Например, «зашифрованное

сообщение» становится «за ши фр ов ан но ес о**Я** об ще ни е**Я**». Для формирования ключевой таблицы выбирается лозунг и далее она заполняется по правилам шифрующей системы Трисемуса. Например, для лозунга «ДЯДИНА» ключевая таблица выглядит следующим образом.

Д	Я	И	Н	А	Б
В	Г	Е	Ё	Ж	З
Й	К	Л	М	О	П
Р	С	Т	У	Ф	Х
Ц	Ч	Ш	Щ	Ъ	Ы
Ь	Э	Ю	-	1	2

Рис.11. Ключевая таблица для шифра Playfair

Затем, руководствуясь следующими правилами, выполняется зашифровывание пар символов исходного текста:

1. Если символы биграммы исходного текста встречаются в одной строке, то эти символы замещаются на символы, расположенные в ближайших столбцах справа от соответствующих символов. Если символ является последним в строке, то он заменяется на первый символ этой же строки.

2. Если символы биграммы исходного текста встречаются в одном столбце, то они преобразуются в символы того же столбца, находящимися непосредственно под ними. Если символ является нижним в столбце, то он заменяется на первый символ этого же столбца.

3. Если символы биграммы исходного текста находятся в разных столбцах и разных строках, то они заменяются на символы, находящиеся в тех же строках, но соответствующие другим углам прямоугольника.

Пример шифрования.

- биграмма «за» формирует прямоугольник – заменяется на «жб»;
- биграмма «ши» находятся в одном столбце – заменяется на «юе»;
- биграмма «фр» находятся в одной строке – заменяется на «хс»;
- биграмма «ов» формирует прямоугольник – заменяется на «йж»;
- биграмма «ан» находятся в одной строке – заменяется на «ба»;
- биграмма «но» формирует прямоугольник – заменяется на «ам»;
- биграмма «ес» формирует прямоугольник – заменяется на «гт»;
- биграмма «оя» формирует прямоугольник – заменяется на «ка»;
- биграмма «об» формирует прямоугольник – заменяется на «па»;
- биграмма «ще» формирует прямоугольник – заменяется на «шё»;
- биграмма «ни» формирует прямоугольник – заменяется на «ан»;
- биграмма «ея» формирует прямоугольник – заменяется на «ги».

Шифрограмма – «жб юе хс йж ба ам гт ка па шё ан ги».

Для расшифровки необходимо использовать инверсию этих правил, откидывая символы **Я** (или **Х**), если они не несут смысла в исходном сообщении.

Шифр Хилла. Первый практически реализуемый способ шифрования с использованием алгебры был придуман в 1929 г. математиком Лестером Хиллом - профессором из Хантер-колледжа в Нью-Йорке, статья которого «Cryptography in an Algebraic Alphabet» была опубликована в журнале «The American Mathematical Monthly».

Каждой букве алфавита сопоставляется число. Для русского алфавита можно использовать простейшую схему: А = 0, Б = 1, ..., Я = 32. Для зашифрования блок исходного сообщения из **n** букв рассматривается как n-мерный вектор чисел и умножается на матрицу размером **n** x **n** по модулю 33. Данная матрица, совместно с кодовой таблицей сопоставления букв алфавита с числами, является ключом зашифрования. Для расшифрования применяется обратная матрица[1] по модулю.

Например, для триграммных замен могут использоваться следующие матрицы зашифрования / расшифрования.

6	27	1
13	16	32
28	17	15

матрица зашифрования

2	26	17
26	20	4
13	30	21

матрица расшифрования

Рис.12. Матрицы зашифрования / расшифрования

Исходное сообщение «АБРАМОВ», дополненное двумя вспомогательными буквами «яя» (для кратности трем), после сопоставления букв с числами будет выглядеть следующим образом «0 1 17 0 13 15 2 32 32». После перемножения троек чисел на матрицу зашифрования шифрограмма примет следующий вид «11 32 8 3 28 17 17 11 24» (или в буквенном эквиваленте «КЯЗ ГЪР РКЧ»).

АБР - 0 1 17

(6 * 0 + 27 * 1 + 1 * 17) mod 33 = 11 (К)

(13 * 0 + 16 * 1 + 32 * 17) mod 33 = 32 (Я)

(18 * 0 + 17 * 1 + 15 * 17) mod 33 = 8 (З)

АМО - 0 13 15

(6 * 0 + 27 * 13 + 1 * 15) mod 33 = 3 (Г)

(13 * 0 + 16 * 13 + 32 * 15) mod 33 = 28 (Ъ)

(28 * 0 + 17 * 13 + 15 * 15) mod 33 = 17 (Р)

Вяя - 2 32 32

(6 * 2 + 27 * 32 + 1 * 32) mod 33 = 17 (Р)

(13 * 2 + 16 * 32 + 32 * 32) mod 33 = 11 (К)

(28 * 2 + 17 * 32 + 15 * 32) mod 33 = 24 (Ч)

Для расшифрования тройки чисел шифрограммы необходимо умножить на матрицу расшифрования.

КЯЗ - 11 32 8
(2 * 11 + 26 * 32 + 17 * 8) mod 33 = 0 (А)
(26 * 11 + 20 * 32 + 4 * 8) mod 33 = 1 (Б)
(13 * 11 + 30 * 32 + 21 * 8) mod 33 = 17 (Р)
ГЪР - 3 28 17
(2 * 3 + 26 * 28 + 17 * 17) mod 33 = 0 (А)
(26 * 3 + 20 * 28 + 4 * 17) mod 33 = 13 (М)
(13 * 3 + 30 * 28 + 21 * 17) mod 33 = 15 (О)
РКЧ - 17 11 24
(2 * 17 + 26 * 11 + 17 * 24) mod 33 = 2 (В)
(26 * 17 + 20 * 11 + 4 * 24) mod 33 = 32 (я)
(13 * 17 + 30 * 11 + 21 * 24) mod 33 = 32 (я)

В результате будет получен набор чисел «0 1 17 0 13 15 2 32 32», соответствующий исходному сообщению со вспомогательными символами «АБРАМОВяя».

[1]**Обратная матрица** - матрица A^{-1}, при умножении на которую, исходная матрица А дает в результате единичную матрицу Е.

Нерегулярные шифры

Еще одним направлением повышения стойкости шифров замены заключается в использовании нерегулярных шифров. В приведенных выше шифрах (**регулярных**) шифрозамены состоят из строго определенного количества символов (букв, цифр, графических элементов и т.д.) или в шифрограмме они отделяются друг от друга специальными символами (пробелом, точкой, запятой и т.д.). В нерегулярных шифрах шифрозамены состоят из разного количества символов и записываются в шифрограмме в подряд (без выделения друг от друга), что значительно затрудняет криптоанализ.

Совмещенный шифр (совмещенная таблица). Данный шифр применялся еще семейством Ардженти - криптологами, разрабатывавшими шифры для Папы Римского в XVI в. В XX столетии этим способом пользовались коммунисты в ходе гражданской войны в Испании. В начале войны противники фашизма в Испании контролировали большинство крупных городов и защищали свою связь, включая радиопередачи, с помощью различных методов шифрования, в том числе совмещенных шифров.

Вариант коммунистов получил название «совмещенный» из-за необычного использования одно- и двухцифровых шифрозамен, благодаря чему сообщение приобретало дополнительную защиту от потенциального дешифровальщика. Некоторые буквы зашифровывались одной цифрой, другие же - парой цифр. При этом криптоаналитик противника совершенно не представлял, где в перехваченных сообщениях находятся одноцифровые, а где двухцифровые шифрозамены.

Таблица шифрозамен состоит из 10 столбцов с нумерацией 0, 9, 8, 7, 6, 5, 4, 3, 2 и 1. В начальную строку вписывается ключевое слово без повторяющихся букв. В последующие строки вписываются по десять не вошедших в него букв по порядку следования в алфавите. Строки, за исключением начальной, нумеруются по порядку, начиная с 1.

	0	9	8	7	6	5	4	3	2	1
	Д	Я	И	Н	А					
1	Б	В	Г	Е	Ё	Ж	З	Й	К	Л
2	М	О	П	Р	С	Т	У	Ф	Х	Ц
3	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	-	-

Рис.13. Пример таблицы шифрозамен совмещенного шифра с ключевым словом «ДЯДИНА»

При шифровании буквы исходного сообщения, входящие в ключевое слово, заменяются на одну цифру (номер столбца), остальные – двумя (номера строки и столбца). Например, для приведенной выше таблицы шифрозамен исходное сообщение «АБРАМОВ» будет зашифровано как «610276202919».

При получении шифрограммы адресат знает, что когда появляются цифры 1, 2 или 3, с ними обязательно связана еще одна цифра, поскольку они представляют собой цифровую пару. Так что 35 - это, несомненно, пара, а 53 - нет, ведь в таблице нет строки с номером 5. Перехват такого сообщения третьей стороной даст ей всего лишь ряд цифр, потому что криптоаналитик противника не имеет ни малейшего представления, какие цифры одиночные, а какие входят в состав пар.

Омофонические шифры

Другое направление повышения стойкости шифров замены состоит в том, чтобы каждое множество шифрообозначений $\mathbf{M_i}$ для отдельного **i**-го символа исходного алфавита содержало более одного элемента. При использовании такого шифра одну и ту же букву (если она встречается несколько раз в сообщении) заменяют на разные шифрозамены из $\mathbf{M_i}$. Это позволяет скрыть истинную частоту встречаемости букв открытого сообщения.

Система омофонов. В 1401 г. Симеоне де Крема стал использовать таблицы омофонов для сокрытия частоты появления гласных букв в тексте при помощи более чем одной шифрозамены. Такие шифры позже стали называться **шифрами многозначной замены** или **омофонами**. Они получили развитие в XV веке. В книге «Трактат о шифрах» Леона Баттисты Альберти (итальянский ученый, архитектор, теоретик искусства, секретарь папы Климентия XII), опубликованной в 1466 г., приводится описание шифра замены, в котором каждой букве ставится в соответствие несколько эквивалентов, число которых пропорционально частоте встречаемости буквы в открытом тексте. Так, если ориентироваться на табл.2, то число шифрозамен для буквы **О** должно составлять 110, для буквы **Е** – 85 и т.д.

При этом каждая шифрозамена должна состоять из 3 цифр и их общее количество равно 1000. На рис.12 представлен фрагмент таблицы шифрозамен.

№ п/п	А	Б	В	…	М	…	О	…	Р	…	Я
1	311	128	175	…	037	…	248	…	064	…	266
2	357	950	194	…	149	…	267	…	189	…	333
…	…	…	…	…	…	…	…	…	…	…	…
16	495	990	199	…	349	…	303	…	374	…	749
…	…		…	…	…	…	…	…	…	…	…
20	519		427	…	760	…	306	…	469	…	845
…	…		…	…	…	…	…	…	…		
32	637		524	…	777	…	432	…	554		
…	…		…				…	…	…		
45	678		644				824	…	721		
…	…						…	…	…		
47	776						828	…	954		
…	…						…				
80	901						886				
…							…				
110							903				

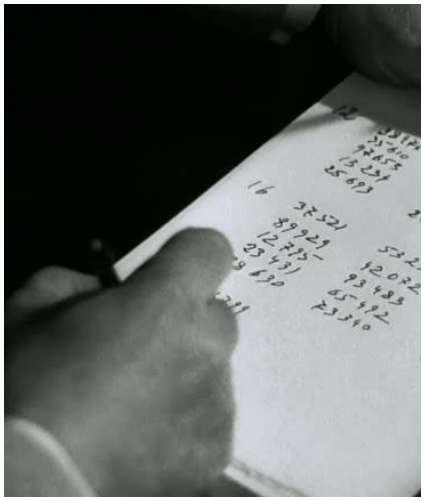
Рис.14. Фрагмент таблицы шифрозамен для системы омофонов

При шифровании символ исходного сообщения заменяется на любую шифрозамену из своего столбца. Если символ встречается повторно, то, как правило, используют разные шифрозамены. Например, исходное сообщение «АБРАМОВ» после шифрования может выглядеть «357 990 374 678 037 828 175».

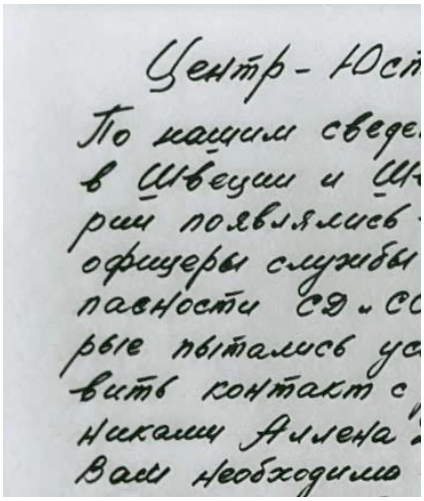
Книжный шифр. Заметным вкладом греческого ученого Энея Тактика в криптографию является предложенный им так называемый книжный шифр, описанный в сочинении «Об обороне укреплённых мест». Эней предложил прокалывать малозаметные дырки в книге или в другом документе над буквами секретного сообщения. Интересно отметить, что в Первой мировой войне германские шпионы использовали аналогичный шифр, заменив дырки на точки, наносимые симпатическими чернилами на буквы газетного текста. Описанные способы передачи секретных сообщений (с помощью точек) относяться к стеганографическим методам сокрытия информации.

После Первой мировой войны книжный шифр приобрел иной вид. Шифрозамена для каждой буквы определялась набором цифр, которые указывали на номер страницы, строки и позиции в строке.

а) ключ (4-ый том собрания сочинений Фридриха Шиллера)



б) шифрограмма



в) сообщение

Рис.15. Пример использования книжного шифра
(кадры из советского сериала «Семнадцать мгновений весны»)

Количество книг, изданных за всю историю человечества, является величиной ограниченной (по крайней мере, явно меньше, чем 15!). Однако отсутствие полной электронной базы по изданиям делает процедуру вскрытия шифрограмм почти не выполнимой.

Вариантные шифры. Вариантные шифры напоминают полибианский квадрат, но для каждой строки и столбца используется по два буквенных идентификатора. В квадрат (прямоугольник) шифрозамен вначале записывается ключевое слово без повторяющихся букв, а затем дополняется не вошедшими в него буквами по порядку следования в алфавите. Каждой строке и столбцу квадрата ставится в соответствие по две буквы алфавита. Буквы для идентификации строк и столбцов не должны повторяться.

	Й Ц	У К	Е Н	Г Ш	Щ З	Х Ъ
Ф Ы	Д	Я	И	Н	А	Б
В А	В	Г	Е	Ё	Ж	З
П Р	Й	К	Л	М	О	П
О Л	Р	С	Т	У	Ф	Х
Д Ж	Ц	Ч	Ш	Щ	Ъ	Ы
Э Я	Ь	Э	Ю	-	-	-

Рис.16. Пример таблицы шифрозамен вариантного шифра с ключевым словом «ДЯДИНА»

Комбинации букв-идентификаторов строки и столбца дают по восемь шифрозамен для каждой буквы исходного текста. Например, для

буквы Д возможны шифрозамены: **ФЙ**, **ЙФ**, **ФЦ**, **ЦФ**, **ЫЙ**, **ЙЫ**, **ЫЦ** и **ЦЫ**. Для таблицы шифрозамен, приведенной на рис.16, исходное сообщение «АБРАМОВ» может быть зашифровано как «ЫЗ ЫХ ОЦ ЗФ ГР РЩ АЙ».

Полиалфавитные шифры

В полиалфавитных шифрах используется нескольких алфавитов шифрозамен. Выбор варианта алфавита шифрозамен для зашифрования отдельного символа или блока символов зависит от особенностей шифра.

Диск Альберти. В «Трактате о шифрах» Альберти приводит первое точное описание многоалфавитного шифра на основе шифровального диска.

Рис.17. Реплика диска Альберти, используемого Конфедерацией во время Гражданской войны в Америке

Он состоял из двух дисков – внешнего неподвижного и внутреннего подвижного дисков, на которые были нанесены буквы алфавита. Процесс шифрования заключался в нахождении буквы открытого текста на внешнем диске и замене ее на букву с внутреннего диска, стоящую под ней. После этого внутренний диск сдвигался на одну позицию и шифрование второй буквы производилось уже по новому шифралфавиту. Ключом данного шифра являлся порядок расположения букв на дисках и начальное положение внутреннего диска относительно внешнего.

Таблица Трисемуса. Одним из шифров, придуманных немецким аббатом Трисемусом, стал многоалфавитный шифр, основанный на так называемой «таблице Трисемуса» - таблице со стороной равной **n**, где **n** – количество символов в алфавите. В первой строке матрицы записываются буквы в порядке их очередности в алфавите, во второй – та же последовательность букв, но с

циклическим сдвигом на одну позицию влево, в третьей – с циклическим сдвигом на две позиции влево и т.д.

А	Б	В	Г	Д	Е	Ё	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
Б	В	Г	Д	Е	Ё	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	А
В	Г	Д	Е	Ё	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	А	Б
Г	Д	Е	Ё	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	А	Б	В
Д	Е	Ё	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	А	Б	В	Г
Е	Ё	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	А	Б	В	Г	Д
Ё	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	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	А	Б	В	Г	Д	Е	Ё
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	А	Б	В	Г	Д	Е	Ё	Ж
И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З
Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И
К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й
Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К
М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л
Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М
О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н
П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П
С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р
Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С
У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т
Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У
Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф
Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х
Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц
Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч
Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш
Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ
Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ

Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы
Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь
Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э
Я	А	Б	В	Г	Д	Е	Ё	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

Рис.18. Таблица Трисемуса

Первая строка является одновременно и алфавитом для букв открытого текста. Первая буква текста шифруется по первой строке, вторая буква по второй и так далее. После использования последней строки вновь возвращаются к первой. Так сообщение «АБРАМОВ» приобретет вид «АВТГРУЗ».

Система шифрования Виженера. В 1586 г. французский дипломат Блез Виженер представил перед комиссией Генриха III описание простого, но довольно стойкого шифра, в основе которого лежит таблица Трисемуса.

Перед шифрованием выбирается ключ из символов алфавита. Сама процедура шифрования заключается в следующем. По i-ому символу открытого сообщения в первой строке определяется столбец, а по i-ому символу ключа в крайнем левом столбце – строка. На пересечении строки и столбца будет находиться i-ый символ, помещаемый в шифрограмму. Если длина ключа меньше сообщения, то он используется повторно. Например, исходное сообщение «АБРАМОВ», ключ – «ДЯДИНА», шифрограмма – «ДАФИЪОЁ».

Справедливости ради, следует отметить, что авторство данного шифра принадлежит итальянцу Джованни Батиста Беллазо, который описал его в 1553 г. История «проигнорировала важный факт и назвала шифр именем Виженера, несмотря на то, что он ничего не сделал для его создания». Беллазо предложил называть секретное слово или фразу **паролем** (ит. password; фр. parole - слово).

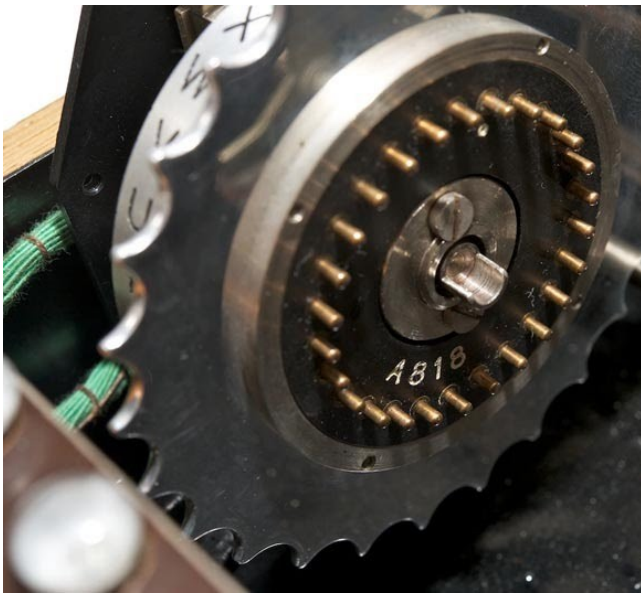
В 1863 г. Фридрих Касиски опубликовал алгоритм атаки на этот шифр, хотя известны случаи его взлома шифра некоторыми опытными криптоаналитиками и ранее. В частности, в 1854 г. шифр был взломан изобретателем первой аналитической вычислительной машины Чарльзом Бэббиджем, хотя этот факт стал известен только в XX в., когда группа ученых разбирала вычисления и личные заметки Бэббиджа. Несмотря на это шифр Виженера имел репутацию исключительно стойкого к «ручному» взлому еще долгое время. Так, известный писатель и математик Чарльз Лютвидж Доджсон (Льюис Кэрролл) в своей статье «Алфавитный шифр», опубликованной в детском журнале в 1868 г., назвал шифр Виженера невзламываемым. В 1917 г. научно-популярный журнал «Scientific American» также отозвался о шифре Виженера, как о неподдающемся взлому.

Роторные машины. Идеи Альберти и Беллазо использовались при создании электромеханических роторных машин первой половины XX века. Некоторые из них использовались в разных странах вплоть до 1980-х годов. В большинстве из них использовались роторы (механические колеса), взаимное расположение которых определяло текущий алфавит шифрозамен, используемый для выполнения подстановки. Наиболее известной из роторных машин является немецкая машина времен Второй мировой войны «Энигма».

Рис.19. Энигма

Выходные штыри одного ротора соединены с входными штырями следующего ротора и при нажатии символа исходного сообщения на клавиатуре замыкали электрическую цепь, в результате чего загоралась лампочка с символом шифрозамены.

а) четыре последовательно соединённых ротора



б) штыри ротора

Рис.20. Роторная система Энигмы

Шифрующее действие «Энигмы» показано для двух последовательно нажатых клавиш - ток течёт через роторы, «отражается» от рефлектора, затем снова через роторы.

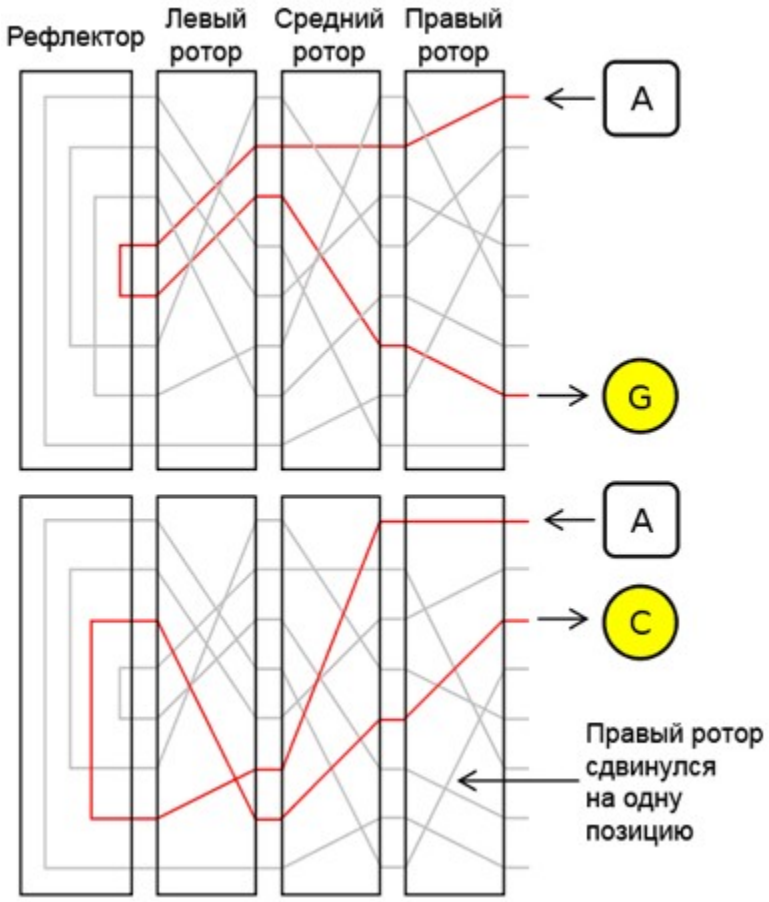


Рис.21. Схема шифрования

Примечание. Серыми линиями показаны другие возможные электрические цепи внутри каждого ротора. Буква **A** шифруется по-разному при последовательных нажатиях одной клавиши, сначала в **G**, затем в **C**. Сигнал идет по другому маршруту за счёт поворота одного из роторов после нажатия предыдущей буквы исходного сообщения.

Шифры Тени. Главными развлечениями для американцев тридцатых годов XX века были бульварное чтиво и радио. Для раскрутки своих книжек издательство Street & Smith проспонсировало радиопередачу, ведущим в которой был Тень (англ. Shadow), загадочный рассказчик со зловещим голосом, который в начале каждого выпуска заявлял: «Кто знает, что за зло прячется в сердцах людей? Тень знает!». Успех радиопередачи подтолкнул издательство к решению начать выпускать серию книг, в которой главным героем был бы Тень. Свои услуги предложил Уолтер Гибсон, большой любитель фокусов и головоломок. Под псевдонимом Максвелл Грант он принялся писать роман за романом, да с такой скоростью, что за свою жизнь написал почти 300 книжек о грозе тех, кто нечист помыслами. В новелле «Цепочка смерти» супергерой воспользовался так называемым кодом направления, хотя на самом деле он действует скорее как шифр, чем как код.

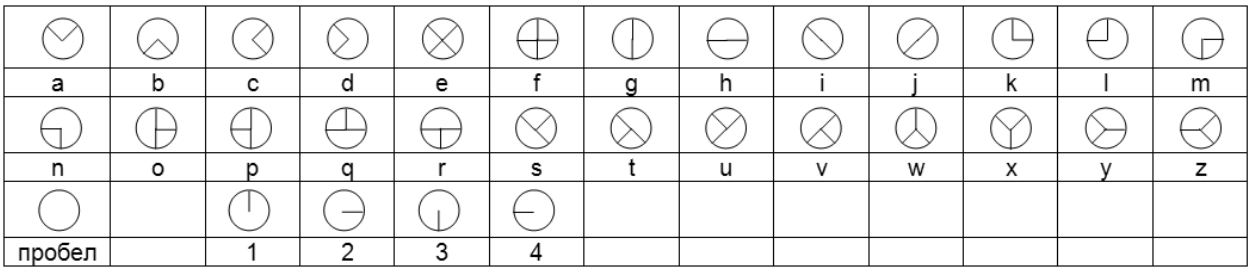


Рис.22. Таблица шифрозамен и управляющих символов

Управляющие символы в последней строке таблицы служат для изменения кода зашифрования/дешифрования (выбора шифралфавита). Линии внутри управляющего символа указывают адресату, как держать лист бумаги для дешифрования очередного символа шифрограммы. Символ 1 означает, что лист надо держать как обычно: верх и низ расположены на своих местах. Символ 2 требует для дешифрования очередного символа поворота листа на 90° вправо. Управляющие символы могут появляться перед любой строчкой текста, а также в ее середине.

Из нижеприведенного примера можно узнать настоящие имя и фамилию супергероя.

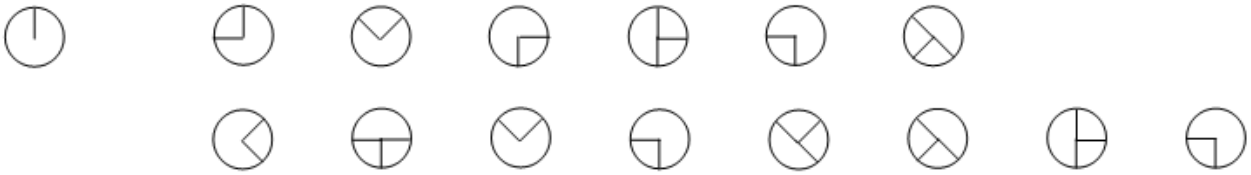


Рис.23. Настоящие имя и фамилия Тени

Согласно первому управляющему символу, лист следует держать обычным образом, не поворачивая, и после замены буквы образуют «Lamont Cranston» (Ламонт Крэнстон).

Вопросы для самоконтроля

1. Социальные причины киберпреступлений.
2. Экономические причины киберпреступлений.
3. Правовые причины киберпреступлений.
4. Проблемы кибербезопасности.

Практическое занятие №3. Комбинированные шифры

Цель: изучить способы шифрования данных различными комбинациями.

Ход работы:

Необходимо ознакомиться с теоретической частью и зашифровать свою фамилию и имя с помощью шифра ADFGVX.

При выполнении задания необходимо привести исходное сообщение (фамилию и имя), таблицу шифрозамен, ключевое слово, перестановочную таблицу и зашифрованное сообщение.

Шифры ADFGX и ADFGVX

Комбинированные (составные) шифры предполагают использование для шифрования сообщения сразу нескольких методов (например, сначала замена символов, а затем их перестановка).

Два самых известных полевых шифра в истории криптографии - ADFGX и ADFGVX. Шифр ADFGX впервые был использован во время решающих этапов Первой мировой войны, когда в марте 1918 г. кайзеровские генералы начали крупное наступление. Шифр ADFG(V)X был разработан полковником Фрицем Небелем, офицером связи, служившим в штабе германской армии. Оба шифра предполагают вначале применение к буквам исходного сообщения замену, после чего для получения окончательной шифрограммы выполняется перестановка.

Таблица шифрозамен ADFGX представляет собой матрицу 5 x 5, а для ADFGVX – 6 x 6. Строки и столбцы обозначаются буквами, входящими в название шифра. Пример таблицы шифрозамен для шифра ADFGVX применительно к русскому алфавиту показан на следующем рисунке.

	A	D	F	G	V	X
A	Ю	У	И	Ч	К	Б
D	В	Г	Е	Ф	Ж	З
F	Й	А	Л	М	О	П
G	Р	Щ	Т	Я	Ё	Х
V	Ц	Н	Ш	С	Ъ	Ы
X	Ь	Э	Д	-	-	-

Рис.1. Пример таблицы шифрозамен для шифра ADFGVX

Шифрозамена для буквы исходного текста состоит из букв, обозначающих строку и столбец, на пересечении которых она находится (см. шифр «Полибианский квадрат»). Например, для сообщения «АБРАМОВ» набор шифрозамен будет «FD AX GA FD FG FV DA».

На втором этапе для выполнения перестановки полученный набор шифрозамен вписывается построчно сверху-вниз в таблицу, количество столбцов в которой строго определено (ADFGX) или соответствует количеству букв в

ключевом слове (ADFGVX). Нумерация столбов либо оговаривается сторонами (ADFGX) либо соответствует положению букв ключевого слова в алфавите, как в шифре вертикальной перестановки (ADFGVX). Например, для полученного выше набора шифрозамен перестановочная таблица с ключевым словом «ДЯДИНА» показана на следующем рисунке.

Д	Я	Д	И	Н	А
2	6	3	4	5	1
F	D	A	X	G	A
F	D	F	G	F	V
D	A				

Рис.2. Перестановочная таблица шифра ADFGVX с ключевым словом «ДЯДИНА»

На третьем этапе буквы выписываются из столбцов в соответствии с их нумерацией, при этом считывание происходит по столбцам, а буквы объединяются в пятибуквенные группы. Таким образом, окончательная шифрограмма для рассматриваемого примера будет выглядеть «AVFFD AFXGG FDDA».

Основы блочного комбинированного шифрования

Среди комбинированных методов шифрования наиболее распространенными являются методы блочного шифрования. **Блочное шифрование** предполагает разбиение исходного открытого текста на равные блоки, к которым применяется однотипная процедура шифрования. В настоящее время блочные шифры широко используются на практике. Российский и бывший американский стандарты шифрования (DES) относятся именно к этому классу шифров. В основе шифров лежат так называемые «**сети Фейстеля**». В 1971 г. Хорст Фейстель (Horst Feistel) запатентовал два устройства с различными алгоритмами шифрования, названные затем общим именем «Люцифер» (Lucifer). Одно из устройств использовало конструкцию, впоследствии названную «сетью Фейстеля» («Feistel cipher», «Feistel network»). Работа над созданием новых криптосистем велась им в стенах IBM вместе с Доном Копперсмитом (Don Coppersmith). Проект «Люцифер» был скорее экспериментальным, но стал базисом для алгоритма Data Encryption Standard (DES). В 1977 г. DES стал стандартом в США на шифрование данных вплоть до 2001 г. и до последнего времени широко использовался в криптографических системах.

Сеть Фейстеля состоит из нескольких **ячеек**.

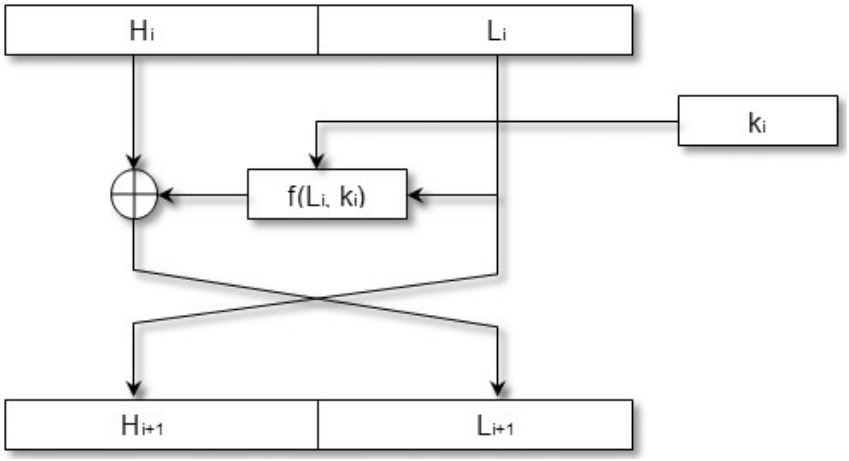


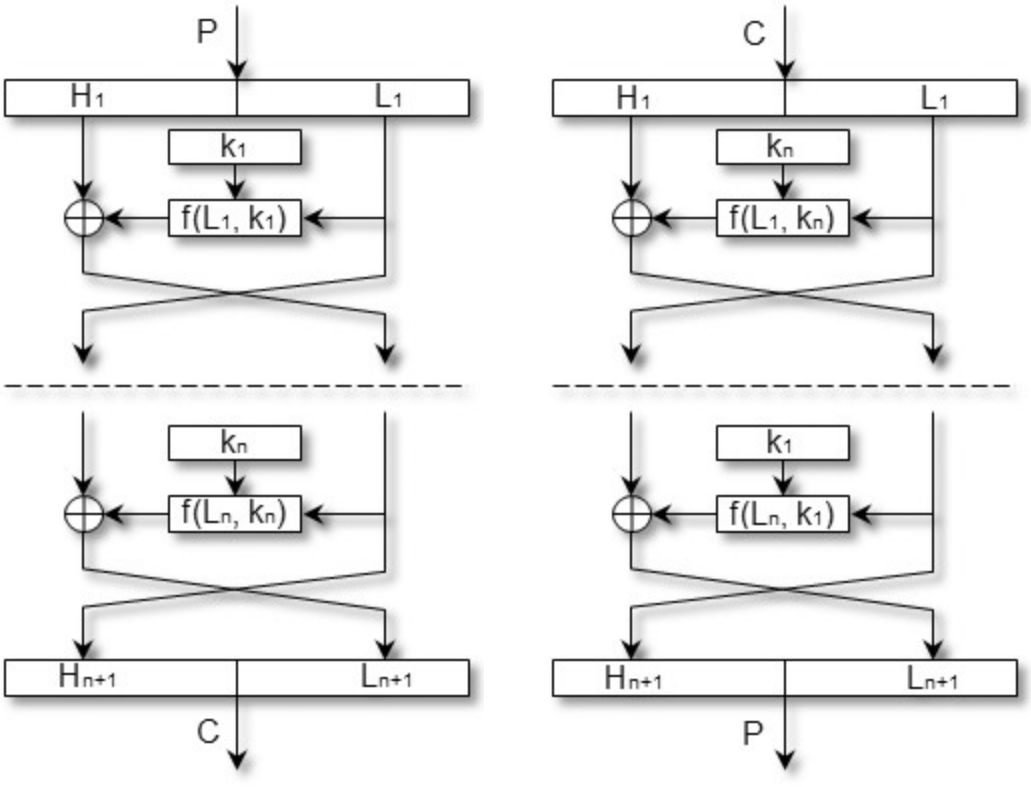
Рис.3. Ячейка Фейстеля

На определенном раунде шифрования шифруемый блок разбивается на две равные половинки - $\mathbf{H_i}$ (высший, high) и $\mathbf{L_i}$ (низкий, low). К правой половинке применяется функция шифрования **f** с использованием ключевого элемента $\mathbf{k_i}$ (части ключа или модификации части ключа). После этого выполняется сложение по модулю два левой половинки и результата модификации правой половинки. В результате шифруемым блоком для следующего раунда будет объединение половинок, полученных по формулам

$$H_{i+1} = L_i, \qquad (1)$$

$$L_{i+1} = H_i \oplus f(L_i, k_i). \qquad (2)$$

В общем виде, центральная часть блочных шифров, основанных на сетях Фейстеля, выглядит следующим образом.



Зашифрование Расшифрование

Рис.4. Сеть Фейстеля

Как видно, процедуры зашифрования и расшифрования полностью идентичны, за исключением порядка использования ключевых элементов $\mathbf{k_i}$. При этом функция шифрования **f** может быть сколь угодно сложной и не обязательно обратимой. Обратимость гарантирует сама конструкция сети.

Шифрование при помощи данной конструкции легко реализуется как на программном уровне, так и на аппаратном, что обеспечивает широкие возможности применения. На основе сетей Фейстеля разработано большое количество шифров, среди которых: DES, ГОСТ 28147-89, Blowfish, CAST, FEAL, IDEA, Khufu, Twofish и многие другие.

DES

DES (Data Encryption Standard, стандарт шифрования данных) - федеральный стандарт шифрования США в 1977-2001 гг. **для использования во всех несекретных правительственных каналах связи** (FIPS PUB 46 «Data Encryption Standard»). Несмотря на то, что в настоящий момент федеральным стандартом шифрования США является Rijndael (AES), рассмотрение DES позволяет понять основные принципы блочного шифрования.

В алгоритме, лежащем в основе DES, используются методы замены, перестановки и гаммирования (сложение по модулю 2).

Открытое сообщение разбивается на блоки длиной 64 бита. Если длина сообщения не кратна 64, оно дополняется справа недостающим количеством битов.

Данные шифруются ключом длиной 56 битов. На самом деле ключ имеет размер 64 бита, однако реально для выработки ключевых элементов используются только 56 из них. Самые младшие биты каждого байта ключа (8-ой, 16-ый, …, 64-ый) не попадают в ключевые элементы и служат исключительно для контроля четности. Требуется, чтобы сумма битов каждого байта ключа, включая контрольный, была четной (четный паритетный бит).

Для решения разнообразных криптографических задач, разработаны **четыре рабочих режима**, реализующих DES:

- электронная кодовая книга ECB (Electronic Code Book);
- сцепление блоков шифра CBC (Cipher Block Chaining);
- обратная связь по шифртексту CFB (Cipher Feed Back);
- обратная связь по выходу OFB (Output Feed Back).

Режим ECB (электронная кодовая книга - Electronic Code Book).

Открытое сообщение разбивают на 64-битовые блоки. Каждый из них шифруют независимо с использованием одного и того же ключа шифрования.

Общая схема шифрования блока изображена на рис.5.

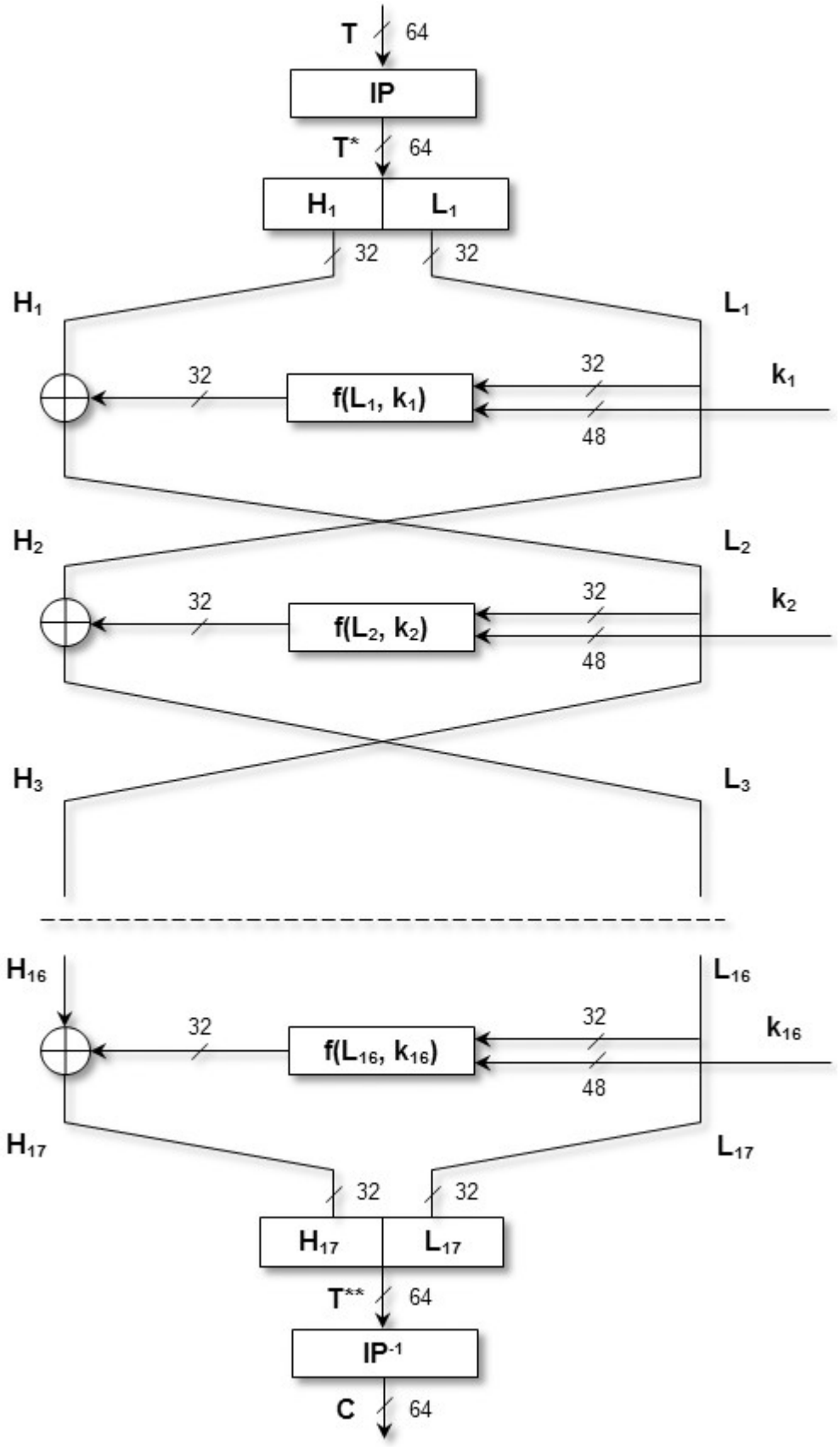


Рис.5. Схема шифрования блока

1. Шифрование 64-битового блока данных **T** начинается с начальной перестановки битов **IP**.

Таблица 1. Начальная перестановка IP

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

В таблице указывается новое положение соответствующего бита. Таким образом, при выполнении начальной перестановки 58-ый бит станет 1-ым, 50-ый – 2-ым, 42-ой – 3-им и т.д.

2. Результат перестановки **T*** разделяется на две 32-битовые части $\mathbf{H_1}$ и $\mathbf{L_1}$, с которыми выполняются 16 раундов преобразования.

3. В каждом раунде **i** старшая половина $\mathbf{H_i}$ блока модифицируется путем побитового прибавления к ней по модулю 2 (⊕) результата вычисления функции шифрования **f**, зависящей от младшей половины блока $\mathbf{L_i}$ и 48-битового ключевого элемента $\mathbf{k_i}$. Ключевой элемент $\mathbf{k_i}$ вырабатывается из ключа шифрования. Между раундами старшая и младшая половины блока меняются местами. В последнем раунде происходит то же самое, за исключением обмена значениями половинок блока.

4. Полублоки $\mathbf{H_{17}}$ и $\mathbf{L_{17}}$ объединяются в полный блок **T****, в котором выполняется конечная битовая перестановка $\mathbf{IP^{-1}}$, обратная начальной.

Таблица 2. Конечная перестановка IP^{-1}

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

Результат последней операции и является выходным значением цикла шифрования – зашифрованным блоком **C**.

Все перестановки в таблицах **IP** и $\mathbf{IP}^{-1}$ подобраны разработчиками таким образом, чтобы максимально затруднить процесс расшифровки путём подбора ключа.

Схема функции шифрования **f** приведена на рис.6.

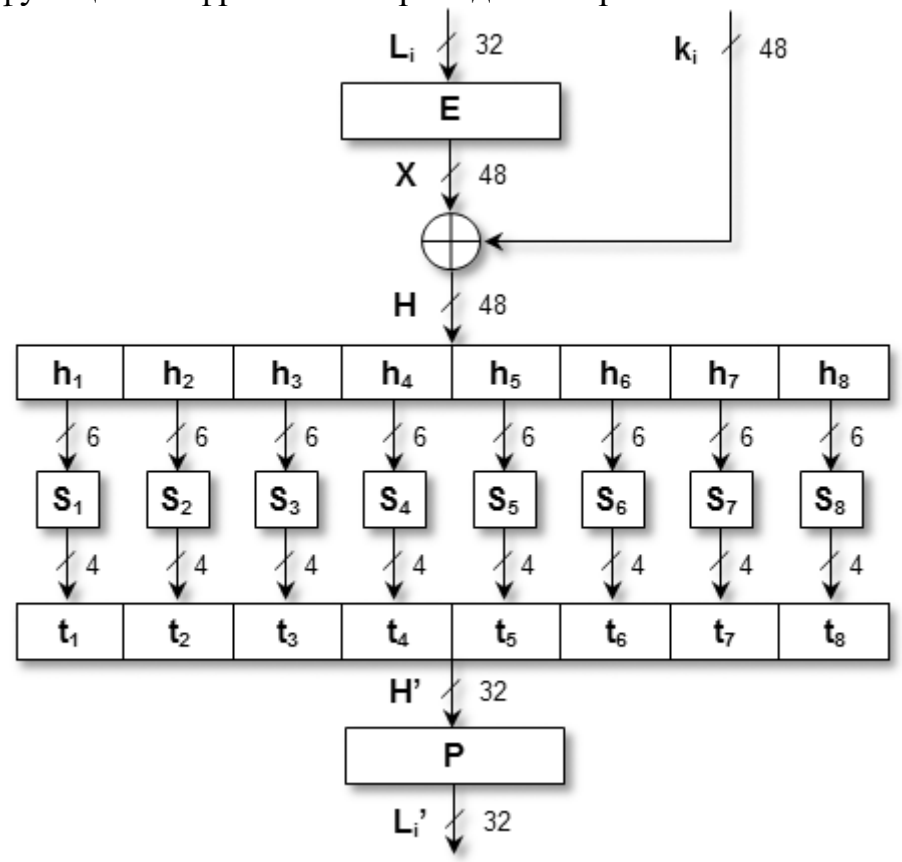


Рис.6. Схема функции шифрования

1. На вход поступает 32-битовая половина шифруемого блока $\mathbf{L_i}$ и 48-битовый ключевой элемент $\mathbf{k_i}$.

2. $\mathbf{L_i}$ разбивается на 8 тетрад по 4 бита. Каждая тетрада по циклическому закону дополняется крайними битами из соседних тетрад до 6-битного слова (функция расширения **E**). Цикличность означает, что первый бит $\mathbf{L_i}$ добавляется последним в последнее слово, а последний бит $\mathbf{L_i}$ добавляется первым в первое слово. Далее выполняется объединение тетрад в 48-битный блок **X**. Например, $\mathbf{L_i}$ = 0111 0110 1… …0 1101_2, тогда **X** = 101110 101101 … 011010_2.

3. **X** побитово суммируется по модулю 2 (⊕) с ключевым элементом $\mathbf{k_i}$.

4. 48-битовый блок данных **H** разделяется на восемь 6-битовых элементов, обозначенных $\mathbf{h_1, h_2, …, h_8}$.

5. Каждое из значений $\mathbf{h_j}$ преобразуется в новое 4-битовое значение $\mathbf{t_j}$ с помощью соответствующего узла замены $\mathbf{S_j}$.

Таблица 3. Значения t_j

Узел замены	Номер строки	Номер столбца															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S_1	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7

	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S_2	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_3	0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4	0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S_5	0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6	0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S_7	0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S_8	0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8

	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Если на вход $\mathbf{S_j}$ поступает блок $\mathbf{h_j=b_1b_2b_3b_4b_5b_6}$, то двухбитовое число $\mathbf{b_1b_6}$ указывает номер строки матрицы, а четырёхбитовое число $\mathbf{b_2b_3b_4b_5}$ - номер столбца в таблице узлов замен. В результате применения узла замены $\mathbf{S_j}$ к блоку $\mathbf{h_j}$ получается число (от 0 до 15), которое преобразуется в $\mathbf{t_j}$. Например, в узел замены S_3 поступает $h_3=101011_2$. Тогда, номер строки равен 3_{10} ($b_1b_6=11_2$), номер столбца – 5_{10} ($b_2b_3b_4b_5=0101_2$), $t_3=9_{10}$ (1001_2).

6. Полученные восемь элементов $\mathbf{t_j}$ вновь объединяются в 32-битовый блок **H'**.

7. В **H'** выполняется перестановка битов **P**.

Таблица 4. Перестановка P

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

Результат последней операции и является выходным значением функции шифрования $\mathbf{L_i'}$.

Ключевые элементы вырабатываются из ключа с использованием сдвигов и битовых выборок-перестановок. Таким образом, ключевые элементы состоят исключительно из битов исходного ключа, «перетасованных» в различном порядке. Схема выработки ключевых элементов показана на следующем рисунке.

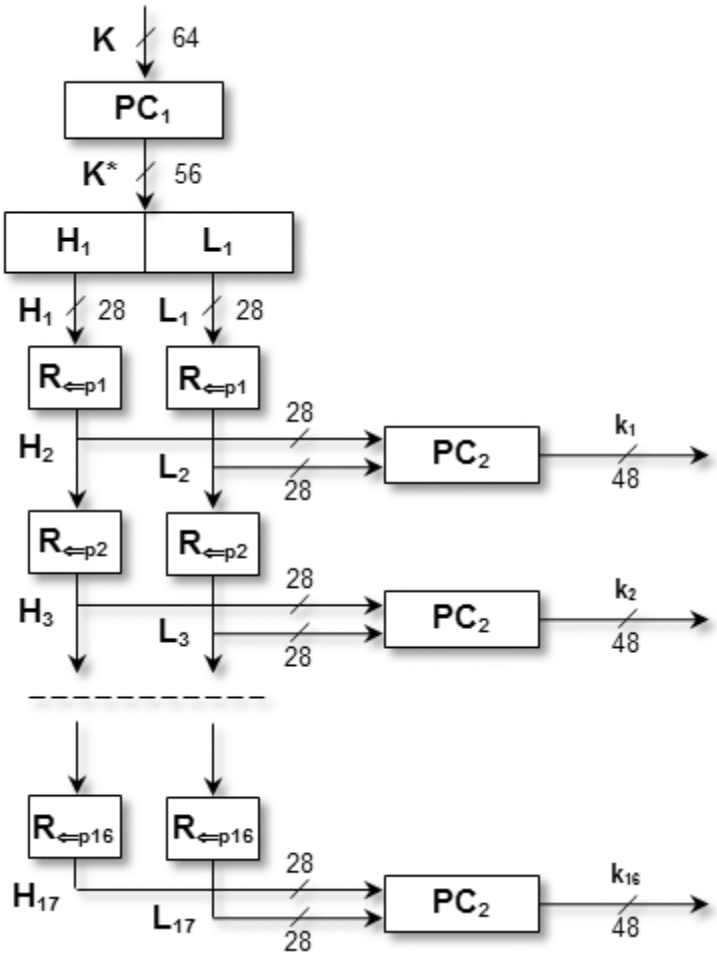


Рис.7. Схема выработки ключевых элементов

1. Выработка ключевых элементов из ключа **K** начинается со входной выборки-перестановки битов **PC**$_1$, которая отбирает 56 из 64 битов ключа и располагает их в другом порядке.

Таблица 5. Перестановка PC$_1$

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

2. Результат выборки-перестановки **K*** разделяется на две 28-битовые части: старшую **H**$_1$ и младшую **L**$_1$.

3. 16 раз выполняется процедура.

3а. В зависимости от номера итерации обе части циклически сдвигаются на 1 или 2 бита влево.

Таблица 6. Циклический сдвиг

Номер итерации	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Сдвиг (бит)	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

3б. После сдвига части объединяются и из них с помощью выборки-перестановки $\mathbf{PC_2}$ отбираются 48 битов, которые и формируют очередной ключевой элемент.

Таблица 7. Перестановка PC_2

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Алгоритмы шифрования и расшифрования DES-ECB в общем виде выражаются следующими схемами

$$C = DES(T) = IP(T) \rightarrow H_1 \oplus f(L_1, k_1), L_1 \rightarrow \ldots \rightarrow H_{16} \oplus f(L_{16}, k_{16}), L_{16} \rightarrow IP^{-1}(2^{32}*L_{17}+H_{17}), \quad (8.3)$$

$$T = DES^{-1}(C) = IP(C) \rightarrow H_1 \oplus f(L_1, k_{16}), L_1 \rightarrow \ldots \rightarrow H_{16} \oplus f(L_{16}, k_1), L_{16} \rightarrow IP^{-1}(2^{32}*L_{17}+ H_{17}). \quad (8.4)$$

Как видно схема алгоритма остается неизменной, меняется лишь порядок использования ключевых элементов. Таким образом, для расшифрования необходимо «прогнать» DES с тем же ключом, но использовать ключевые элементы в обратном порядке.

Использование различных методов шифрования:

- замена - функция расширения **E**, узлы замены **S**;
- перестановка – перестановки **IP**, $\mathbf{IP^{-1}}$, **P**, $\mathbf{PC_1}$, $\mathbf{PC_2}$, чередование $\mathbf{L_i}$ и $\mathbf{H_i}$, циклический сдвиг;
- гаммирование – $\oplus$.

Из-за небольшого числа возможных ключей (всего 256), появляется возможность их полного перебора на быстродействующей вычислительной технике за реальное время. В 1998 г. Electronic Frontier Foundation используя специальный компьютер DES-Cracker, удалось взломать DES за 3 дня. По неподтвержденным данным, Агентство национальной безопасности США уже в

1996 г. могло вскрывать ключ DES за 3-15 мин. с помощью устройства стоимостью 50000 долларов.

Режим CBC (сцепление блоков шифра - Cipher Block Chaining).

Схемы алгоритмов представлены на следующем рисунке. По каждой из стрелок передается 64 бита информации.

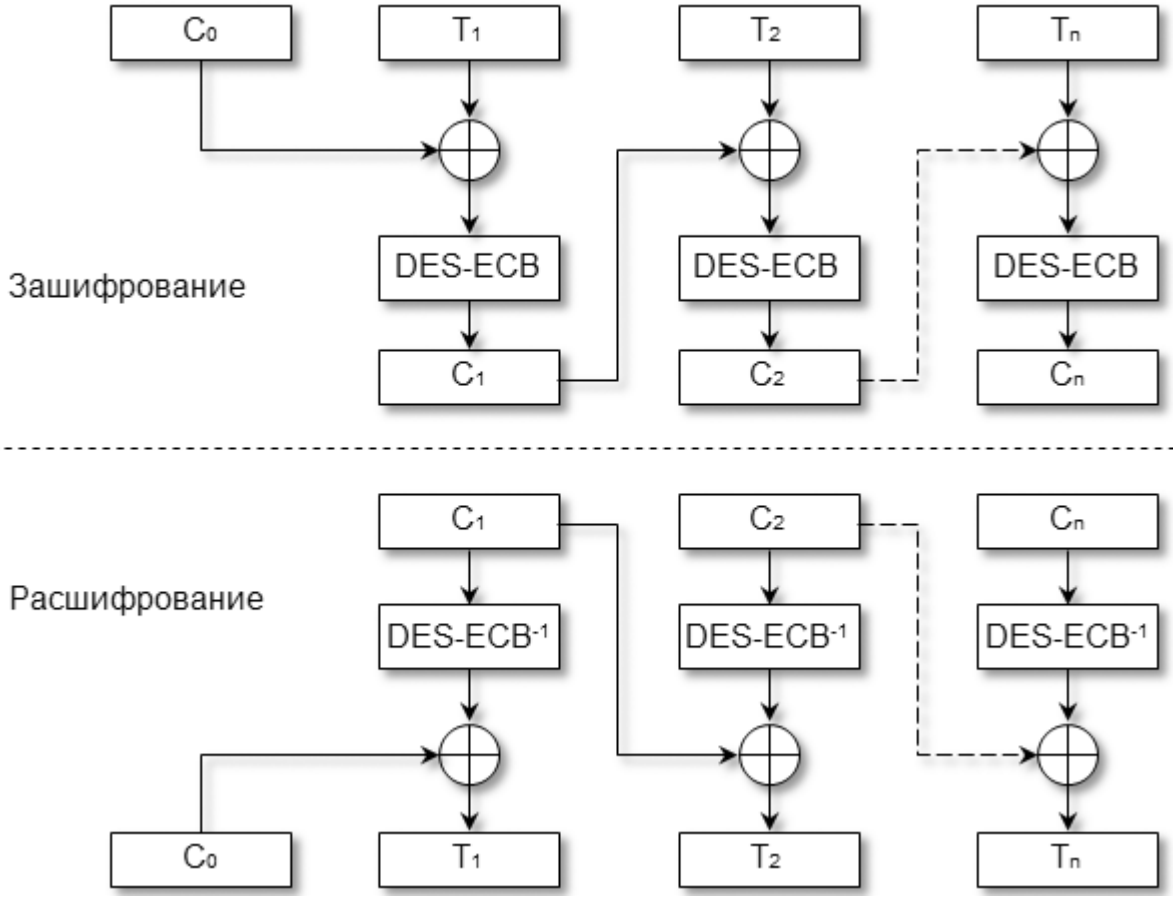


Рис.8. Схемы алгоритмов DES в режиме сцепления блоков шифра (CBC)

1. Исходное сообщение разбивается на 64-битовые блоки: $\mathbf{T_1, T_2, ..., T_n}$.

2. Первый блок $\mathbf{T_1}$ складывается по модулю 2 с 64-битовым начальным вектором $\mathbf{C_0}$, который меняется и держится в секрете ($\mathbf{C_0}$ - **синхропосылка**, по сути, второй ключ).

3. Полученная сумма затем шифруется с использованием ключа DES.

4. 64-битовый шифр $\mathbf{C_1}$ складывается по модулю 2 со вторым блоком текста, ре¬зультат шифруется и получается второй 64-битовый шифр $\mathbf{C_2}$, и т.д.

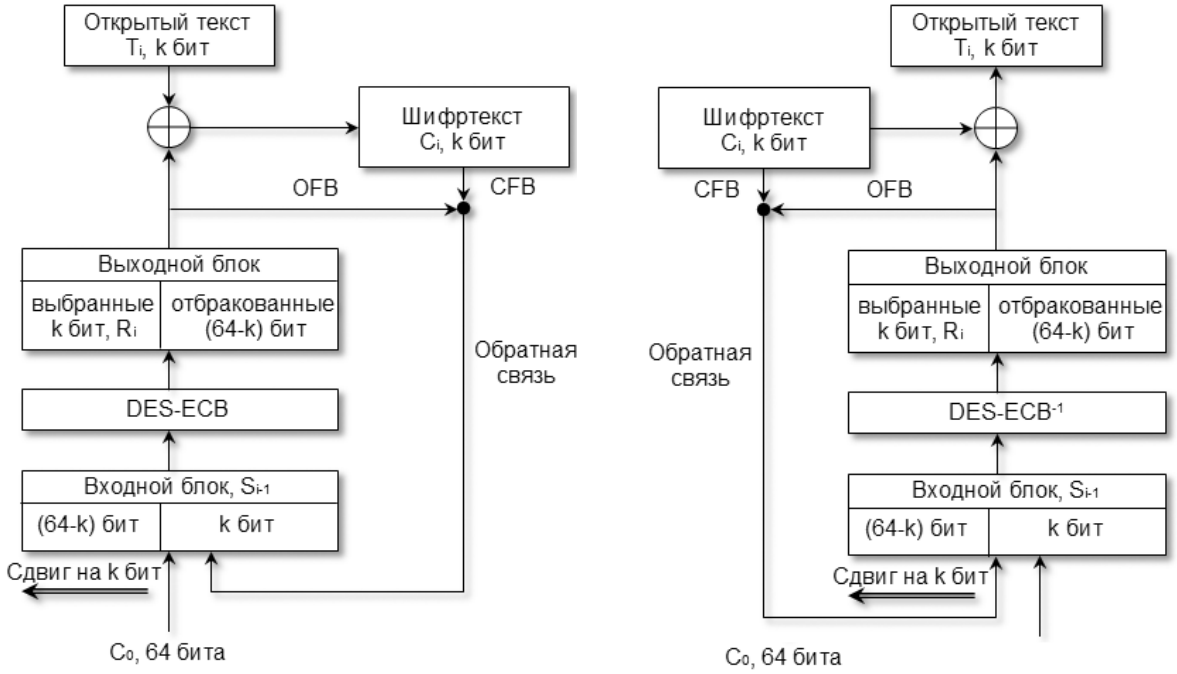
5. Процедура повторяется до тех пор, пока не будут обработаны все блоки текста. Таким образом, для всех $\mathbf{i} = 1...n$ ($\mathbf{n}$ - число блоков) результат шифрования $\mathbf{C_i}$ определяется следующим образом: $\mathbf{C_i = DES(T_i \oplus C_{i-1})}$.

Очевидно, что последний 64-битовый блок шифртекста является функцией секретного ключа, начального вектора и каждого бита открытого текста независимо от его длины. Этот блок шифртекста называют **кодом аутентификации сообщения** (**КАС**). Код КАС может быть легко проверен получателем, владеющим секретным ключом и начальным вектором, путем повторения процедуры, выполненной отправителем. Достоинство данного режима в том, что он не позволяет накапливаться ошибкам при передаче. Блок $\mathbf{T_i}$ является

функцией только C_{i-1} и C_i. Поэтому ошибка при передаче приведет к потере только двух блоков исходного текста.

Режимы CFB и OFB (обратная связь по шифртексту - Cipher Feed Back; обратная связь по выходу - Output Feed Back).

В алгоритмах CFB и OFB используется такой же подход, что и в алгоритме CBC, когда результат шифрования (расшифрования) блока на предыдущем шаге используется для шифровки (расшифровки) очередного блока. В этих режимах размер входного блока может быть меньше 64 бит.



Зашифрование Расшифрование

Рис.9. Схемы алгоритмов DES в режимах CFB и OFB

Алгоритм шифрования.

1. Исходное сообщение разбивается на блоки, длиной **k** бит: $\mathbf{T_1, T_2, ..., T_n}$.

2. Синхропосылка шифруется с использованием ключа DES.

3. Из полученного результата выбираются старшие **k** бит, которые складываются с блоком открытого текста $\mathbf{T_i}$ по модулю 2 для получения шифртекста $\mathbf{C_i}$.

4. Процедура повторяется до тех пор, пока не будут обработаны все блоки текста. При этом, содержимое входного блока сдвигается на **k** бит влево (причем сдвиг не циклический) и младшие **k** бит заполняются либо шифртекстом (CFB), либо старшими **k** бит выходного блока (OFB).

Алгоритм расшифрования аналогичен алгоритму шифрования и отличается только входом и выходом стрелок в блок сложения по модулю 2.

Тройной DES.

При тройном DES текст шифруется 3 раза DES. Таким образом, длина ключа возрастает до 168-бит (56x3). Однако, применение тройного DES не всегда означает увеличение уровня безопасности сообщения.

Типы тройного шифрования DES:

- DES-EEE3: Шифруется 3 раза с 3 различными ключами.

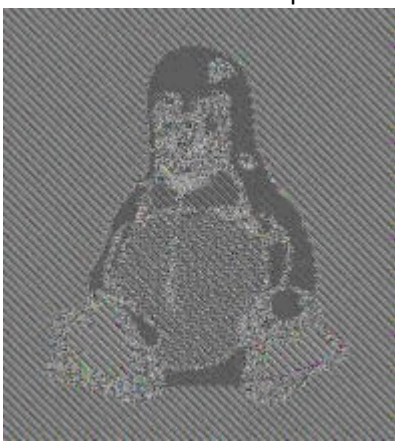
- DES-EDE3: 3 DES операции шифровка-расшифровка-шифровка с 3 различными ключами.

- DES-EEE2 и DES-EDE2: Как и предыдущие, за исключением того, что первая и третья операции используют одинаковый ключ.

Сферы применения разных режимов DES.

Электронная кодовая книга ECB рекомендуется использовать для шифрования сообщений длиной в один блок (как правило, ключей). Недостаток ECB, в сравнении с другими режимами шифрования - сохранение статистических особенностей открытого текста.

оригинальная битовая карта изображения



криптограмма в режиме ECB (сохранились статистические особенности)

другие режимы шифрования (псевдослучайная последовательность)

Рис.10. Сохранение статистических особенностей открытого текста

Сцепление блоков шифра CBC и тройной DES могут использоваться для шифрование больших объемов данных (в частности тройной DES используется для эмиссии и обработки кредитных карт VISA, EuroPay и других платежных систем).

Обратная связь по шифртексту CFB и обратная связь по выходу OFB рекомендуется для шифрования отдельных символов или небольших сообщений.

Вопросы для самоконтроля

1. Методы защиты данных.
2. Криптография.
3. Криптоанализ.
4. Криптология.
5. Криптостойкость.
6. Электронная цифровая подпись.

Практическое занятие №4. Шифрование с открытым ключом

Цель: изучить способы шифрования данных с открытым ключом.

Ход работы:

Необходимо ознакомиться с теоретической частью и зашифровать свою фамилию с помощью следующих шифров:

- алгоритма RSA;
- алгоритма на основе задачи об укладке ранца;
- алгоритма шифрования Эль-Гамаля.

При выполнении задания необходимо привести исходное сообщение (фамилию) и таблицы генерации ключей, шифрования и расшифрования.

Для первого и третьего способов принять код символа в соответствии с его положением в алфавите, для второго – с кодировкой Windows 1251.

Для алгоритма шифрования Эль-Гамаля случайные числа **k** для каждой буквы исходного сообщения должны быть разными.

Основы шифрования

Главная проблема использования одноключевых (симметричных) криптосистем заключается в распределении ключей. Для того, чтобы был возможен обмен информацией между двумя сторонами, ключ должен быть сгенерирован одной из них, а затем в конфиденциальном порядке передан другой. Особую остроту данная проблема приобрела в наши дни, когда криптография стала общедоступной, вследствие чего количество пользователей больших криптосистем может исчисляться сотнями и тысячами.

Начало асимметричным шифрам было положено в работе «Новые направления в современной криптографии» Уитфилда Диффи и Мартина Хеллмана, опубликованной в 1976 г. Находясь под влиянием работы Ральфа Меркла (Ralph Merkle) о распространении открытого ключа, они предложили метод получения секретных ключей для симметричного шифрования, используя открытый канал. В 2002 г. Хеллман предложил называть данный алгоритм «Диффи - Хеллмана - Меркла», признавая вклад Меркла в изобретение криптографии с открытым ключом.

Хотя работа Диффи-Хеллмана создала большой теоретический задел для открытой криптографии, первой реальной криптосистемой с открытым ключом считают алгоритм RSA (названный по имени авторов - Рон Ривест (Ronald Linn Rivest), Ади Шамир (Adi Shamir) и Леонард Адлеман (Leonard Adleman)).

Справедливости ради следует отметить, что в декабре 1997 г. была обнародована информация, согласно которой британский математик Клиффорд Кокс (Clifford Cocks), работавший в центре правительственной связи (GCHQ) Великобритании, описал систему, аналогичную RSA, в 1973 г., а несколькими месяцами позже в 1974 г. Малькольм Вильямсон изобрел математический алгоритм, аналогичный алгоритму Диффи – Хеллмана - Меркла.

Суть шифрования с открытым ключом заключается в том, что для шифрования данных используется один ключ, а для расшифрования другой (поэтому такие системы часто называют **асимметричными**).

Основная предпосылка, которая привела к появлению шифрования с открытым ключом, заключалось в том, что отправитель сообщения (тот, кто зашифровывает сообщение), не обязательно должен быть способен его расшифровывать. Т.е. даже имея исходное сообщение, ключ, с помощью которого оно шифровалось, и зная алгоритм шифрования, он не может расшифровать закрытое сообщение без знания ключа расшифрования.

Первый ключ, которым шифруется исходное сообщение, называется **открытым** и может быть опубликован для использования всеми пользователями системы. Расшифрование с помощью этого ключа невозможно. Второй ключ, с помощью которого дешифруется сообщение, называется **закрытым** и должен быть известен только законному получателю закрытого сообщения.

Алгоритмы шифрования с открытым ключом используют так называемые необратимые или односторонние функции. Эти функции обладают следующим свойством: при заданном значении аргумента **x** относительно просто вычислить значение функции **f(x)**, однако, если известно значение функции **y = f(x)**, то нет простого пути для вычисления значения аргумента **x**. Например, функция **SIN**. Зная x, легко найти значение **SIN(x)** (например, x = **π**, тогда SIN(**π**) = 0). Однако, если SIN(x) = 0, однозначно определить **x** нельзя, т.к. в этом случае **x** может быть любым числом, определяемым по формуле **i * π**, где **i** – целое число.

Однако не всякая необратимая функция годится для использования в реальных криптосистемах. В их числе и функция **SIN**. Следует также отметить, что в самом определении необратимости функции присутствует неопределенность. Под необратимостью понимается не теоретическая необратимость, а практическая невозможность вычислить обратное значение, используя современные вычислительные средства за обозримый интервал времени.

Поэтому чтобы гарантировать надежную защиту информации, к криптосистемам с открытым ключом предъявляются два важных и очевидных **требования**.

1. Преобразование исходного текста должно быть условно необратимым и исключать его восстановление на основе открытого ключа.

2. Определение закрытого ключа на основе открытого также должно быть невозможным на современном технологическом уровне.

Все предлагаемые сегодня криптосистемы с открытым ключом опираются на один из следующих **типов односторонних преобразований**.

1. Разложение больших чисел на простые множители (алгоритм RSA).

2. Вычисление дискретного логарифма или дискретное возведение в степень (алгоритм Диффи-Хелмана-Меркла, схема Эль-Гамаля).

3. Задача об укладке рюкзака (ранца) (авторы Хелман и Меркл).

4. Вычисление корней алгебраических уравнений.

5. Использование конечных автоматов (автор Тао Ренжи).

6. Использование кодовых конструкций.

7. Использование свойств эллиптических кривых.

По особенностям процедур преобразования сообщения ассиметричные шифры относятся к шифрам замены и делятся на два **типа**:

- детерминированные - при зашифровании одного и того же открытого сообщения одним и тем же открытым ключом всегда будет получаться одна и та же шифрограмма. Т.е. для заданного открытого ключа один и тот же символ (блок символов) открытого сообщения всегда будет представляться одной и той же шифрозаменой. Данный тип шифров относится к регулярным шифрам однозначной замены;

- вероятностные - при зашифровании одного и того же открытого сообщения одним и тем же открытым ключом могут получаться разные шифрограммы. Т.е. для заданного открытого ключа один и тот же символ (блок символов) открытого сообщения может представляться разными шифрозаменами. Это достигается за счет использования случайной величины при зашифровании символа (блока символов), что эквивалентно переключению алфавитов шифрозамен. Данный тип шифров относится к полиалфавитным шифрам многозначной замены.

Алгоритм RSA

Стойкость RSA основывается на большой вычислительной сложности известных алгоритмов разложения числа на простые сомножители (делители). Например, легко найти произведение двух простых чисел 7 и 13 даже в уме – 91. Попробуйте в уме найти два простых числа, произведение которых равно 323 (числа 17 и 19). Конечно, для современной вычислительной техники найти два простых числа, произведение которых равно 323, не проблема. Поэтому для надежного шифрования алгоритмом RSA, как правило, выбираются простые числа, количество двоичных разрядов которых равно нескольким сотням.

В августе 1977 г. знаменитый американский писатель и популяризатор науки Мартин Гарднер озаглавил свою колонку по занимательной математике в журнале Scientific American так: «Новый вид шифра, на расшифровку которого потребуются миллионы лет». После объяснения принципа системы шифрования с открытым ключом он показал само зашифрованное сообщение и открытый ключ N, используемый в этом шифре:

N = 114 381 625 757 888 867 669 235 779 976 146 612 010 218 296 721 242 362 562 561 842 935 706 935 245 733 897 830 597 123 563 958 705 058 989 075 147 599 290 026 879 543 541.

Гарднер призвал читателей попробовать расшифровать сообщение, используя предоставленную информацию, и даже дал подсказку: для решения необходимо разложить число N на простые множители p и q. Более того, Гарднер назначил приз в размере $100 (приличная сумма на тот момент) тому, кто первым получит правильный ответ. Каждый, кто захочет побольше узнать о шифре, писал Гарднер, может обратиться к создателям шифра - Рону Ривесту, Ади Шамиру и Леонарду Адлеману из Лаборатории информации Массачусетского технологического института.

Правильный ответ был получен лишь через 17 лет. Он стал результатом сотрудничества более чем 600 человек. Ключами оказались p = 32 769 132 993 266

709 549 961 988 190 834 461 413 177 642 967 992 942 539 798 288 533 и q = 3 490 529 510 847 650 949 147 849 619 903 898 133 417 764 638 493 387 843 990 820 577, а зашифрованная фраза звучала так: «Волшебные слова - это брезгливый ягнятник».

Авторы RSA поддерживали идею её активного распространения. В свою очередь, Агентство национальной безопасности (США), опасаясь использования этого алгоритма в негосударственных структурах, на протяжении нескольких лет безуспешно требовало прекращения распространения системы. Ситуация порой доходила до абсурда. Например, когда программист Адам Бек (Adam Back) описал на языке Perl алгоритм RSA, состоящий из пяти строк, правительство США запретило распространение этой программы за пределами страны. Люди, недовольные подобным ограничением, в знак протеста напечатали текст этой программы на своих футболках.

Первым этапом любого асимметричного алгоритма является создание получателем шифрограмм пары ключей: открытого и секретного. Для алгоритма RSA этап создания ключей состоит из следующих операций.

Таблица 1. Процедура создания ключей

№ п/п	Описание операции	Пример
1	Выбираются два простых числа[1] **p** и **q**.	p = 7, q = 13
2	Вычисляется произведение **n** = p * q.	n = 91
3	Вычисляется **функция Эйлера**[2] **φ(n)**.	φ(n) = (7-1)(13-1) = 91-7-13+1 = 72
4	Выбирается открытый ключ **e** - произвольное натуральное число (0 < e < n), взаимно простое[3] с результатом функции Эйлера (e ⊥ φ(n)).	e = 5
5	Вычисляется закрытый ключ **d** - обратное число[4] к **e** по модулю φ(n) из соотношения (d * e) mod φ(n) = 1.	d = 29 [(29*5) mod 72 = 1]
6	Публикуется открытый ключ {**e**, **n**} в специальном хранилище, где исключается возможность его подмены (**общедоступном сертифицированном справочнике**).	

Примечания.

1) **Простое число** - натуральное число, большее единицы и не имеющее других натуральных делителей, кроме самого себя и единицы.

2) Результат расчета **функции Эйлера φ(n)** равен количеству положительных чисел, не превосходящих **n** и взаимно простых с **n**. Некоторые случаи и способы расчета функции Эйлера приведены в следующей таблице.

Таблица 2. Способы расчета функции Эйлера

Расчетный случай	Формула	Пример (число / расчетная формула / список взаимно простых чисел)
n простое число	$\varphi(n) = n - 1$	n = 7 $\varphi(7) = 7 - 1 = 6$ {1, 2, 3, 4, 5, 6}
n = p q произведение двух простых чисел	$\varphi(n) = \varphi(p)\,\varphi(q) = (p - 1)(q - 1) = n - p - q + 1$ (за исключением случая p = q = 2)	n = 15 = 3 * 5 $\varphi(15) = \varphi(3)\,\varphi(5) = (3 - 1)(5 - 1) = 15 - 3 - 5 + 1 = 8$ {1, 2, 4, 7, 8, 11, 13, 14}
$n = p^q$ простое число в степени	$\varphi(n) = p^q - p^{q-1}$	n = 9 = 3^2 $\varphi(9) = 3^2 - 3^{2-1} = 9 - 3 = 6$ {1, 2, 4, 5, 7, 8}
$n = p_1^{q1}\, p_2^{q1} \dots p_k^{qk}$ разложение числа согласно основной теоремы арифметики (общий случай)	$\varphi(n) = \varphi(p_1^{q1})\varphi(p_2^{q2}) \dots \varphi(p_k^{qk}) = p_1^{q1}\left(1 - \frac{1}{p_1}\right) p_2^{q2}\left(1 - \frac{1}{p_2}\right) \dots p_k^{qk}\left(1 - \frac{1}{p_k}\right) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right) \dots \left(1 - \frac{1}{p_k}\right)$	n = 84 = $2^2\, 3^1\, 7^1$ $\varphi(84) = 84\left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{3}\right)\left(1 - \frac{1}{7}\right) =$ {1, 5, 11, 13, 17, 19, 23, 25, 29, 31, 37, 41, 43, 47, 53, 55, 59, 61, 65, 67, 71, 73, 79, 83}

3) **Взаимно простые числа** – числа, не имеющие общих делителей, кроме 1, т.е. наибольший общий делитель которых равен 1.

4) **Обратными числами по модулю m** называются такие числа **n** и $\mathbf{n^{-1}}$, для которых справедливо выражение $\mathbf{(n * n^{-1}) \bmod m = 1}$. Для вычисления обратных чисел по модулю обычно используется расширенный алгоритм Евклида. В безмодульной математике **обратное число** $\mathbf{n^{-1}}$ (обратное значение, обратная величина) - число, на которое надо умножить данное число **n**, чтобы получить единицу ($\mathbf{n * n^{-1} = 1}$). Пара чисел, произведение которых равно единице, называются **взаимно обратными**. Например: 5 и 1/5, -6/7 и -7/6.

Процедуры шифрования и дешифрования выполняются по следующим формулам

$$C = T^e \bmod n, \qquad (1)$$

$$T = C^d \bmod n. \qquad (2)$$

где T, C - числовые эквиваленты символов открытого и шифрованного сообщения.

Пример шифрования по алгоритму RSA приведен в следующей таблице. Коды букв соответствуют их положению в русском алфавите (начиная с 1).

Таблица 3. Пример шифрования по алгоритму RSA

Открытое сообщение, T	Символ	А	Б	Р	А	М	О	В
	Код	1	2	18	1	14	16	3
Шифрограмма, $C = T^5 \bmod 91$		1	32	44	1	14	74	61
Открытое сообщение, $T = C^{29} \bmod 91$		1	2	18	1	14	16	3

Следует отметить, что **p** и **q** выбираются таким образом, чтобы **n** было больше кода любого символа открытого сообщения. В автоматизированных системах исходное сообщение переводиться в двоичное представление, после чего шифрование выполняется над блоками бит равной длины. При этом длина блока должна быть меньше, чем длина двоичного представления **n**.

Алгоритм RSA требует много машинного времени и очень мощных процессоров. До 1980-х гг. только правительства, армия и крупные предприятия имели достаточно мощные компьютеры для работы с RSA. В результате у них была фактически монополия на эффективное шифрование. Летом 1991 г. Филипп Циммерман, американский физик и борец за сохранение конфиденциальности, предложил бесплатную систему шифрования PGP (англ. Pretty Good Privacy - «достаточно хорошая степень конфиденциальности»), алгоритм которой мог работать на домашних компьютерах. PGP использует классическое симметричное шифрование, что и обеспечивает ей большую скорость на домашних компьютерах, но она шифрует ключи по асимметричному алгоритму RSA.

Циммерман объяснил причины этой меры в открытом письме, которое заслуживает быть процитированным здесь, по крайней мере, частично из-за пророческого описания того, как мы живем, работаем и общаемся два десятилетия спустя.

Это личное. Это конфиденциальное. И это только ваше дело и ничье другое. Вы можете планировать политическую кампанию, обсуждать ваши налоги или иметь тайную любовную связь. Или вы можете заниматься тем, что вам не кажется незаконным, хотя таковым является. Что бы то ни было, вы не хотите, чтобы ваши личные электронные письма или конфиденциальные документы были прочитаны кем-то еще. Нет ничего плохого в том, чтобы охранять вашу частную жизнь. Частная жизнь неприкосновенна, как Конституция...

Мы движемся к будущему, где мир будет опутан волоконно-оптическими сетями высокой емкости, связывающими наши повсеместно распространенные персональные компьютеры. Электронная почта станет нормой для всех, а не новинкой, как сегодня. Правительство будет защищать наши электронные сообщения государственными протоколами шифрования. Наверное, большинство людей примет это. Но, возможно, некоторые захотят иметь свои собственные защитные меры... Если конфиденциальность признать вне закона, только люди вне закона будут ею обладать.

Спецслужбы обладают лучшими криптографическими технологиями. Как и торговцы оружием и наркотиками. Как и военные подрядчики, нефтяные компании и другие корпорации-гиганты. Но обычные люди и общественные организации практически не имеют недорогих защитных криптографических технологий с открытым ключом. До сих пор не имели.

PGP дает людям возможность самим защищать свою конфиденциальность. Сегодня существует растущая социальная потребность в этом. Вот почему я написал PGP.

В заключении следует отметить стойкость данного алгоритма. В 2003 г. Ади Шамир и Эран Тромер разработали схему устройства TWIRL, которое при стоимости $ 10 000 может дешифровать 512-битный ключ за 10 минут, а при стоимости $ 10 000 000 – 1024-битный ключ меньше, чем за год. В настоящее время Лаборатория RSA рекомендует использовать ключи (параметр **n**) размером 2048 битов.

Алгоритм на основе задачи об укладке ранца

В 1978 г. Меркл и Хеллман предложили использовать задачу об укладке ранца (рюкзака) для асимметричного шифрования. Она относится к классу NP-полных задач и формулируется следующим образом. Дано множество предметов различного веса. Спрашивается, можно ли положить некоторые из этих предметов в ранец так, чтобы его вес стал равен определенному значению? Более формально задача формулируется так: дан набор значений $\mathbf{M_1, M_2, ..., M_n}$ и суммарное значение **S**; требуется вычислить значения $\mathbf{b_i}$ такие что

$$S = b_1M_1 + b_2M_2 + ... + b_nM_n, \quad (3)$$

где n – количество предметов; b_i - бинарный множитель. Значение $b_i = 1$ означает, что предмет i кладут в рюкзак, $b_i = 0$ - не кладут.

Например, веса предметов имеют значения 1, 5, 6, 11, 14, 20, 32 и 43. При этом можно упаковать рюкзак так, чтобы его вес стал равен 22, использовав предметы весом 5, 6 и 11. Невозможно упаковать рюкзак так, чтобы его вес стал равен 24.

В основе алгоритма, предложенного Мерклом и Хеллманом, лежит идея шифрования сообщения на основе решения серии задач укладки ранца. Предметы из кучи выбираются с помощью блока открытого текста, длина которого (в битах) равна количеству предметов в куче. При этом биты открытого текста соответствуют значениям **b**, а текст является полученным суммарным весом. Пример шифрограммы, полученной с помощью задачи об укладке ранца, показан в следующей таблице.

Таблица 4. Пример шифрования на основе задачи об укладке ранца

Открытый текст	1	1	1	0	0	1	0	0	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0
Рюкзак (ключ)	1	5	6	11	14	20	32	43	1	5	6	11	14	20	32	43	1	5	6	11	14	20	32	43
Шифрограмма	32 (1+5+6+20)								73 (5+11+14+43)								0							

Суть использования данного подхода для шифрования состоит в том, что на самом деле существуют две различные задачи укладки ранца - одна из них решается легко и характеризуется линейным ростом трудоемкости, а другая, как принято считать, нет. Легкий для укладки ранец можно превратить в трудный. Раз так, то можно применить в качестве открытого ключа **трудный** для укладки ранец, который легко использовать для шифрования, но невозможно - для дешифрования. А в качестве закрытого ключа применить **легкий** для укладки ранец, который предоставляет простой способ дешифрования сообщения.

В качестве закрытого ключа (легкого для укладки ранца) используется сверхвозрастающая
последовательность. **Сверхвозрастающей** называется **последовательность**, в которой каждый последующий член больше суммы всех предыдущих. Например, последовательность {2, 3, 6, 13, 27, 52, 105, 210} является сверхвозрастающей, а {1, 3, 4, 9, 15, 25, 48, 76} - нет.

Решение для сверхвозрастающего ранца найти легко. В качестве текущего выбирается полный вес, который надо получить, и сравнивается с весом самого тяжелого предмета в ранце. Если текущий вес меньше веса данного предмета, то его в рюкзак не кладут, в противном случае его укладывают в рюкзак. Уменьшают текущий вес на вес положенного предмета и переходят к следующему по весу предмету в последовательности. Шаги повторяются до тех пор, пока процесс не закончится. Если текущий вес уменьшится до нуля, то решение найдено. В противном случае, нет.

Например, пусть полный вес рюкзака равен 270, а последовательность весов предметов равна {2, 3, 6, 13, 27, 52, 105, 210}. Самый большой вес – 210. Он меньше 270, поэтому предмет весом 210 кладут в рюкзак. Вычитают 210 из 270 и получают 60. Следующий наибольший вес последовательности равен 105. Он больше 60, поэтому предмет весом 105 в рюкзак не кладут. Следующий самый тяжелый предмет имеет вес 52. Он меньше 60, поэтому предмет весом 52 также кладут в рюкзак. Аналогично проходят процедуру укладки в рюкзак предметы весом 6 и 2. В результате полный вес уменьшится до 0. Если бы этот рюкзак был бы использован для дешифрования, то открытый текст, полученный из значения шифртекста 270, был бы равен 10100101.

Открытый ключ представляет собой не сверхвозрастающую (нормальную) последовательность. Он формируется на основе закрытого ключа и, как принято считать, не позволяет легко решить задачу об укладке ранца. Для его получения все значения закрытого ключа умножаются на число **n** по модулю **m**. Значение модуля **m** должно быть больше суммы всех чисел последовательности, например, 420 (2+3+6+13+27+52+105+210 = 418). Множитель **n** должен быть взаимно простым числом с модулем **m**, например, 31. Результат построения нормальной последовательности (открытого ключа) представлен в следующей таблице.

Таблица 5. Пример получения открытого ключа

Закрытый ключ, k_i	2	3	6	13	27	52	105	210
Открытый ключ, $(k_i*n) \bmod m = (k_i*31) \bmod 420$	62	93	186	403	417	352	315	210

Для шифрования сообщение сначала разбивается на блоки, по размерам равные числу элементов последовательности в рюкзаке. Затем, считая, что единица указывает на присутствие элемента последовательности в рюкзаке, а ноль - на его отсутствие, вычисляются полные веса рюкзаков – по одному рюкзаку для каждого блока сообщения.

В качестве примера возьмем открытое сообщение «АБРАМОВ», символы которого представим в бинарном виде в соответствии с таблицей кодов символов Windows 1251. Результат шифрования с помощью открытого ключа {62, 93, 186, 403, 417, 352, 315, 210} представлен в следующей таблице.

Таблица 6. Пример шифрования

Открытое сообщение		Сумма весов	Шифрограмма (рюкзак), c_i
Символ	Bin-код		
А	1100 0000	62+93	155
Б	1100 0001	62+93+210	365
Р	1101 0000	62+93+403	558
А	1100 0000	62+93	155
М	1100 1100	62+93+417+352	924
О	1100 1110	62+93+417+352+315	1239
В	1100 0010	62+93+315	470

Для расшифрования сообщения получатель должен сначала определить обратное число $\mathbf{n^{-1}}$, такое что $\mathbf{(n * n^{-1}) \bmod m = 1}$. После определения обратного числа каждое значение шифрограммы умножается на $\mathbf{n^{-1}}$ по модулю **m** и с помощью закрытого ключа определяются биты открытого текста.

В нашем примере сверхвозрастающая последовательность равна {2, 3, 6, 13, 27, 52, 105, 210}, m = 420, n = 31. Значение n^{-1} равно 271 (31*271 mod 420 = 1).

Таблица 7. Пример расшифрования

Шифрограмма (рюкзак), c_i	$(c_i*n^{-1}) \bmod m = (c_i*271) \bmod 420$	Сумма весов	Открытое сообщение	
			Символ	Bin-код
155	5	2+3	1100 0000	А
365	215	2+3+210	1100 0001	Б
558	18	2+3+13	1101 0000	Р
155	5	2+3	1100 0000	А
924	84	2+3+27+52	1100 1100	М

1239	189	2+3+27+52+105	1100 1110	O
470	110	2+3+105	1100 0010	B

В своей работе авторы рекомендовали брать длину ключа, равную 100 (количество элементов последовательности). В заключении следует отметить, что задача вскрытия данного способа шифрования успешно решена Шамиром и Циппелом в 1982 г.

Вероятностное шифрование

Авторами идеи и первого алгоритма вероятностного шифрования являются Шафи Гольдвассер (Shafi Goldwasser) и Сильвио Микали (Silvio Micali). Для данного типа шифров при шифровании одного и того же исходного сообщения с помощью одного и того же открытого ключа $\mathbf{k_1}$ можно получить разные шифртексты, которые при расшифровке закрытым ключом $\mathbf{k_2}$ дают изначальное исходное сообщение.

$C_1 = E_{k1}(T), C_2 = E_{k1}(T), C_3 = E_{k1}(T), ..., C_N = E_{k1}(T),$ (4)

$T = D_{k2}(C_1) = D_{k2}(C_2) = D_{k2}(C_3) = ... = D_{k2}(C_N).$ (5)

Такая особенность вероятностного шифрования делает бессмысленными атаки на шифр "с известным открытым текстом" и "с выбором открытого текста".

Ниже рассматриваются два алгоритма вероятностного шифрования:

- алгоритм шифрования Эль-Гамаля;
- алгоритм на основе эллиптических кривых.

Алгоритм шифрования Эль-Гамаля

Схема была предложена Тахером Эль-Гамалем в 1984 г. Он усовершенствовал систему Диффи-Хеллмана и получил два алгоритма, которые использовались для шифрования и обеспечения аутентификации. Стойкость данного алгоритма базируется на сложности решения задачи дискретного логарифмирования.

Суть задачи заключается в следующем. Имеется уравнение

$g^x \bmod p = y.$ (6)

Требуется по известным **g**, **y** и **p** найти натуральное (целое положительное) число **x** (**дискретный логарифм**).

Порядок создания ключей приводится в следующей таблице.

Таблица 8. Процедура создания ключей

№ п/п	Описание операции	Пример
1	Выбирается простое число **p**.	p = 37
2	Выбирается число **g** (0 < g < p), являющееся первообразным корнем по модулю **p**[*)].	g = 2
3	Выбирается закрытый ключ **x** (дискретный логарифм) -	x = 5

	произвольное натуральное число ($0 < x < p$).	
4	Вычисляется **y** = $g^x \bmod p$.	$y = 2^5 \bmod 37 = 32 \bmod 37 = 32$
5	Публикуются открытый ключ {**y**, **g**, **p**} в специальном хранилище, где исключается возможность его подмены (**общедоступном сертифицированном справочнике**). Параметры **g** и **p** можно сделать общими для группы пользователей.	

*) **Первообразный (примитивный) корень по модулю p** – натуральное число g такое, что

$g^{\varphi(p)} \bmod p = 1$

и

$g^i \bmod p \neq 1$, для $1 \leq i < \varphi(p)$

где $\varphi(p)$ – функция Эйлера (т.к. p – простое число, то $\varphi(p) = p - 1$).

Проверка. При g = 2 и p = 37, $\varphi(37) = 37 - 1 = 36$.

$2^{36} \bmod 37 = 1$;

$2^1 \bmod 37 = 2 (\neq 1)$;

$2^2 \bmod 37 = 4 (\neq 1)$;

$2^3 \bmod 37 = 8 (\neq 1)$;

$2^4 \bmod 37 = 16 (\neq 1)$;

...;

$2^{34} \bmod 37 = 28 (\neq 1)$;

$2^{35} \bmod 37 = 19 (\neq 1)$.

Для шифрования каждого отдельного блока исходного сообщения должно выбираться случайное число **k** ($1 < k < p - 1$). После чего шифрограмма генерируется по следующим формулам:

$a = g^k \bmod p$, (7)

$b = (y^k T) \bmod p$, (8)

где T – исходное сообщение; (a, b) – зашифрованное сообщение.

Дешифрование сообщения выполняется по следующей формуле:

$T = (b (a^x)^{-1}) \bmod p$ (9)

или

$T = (b a^{p-1-x}) \bmod p$, (10)

где $(a^x)^{-1}$ – обратное значение числа a^x по модулю p.

Пример шифрования и дешифрования по алгоритму Эль-Гамаля при k = 7 приведен в таблице, хотя для шифрования каждого блока (в нашем случае буквы) исходного сообщения надо использовать свое случайное число **k**.

Первая часть шифрованного сообщения – $a = 2^7 \bmod 37 = 17$.

$a^x = 17^5 = 1419857$, $(a^x)^{-1} = 2$ ($1419857 * 2 \bmod 37 = 1$) или $a^{p-1-x} = 17^{37-1-5} \approx 1.3928892 * 10^{38}$.

Таблица 9. Пример шифрования по алгоритму Эль-Гамаля (при k = const)

Открытое сообщение, T	Символ	А	Б	Р	А	М	О	В
	Код	1	2	18	1	14	16	3
Шифрование	Случайное число k	7	7	7	7	7	7	7
	Первая часть шифрограммы, $a = 2^k \bmod 37$	17	17	17	17	17	17	17
	Вторая часть шифрограммы, $b = (32^k * T) \bmod 37$	19	1	9	19	7	8	20
Расшифрование	Обратное значение числа a^x по модулю p, $(a^x)^{-1}$	2	2	2	2	2	2	2
	Открытое сообщение, $T = (b * (a^x)^{-1}) \bmod 37$	1	2	18	1	14	16	3

Ввиду того, что число **k** является случайным, то такую схему еще называют схемой **вероятностного шифрования**. Вероятностный характер шифрования является преимуществом для схемы Эль-Гамаля, т.к. у схем вероятностного шифрования наблюдается большая стойкость по сравнению со схемами с определенным процессом шифрования. Недостатком схемы шифрования Эль-Гамаля является удвоение длины зашифрованного текста по сравнению с начальным текстом. Для схемы вероятностного шифрования само сообщение **T** и ключ не определяют шифртекст однозначно. В схеме Эль-Гамаля необходимо использовать различные значения случайной величины **k** для шифровки различных сообщений **T** и **T'**. Если использовать одинаковые **k**, то для соответствующих шифртекстов (**a**, **b**) и (**a'**, **b'**) выполняется соотношение $\mathbf{b\,(b')^{-1} \equiv T\,(T')^{-1}\ (mod\ p)}$. Из этого выражения можно легко вычислить **T**, если известно **T'**.

Пример. Предположим злоумышленник перехватил зашифрованное сообщение $\mathbf{C = ((a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n))}$, для которого использовалось одно и тоже случайное число **k**. Он знает один из блоков $\mathbf{T_i = E_{k1}(C_i)}$ или при известном открытом ключе (**y**, **g**, **p**) ему удалось подобрать **k'**, которое совпало с используемым при шифровании **k**. Например по второму варианту, для шифрования символа **X** (T' = 22) злоумышленник использовал k' = 7 (равное k). Тогда, $b(X) = b' = (32^7 * 22) \bmod 37 = 11$, $(b')^{-1} = 27$, $(T')^{-1} = 32$. Расшифрование перехваченного сообщения приведено в следующей таблице.

Таблица 10. Пример расшифрования перехваченного сообщения

Вторая часть перехваченной шифрограммы, b		19	1	9	19	7	8	20
$L = (b * (b')^{-1}) \bmod p = (b * 27) \bmod 37$		32	27	21	32	4	31	22
Вскрытое открытое сообщение, T	Код, определяемый по уравнению $(T * (T')^{-1}) \bmod p = L$ $(T * 32) \bmod 37 = L$	1	2	18	1	14	16	3
	Символ	А	Б	Р	А	М	О	В

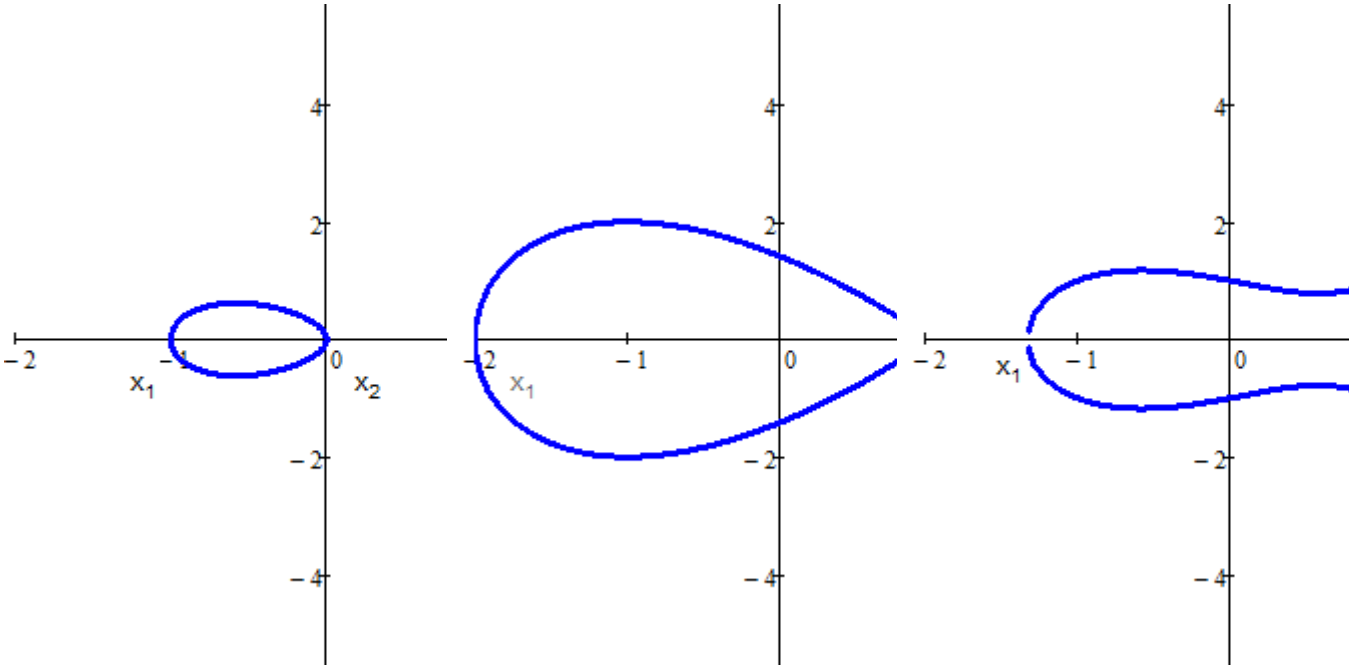
Алгоритм на основе эллиптических кривых

Использование эллиптических кривых для создания криптосистем было независимо предложено Нилом Коблицем (Neal Koblitz) и Виктором Миллером (Victor Miller) в 1985 г. При использовании алгоритмов на эллиптических кривых полагается, что не существует быстрых алгоритмов для решения задачи дискретного логарифмирования в группах их точек. В настоящий момент известны лишь экспоненциальные алгоритмы вычисления обратных функций для эллиптических кривых. По сравнению с субэкспоненциальными алгоритмами разложения числа на простые сомножители (см. криптосистему RSA), это позволяет при одинаковом уровне стойкости уменьшить размерность ключа в несколько раз, а, следовательно, упростить программную и аппаратную реализацию криптосистем.

Эллиптической кривой **E** называется множество точек (**x**, **y**), удовлетворяющих однородному уравнению Вейерштрасса:

$$y^2 + a_1xy + a_2y = x^3 + a_3x^2 + a_4x + a_5, \qquad (11)$$

где a_i - коэффициенты уравнения.



а) $y^2 = x^3 - x$ б) $y^2 = x^3 - 3x + 2$ в) $y^2 = x^3 - x + 1$

Рис.1. Примеры эллиптических кривых

В криптографии эллиптические кривые рассматриваются над двумя типами конечных полей: простыми полями нечётной характеристики ($\mathbb{Z}_p$, где $p > 3$ - простое число) и полями характеристики 2 (**GF(2^m)**).

Эллиптические кривые над полями нечётной характеристики $\mathbb{Z}_p$ можно привести к виду, называемому эллиптической кривой в короткой форме Вейерштрасса:

$$y^2 \equiv x^3 + Ax + B \pmod p, \qquad (12)$$

где A, B ∈ $\mathbb{Z}_p$ - коэффициенты эллиптической кривой, удовлетворяющие $4A^3 + 27B^2 \neq 0 \pmod p$.

Запись **A ≡ B (mod p)** означает, что остатки от деления на **p** левой (**A**) и правой (**B**) частей выражения равны. В таких случаях говорят, что правая и левая части выражения **"сравнимы по модулю"**. Например, $7^2 \equiv 2^3 + 4 * 2 + 3 \pmod 5$ ⇒ $(7^2) \bmod 5 = (2^3 + 4 * 2 + 3) \bmod 5$ ⇒ $49 \bmod 5 = 19 \bmod 5$ ⇒ $4 = 4$.

Поскольку $y = \pm\sqrt{x^3 + Ax + B}$, график кривой симметричен относительно оси абсцисс. Чтобы найти точки его пересечения с осью абсцисс, необходимо решить кубическое уравнение

$x^3 + Ax + B = 0.$ (13)

Это можно сделать с помощью известных формул Кардано. Дискриминант этого уравнения

$$\Delta = \left(\frac{A}{3}\right)^3 + \left(\frac{B}{2}\right)^2 = \frac{4A^3+27B^2}{108}.$$ (14)

Если Δ > 0, то уравнение имеет три различных действительных корня (см. рис. 1а – x_1, x_2 и x_3). Если Δ = 0, то уравнение имеет три действительных корня, по крайней мере, два из которых равны (см. рис. 1б – x_1 и x_2). Если Δ < 0, то уравнение имеет один действительный корень (см. рис. 1в – x_1) и два комплексно сопряженных.

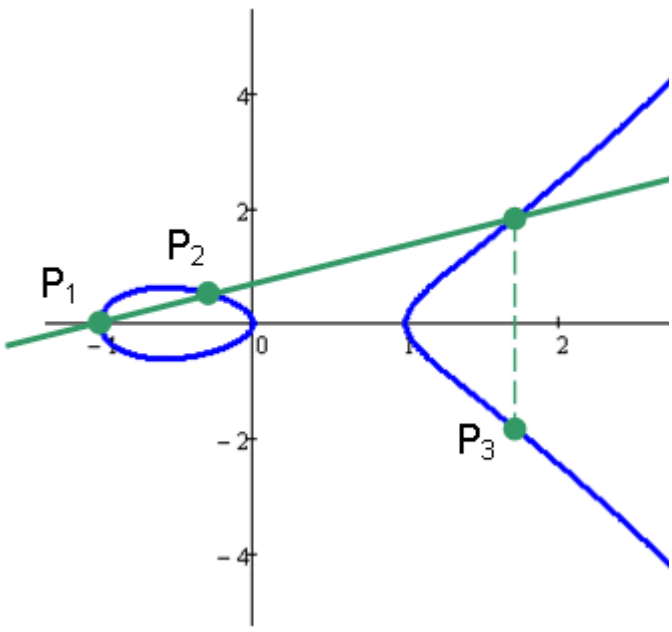
Используемые в криптографии кривые не должны иметь особых точек. Геометрически это значит, что график не должен иметь точек возврата и самопересечений (см. рис. 1б). Если кривая не имеет особых точек, то её график имеет две части при положительном дискриминанте (см. рис. 1а), и одну - при отрицательном (см. рис. 1в). Например, для графиков выше в первом случае дискриминант равен 64, а в третьем он равен -368.

Следует отметить, что в $\mathbb{Z}_p$ у каждого ненулевого элемента есть либо два квадратных корня, либо нет ни одного, поэтому точки эллиптической кривой разбиваются на пары вида P = (x, y) и -P = (x, -y). Например, эллиптическая кривая $y^2 = x^3 + 3x + 2$ над полем $\mathbb{Z}_5$ при x = 1 и p = 5 имеет две точки в качестве решения: P = (1, 1) и -P = (1, -1), т.к. $1^2 \equiv 1^3 + 3 * 1 + 2 \pmod 5$ и $(-1)^2 \equiv 1^3 + 3 * 1 + 2 \pmod 5$.

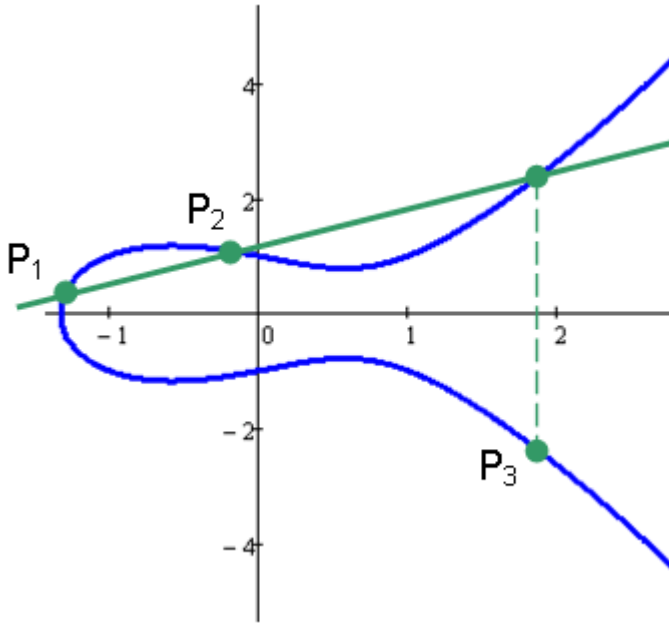
Введем две операции, которые можно выполнять над точками кривой.

Сложение точек $P_3(x_3, y_3) = P_1(x_1, y_1) + P_2(x_2, y_2)$.

Для геометрического сложения точек P_1 и P_2 необходимо через них провести прямую до третьего пересечения с эллиптической кривой и из него опустить перпендикуляр относительно оси абсцисс до четвертого пересечения - это и будет результат сложения (точка P_3).



а) $y^2 = x^3 - x$



б) $y^2 = x^3 - x + 1$

Рис.2. Сложение точек

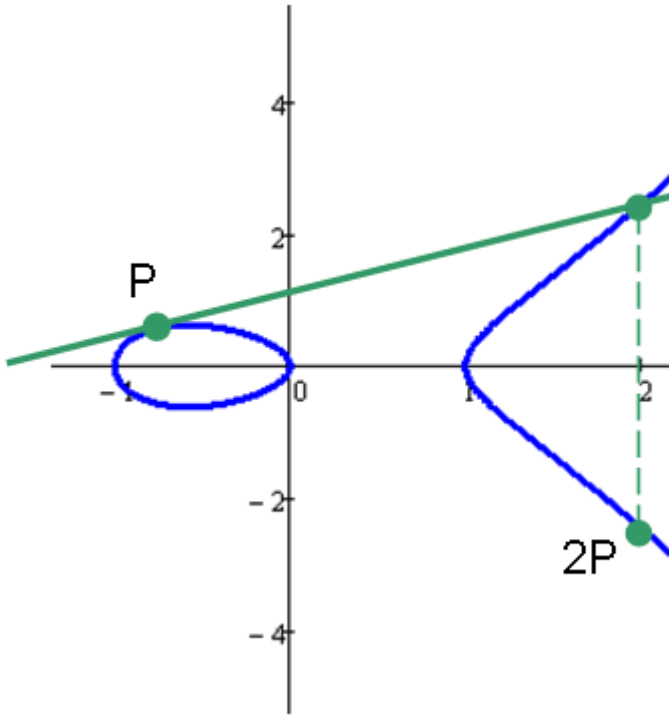
Аналитически сложение точек P_1 и P_2 выполняется по формулам:

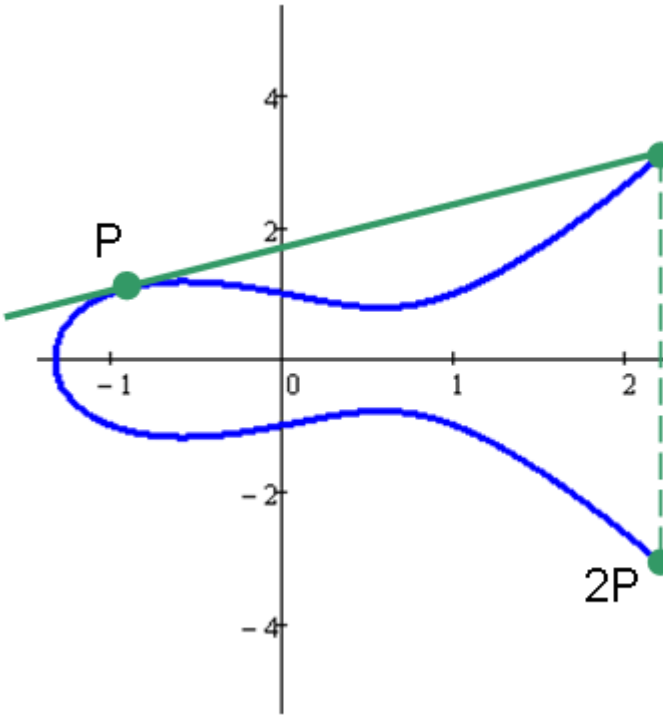
$$\begin{cases} x_3 = (\lambda^2 - x_1 - x_2) \bmod p \\ y_3 = (\lambda(x_1 - x_3) - y_1) \bmod p, \end{cases} \quad (15)$$

где λ - угловой коэффициент прямой, проходящей через точки P_1 и P_2, рассчитываемый по выражениям:

- при $x_1 \neq x_2$: $\lambda \equiv \dfrac{y_2 - y_1}{x_2 - x_1} \pmod p$;
- при $x_1 = x_2$: $\lambda \equiv \dfrac{3x_1^2 + A}{2y_1} \pmod p$.

Если $P_1 = P_2$, то λ равен значению производной в точке P_1.





а) $y^2 = x^3 - x$ б) $y^2 = x^3 - x + 1$

Рис.3. Удвоение точки

Умножение точки на число $P_k(x_k, y_k) = k * P(x, y)$.

Способ 1. Выполнить **k** раз сложение точки **P**

$$P_k(x_k,y_k) = k \cdot P(x,y) = \underbrace{P + P + P + \ldots + P}_{k} \quad (16)$$

Способ 2. С использование двоичного представления числа **k = (b_L, …, b_2, b_1)** и операции удвоения точки. Например, $k = 110_{10} = 1101110_2$, тогда $P_k = 64P + 32P + 8P + 4P + 2P$.

Алгоритм вычисления $\mathbf{P_k}$ может выглядеть следующим образом.

1. P_k := null
2. Q := P
3. Цикл. Для i := 1 до L

3.1. Если $b_i = 1$

то Если P_k = null

то P_k = Q

иначе $P_k = P_k + Q$

3.2. Q = 2 * Q // ≈ Q = Q + Q

Для k = 110 вместо 109 сложений будет 1 присваивание, 4 сложения и 6 удвоений (сложений).

Рассмотрим процедуру создания ключей.

Таблица 11. Процедура создания ключей

№ п/п	Описание операции	Пример
1	Выбирается простое число **p** - модуль эллиптической кривой (по ГОСТ – $p > 2^{255}$).	p = 41
2	Выбираются коэффициенты эллиптической кривой **A** и **B**. Должно соблюдаться условие $(4 A^3 + 27 B^2) \bmod p \neq 0$, в противном случае меняются параметры эллиптической кривой p, A или B.	A = 3, B = 7 $(4 * 3^3 + 27 * 7^2) \bmod 41 = 37$
3	Определяется точка эллиптической кривой $\mathbf{P(x_p, y_p)}$ и порядок циклической подгруппы группы точек эллиптической кривой **q**[*)]. Выбирается произвольное натуральное число x_p ($0 < x_p < p$) и определяется y_p из уравнения эллиптической кривой. Должны соблюдаться условия: - для x_p должен существовать y_p (не для всякого x_p при данных параметрах кривой (A, B, p) может существовать y_p);	$x_p = 7$, $y_p = 17$ [$17^2 \bmod 41 = (7^3+3*7+7) \bmod 41 = 2$] q = 47

	- $P \neq O$ и $q P = O$, где O - нулевая точка эллиптической кривой ($P + (-P) = O$); - q – простое число; - $p^t \bmod q \neq 1$, для всех целых $t = 1, 2, \ldots, T$, где $T \leq 31$; - $2^{254} < q < 2^{256}$ (по ГОСТ). В противном случае меняются либо параметры эллиптической кривой p, A или B либо выбирается другая точка P.	
4	Выбирается закрытый ключ **d** (дискретный логарифм в группе точек эллиптической кривой) - произвольное натуральное число ($0 < d < q$).	$d = 10$
5	Определяется точка эллиптической кривой $\mathbf{Q(x_q, y_q)} = d * P(x_p, y_p)$.	$x_q = 36$, $y_q = 20$
6	Публикуется открытый ключ $\mathbf{\{(A, B), P(x_p, y_p), p, Q(x_q, y_q)\}}$ в специальном хранилище, где исключается возможность его подмены (**общедоступном сертифицированном справочнике**). Для выработки и проверки электронной цифровой подписи **q** является частью открытого ключа вместо **p**.	

*) Прямой способ вычисления порядка группы точек эллиптической кривой **q**.

1) Рассчитываются координаты первой точки из выражения

$y^2 \equiv x^3 + Ax + B \pmod p \quad \Rightarrow \quad y^2 \equiv x^3 + 3x + 7 \pmod{41}$.

Примем $x_1 = 7$, тогда $y_1^2 \bmod 41 = (7^3 + 3 * 7 + 7) \bmod 41 = 2$, откуда $y_1 = 17$.

$P(x_p, y_p) = P(x_1, y_1) = P(7, 17)$.

2) Находим координаты второй точки. Для этого вначале вычисляется коэффициент λ

$$\lambda \equiv \frac{(3x_1^2 + A)}{2y_1} (mod\ p) \equiv \frac{3*7^2 + 3}{2*17} (mod\ 41) \equiv \frac{150}{34} (mod\ 41) \Rightarrow$$

$$34\lambda\ mod\ 41 = 150\ mod\ 41 \Rightarrow$$

$$34\lambda\ mod\ 41 = 27$$

Решая последнее уравнение, получаем $\lambda = 2$ ($34 * 2 \bmod 41 = 27$).

Координаты второй точки получаем путем удваивания первой из выражений:

$x_2 = (\lambda^2 - 2x_1) \bmod p = (2^2 - 2 * 7) \bmod 41 = -10 \bmod 41 = 31$,

$y_2 = (\lambda(x_1 - x_2) - y_1) \bmod p = (2\ (7 - 31) - 17) \bmod 41 = -65 \bmod 41 = 17$.

3) Каждую следующую точку рассчитываем по формулам, пока в знаменателе первой формулы не будет получен 0:

$$\lambda_i \equiv \frac{y_{i-1} - y_1}{x_{i-1} - x_1} \ (mod\ p),$$

$$\begin{cases} x_i = (\lambda_i^2 - x_1 - x_{i-1})\ mod\ p \\ y_i = (\lambda_i(x_1 - x_i) - y_1)\ mod\ p \end{cases} \qquad (17)$$

Получаем:
- $\lambda_3 = 0$, $x_3 = 3$, $y_3 = 24$;
- $\lambda_4 = 29$, $x_4 = 11$, $y_4 = 31$;
- $\lambda_5 = 24$, $x_5 = 25$, $y_5 = 2$;
- …;
- $\lambda_{46} = 2$, $x_{46} = 7$, $y_{46} = 24$.

К полученному числу точек добавляем точку **O**, в результате чего q = 46 + 1 = 47. Точка **O** есть результат сложения любых двух точек $P(x_i, y_i)$ и $-P(x_i, -y_i)$ и представляет собой бесконечно удаленную точку, в которой гипотетически сходятся все вертикальные кривые.

Процедура шифрования отдельного блока выполняется следующим образом.

Таблица 12. Процедура шифрования отдельного блока (буквы)

№ п/п	Описание операции	Пример
1	Определяется десятичное представление буквы **t**.	Буква «К» t = 12
2	Выбирается случайное число **k** (0 < k < p).	k = 5
3	Определяется точка эллиптической кривой $\mathbf{P_k(x_{pk}, y_{pk})} = k * P$.	$P_k(25, 2)$
4	Определяется точка эллиптической кривой $\mathbf{Q_k(x_{qk}, y_{qk})} = k * Q$.	$Q_k(3, 24)$
5	Вычисляется $\mathbf{c} = (t * x_{qk}) \bmod p$.	$c = (12 * 3) \bmod 41 = 36$
6	Шифрограмма – пара $\{P_k, c\}$.	$\{P_k(25, 2), 36\}$

Процедура расшифрования отдельного блока выполняется следующим образом.

Таблица 13. Процедура расшифрования отдельного блока (буквы)

№ п/п	Описание операции	Пример
1	Определяется точка эллиптической кривой $\mathbf{D(x_d, y_d)} = d * P_k$.	D(3, 24)
2	Вычисляется десятичное представление зашифрованной буквы $\mathbf{t} = (c * x_d^{-1}) \bmod p$, где x_d^{-1} – обратное число к x_d по модулю p.	$x_d^{-1} = 14$ $[(3 * 14) \bmod 41 = 1]$ $t = (36 * 14) \bmod 41 = 12$

3	Определяется исходное сообщение по ее десятичному представлению.	Буква «К»

Приведенный выше способ шифрования является вариацией шифрования Эль-Гамаля. Если стойкость алгоритма шифрования Эль-Гамаля базируется на сложности решения задачи дискретного логарифмирования, то стойкость шифрования с помощью эллиптических кривых базируется на сложности нахождения множителя **k** точки **P** по их произведению. Т.е. если **Q = k P**, то зная **P** и **k** довольно легко вычислить **Q**. Эффективное решение обратной задачи (найти **k** при известных **P** и **Q**) на текущий момент пока не опубликовано.

Вопросы для самоконтроля

1. Требования к системе защиты информации.
2. Криптографические атаки.
3. Виды атак.
4. Принцип Керкхоффса.

Практическое занятие №5. Хеш-функция (MD5)
Цель: изучить способы шифрования данных с помощью Хеш-функций.
Ход работы:

Необходимо ознакомиться с теоретической частью и по алгоритму MD5 получить хеш-образ сообщения, состоящего из первых трех букв своей фамилии.

При выполнении задания необходимо привести:

- теоретическую часть, включающую "Шаг основного цикла вычисления хеша" (рис.2), "Шаг цикла раунда" (рис.3) и таблицу "Раундовые функции RF" (табл.1);
- исходное сообщение (3 буквы фамилии) в символьном и десятичном представлениях в соответствии с кодировкой Windows 1251;
- прообраз сообщения в шестнадцатеричном представлении (512 бит), включая вспомогательные единичный и нулевые биты, а также биты, определяющие длину сообщения;
- исходные значения переменных A, B, C и D в шестнадцатеричном представлении (по 32 бита);
- для 1-го, 2-го и 16-го 32-битового блока прообраза - результаты вычислений переменных A, B, C и D в шестнадцатеричном представлении для всех раундов;

Раунд	Итерация	Значения переменных							
		в начале итерации				в конце итерации			
		A	B	C	D	A	B	C	D
1	1								
	2								
	...								
	16								
2	1								
	2								
	...								
	16								
3	1								
	2								
	...								
	16								
4	1								

	2								
	...								
	16								

- результат итогового сложения по модулю 2^{32} исходных значений переменных A, B, C и D со значениями этих переменных, полученных после 4-го раунда в шестнадцатеричном представлении (128 бит) до и после перестановки байт.

Основные понятия

Хеширование (иногда хэширование, англ. hashing) - преобразование входного массива данных произвольной длины в выходную строку фиксированной длины. Такие преобразования также называются **хеш-функциями** или **функциями свёртки**, входной массив – **прообразом**, а результаты преобразования - **хешем, хеш-кодом, хеш-образом, цифровым отпечатком** или **дайджестом сообщения** (англ. message digest).

Хеш-функция – легко вычислимая функция, преобразующая исходное сообщения произвольной длины (прообраз) в сообщение фиксированной длины (хеш-образ), для которого не существует эффективного алгоритма поиска коллизий.

Коллизией для функции **h** называется такая пара значений **(x, y)**, что **h(x) = h(y)** при $x \neq y$. Таким образом хеш-функция должна обладать следующими свойствами:

- для данного значения **h(x)** невозможно найти значение аргумента **x**. Такие хеш-функции называют **стойкими в смысле обращения** или **стойкими в сильном смысле**;

- для данного аргумента **x** невозможно найти другой аргумент **y** такой, что **h(x) = h(y)**. Такие хеш-функции называют **стойкими в смысле вычисления коллизий** или **стойкими в слабом смысле**.

В случае, когда значение хеш-функции зависит не только от прообраза, но и закрытого ключа, то это значение называют **кодом проверки подлинности сообщений (Message Authentication Code, MAC), кодом проверки подлинности данных (Data Authentication Code, DAC)**w или **имитовставкой**.

На практике хеш-функции используют в следующих целях:

- для ускорения поиска данных в БД;
- для проверки целостности и подлинности сообщений;
- для создания сжатого образа, применяемого в процедурах ЭЦП;
- для защиты пароля в процедурах аутентификации.

Ускорения поиска данных. Например, при записи текстовых полей в базе данных может рассчитываться их хеш-код и данные могут помещаться в раздел, соответствующий этому хеш-коду. Тогда при поиске данных надо будет сначала вычислить хеш-код текста и сразу станет известно, в каком разделе их надо искать, т.е. искать надо будет не по всей базе, а только по одному её разделу (это сильно ускоряет поиск).

Бытовым аналогом хеширования в данном случае может служить размещение слов в словаре по алфавиту. Первая буква слова является его хеш-кодом, и при поиске мы просматриваем не весь словарь, а только раздел с нужной буквой.

Процедура вычисления (стандартная схема алгоритма) хеш-функции представлена на следующем рисунке.

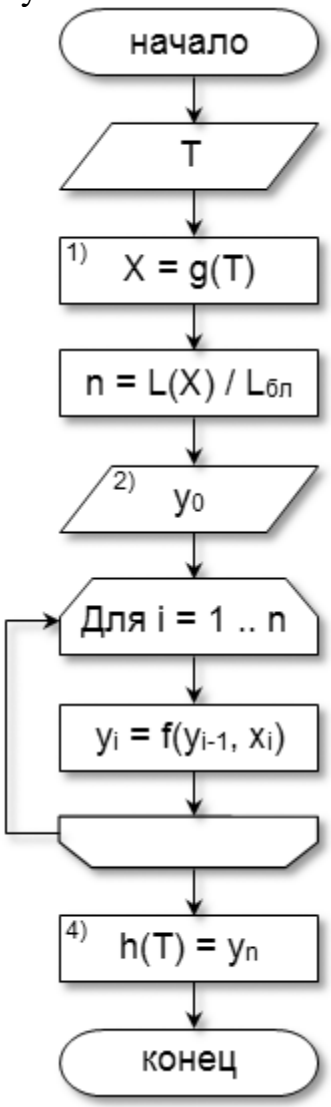


Рис.1. Процедура вычисления значения хеш-функции

1) К исходному сообщению **T** добавляется вспомогательная информация (например, длина прообраза, вспомогательные символы и т.д.) так, чтобы длина прообраза **X** стала кратной величине $\mathbf{L_{бл}}$, определенной спецификацией (стандартом) хеш-функции.

2) Для инициализации процедуры хеширования используется синхропосылка $\mathbf{y_0}$.

3) Прообраз **X** разбивается на **n** блоков $\mathbf{x_i}$ (i = 1 .. n) фиксированной длины $\mathbf{L_{бл}}$, над которыми выполняется однотипная процедура хеширования $\mathbf{f(y_{i-1}, x_i)}$, зависящая от результата хеширования предыдущего блока $\mathbf{y_{i-1}}$.

4) Хеш-образом **h(T)** исходного сообщения **T** будет результат процедуры хеширования $\mathbf{y_n}$, полученный после обработки последнего блока $\mathbf{x_n}$.

MD5

MD5 (англ. Message Digest 5) – 128-битный алгоритм хеширования, разработанный профессором Рональдом Л. Ривестом из Массачусетского технологического института (Massachusetts Institute of Technology, MIT) в 1991 г. Является улучшенной в плане безопасности версией MD4.

Ниже приведен алгоритм вычисления хеша.

1. Выравнивание потока.

В конец исходного сообщения, длиной **L**, дописывают единичный бит, затем необходимое число нулевых бит так, чтобы новый размер **L'** был сравним с 448 по модулю 512 (L’ mod 512 = 448). Добавление нулевых бит выполняется, даже если новая длина, включая единичный бит, уже сравнима с 448.

2. Добавление длины сообщения.

К модифицированному сообщению дописывают 64-битное представление длины данных (количество бит в сообщении). Т.е. длина сообщения **T** становится кратной 512 (T mod 512 = 0). Если длина исходного сообщения превосходит 2^{64} - 1, то дописывают только младшие 64 бита. Кроме этого, для указанного 64-битного представления длины вначале записываются младшие 32 бита, а затем старшие 32 бита.

3. Инициализация буфера.

Для вычислений инициализируются 4 переменных размером по 32 бита и задаются начальные значения (шестнадцатеричное представление):

A = 67 45 23 01;

B = EF CD AB 89;

C = 98 BA DC FE;

D = 10 32 54 76.

В этих переменных будут храниться результаты промежуточных вычислений. Начальное состояние **ABCD** называется инициализирующим вектором.

4. Вычисление хеша в цикле.

Исходное сообщение разбивается на блоки **T**, длиной 512 бит. Для каждого блока в цикле выполняется процедура, приведенная на рис.2. Результат обработки всех блоков исходного сообщения в виде объединения 32-битных значений переменных **ABCD** и будет являться хешем.

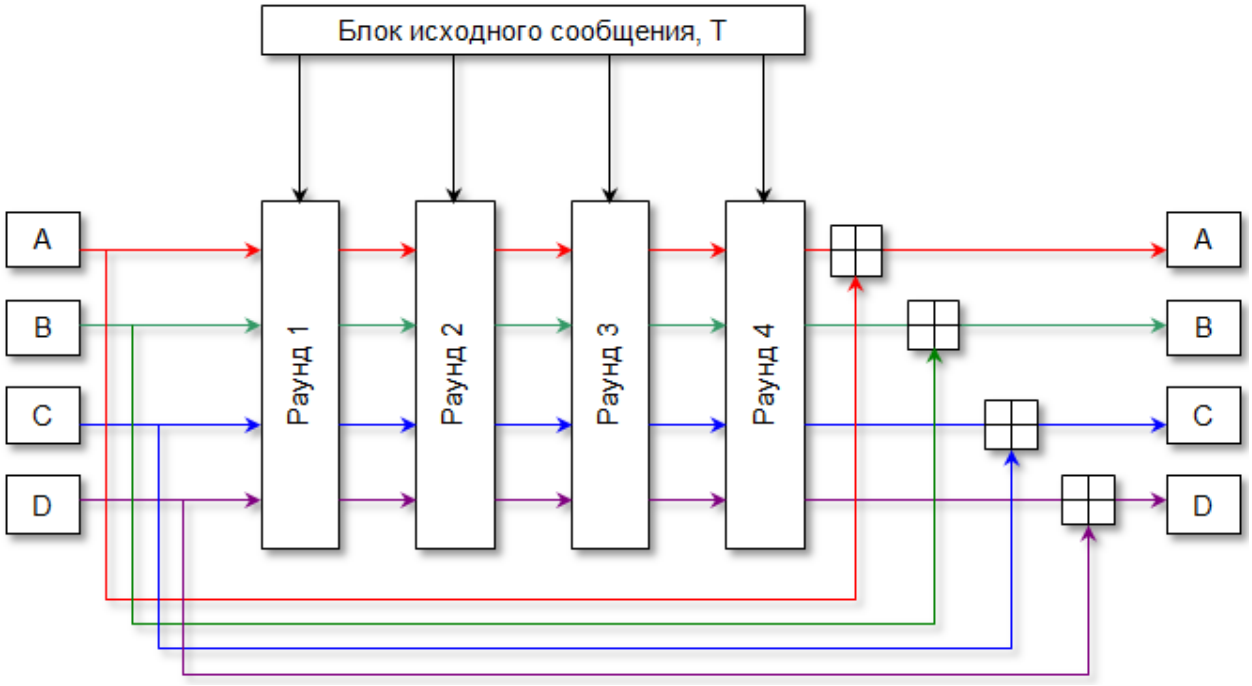


Рис.2. Шаг основного цикла вычисления хеша

В каждом раунде над переменными **ABCD** и блоком исходного текста **T** в цикле (16 итераций) выполняются однотипные преобразования по следующей схеме.

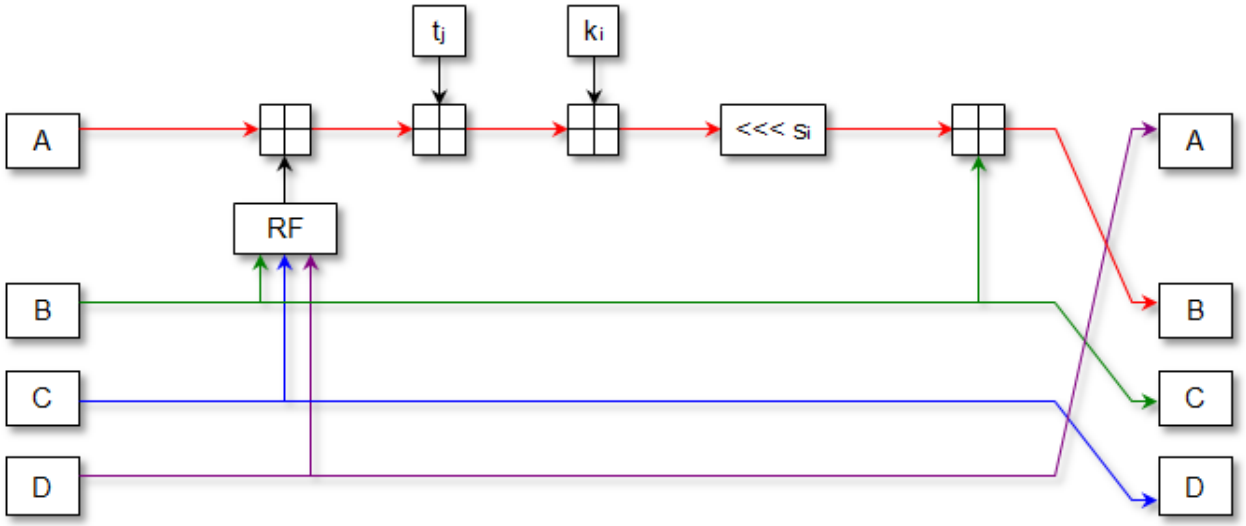


Рис.3. Одна итерация цикла раунда

Условные обозначения.

1) **RF** - раундовая функция, определяемая по следующей таблице.

Таблица 1. Раундовые функции RF

№ раунда	Обозначение функции	Формула расчета
1	F	F(B, C, D) = (B ∧ C) ∨ (¬B ∧ D)
2	G	G(B, C, D) =

		$(B \wedge D) \vee (\neg D \wedge C)$
3	H	$H(B, C, D) = B \oplus C \oplus D$
4	I	$I(B, C, D) = C \oplus (\neg D \vee B)$

2) $\mathbf{t_j}$ - j-ая 32-битовая часть блока исходного сообщения **T** с обратным порядком следования байт;

3) $\mathbf{k_i}$ - целая часть константы, определяемой по формуле

$k_i = 2^{32} * | \sin(i + 16 * (r - 1)) |$, (1)

где i – номер итерации цикла (i = 1..16); r – номер раунда (r = 1..4).

Аргумент функции sin измеряется в радианах.

4) $\boxplus$ – сложение по модулю 2^{32}.

5) <<< $\mathbf{s_i}$ – циклический сдвиг влево на s_i разрядов.

Используемая 32-битовая часть блока исходного сообщения $\mathbf{t_j}$ и величина циклического сдвига влево $\mathbf{s_i}$ зависят от номера итерации и приведены в следующей таблице.

Таблица 2. Величины, используемые на шаге цикла раунда

№ итерации		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Раунд 1	t_j	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}
	s_i	7	12	17	22	7	12	17	22	7	12	17	22	7	12	17	22
Раунд 2	t_j	t_2	t_7	t_{12}	t_1	t_6	t_{11}	t_{16}	t_5	t_{10}	t_{15}	t_4	t_9	t_{14}	t_3	t_8	t_{13}
	s_i	5	9	14	20	5	9	14	20	5	9	14	20	5	9	14	20
Раунд 3	t_j	t_6	t_9	t_{12}	t_{15}	t_2	t_5	t_8	t_{11}	t_{14}	t_1	t_4	t_7	t_{10}	t_{13}	t_{16}	t_3
	s_i	4	11	16	23	4	11	16	23	4	11	16	23	4	11	16	23
Раунд 4	t_j	t_1	t_8	t_{15}	t_6	t_{13}	t_4	t_{11}	t_2	t_9	t_{16}	t_7	t_{14}	t_5	t_{12}	t_3	t_{10}
	s_i	6	10	15	21	6	10	15	21	6	10	15	21	6	10	15	21

После 4 раундов новое (модифицированное) значение каждой из переменных **ABCD** складывается ($\boxplus$) с исходным (значением переменной до 1-го раунда).

5. Перестановка байт в переменных ABCD. После обработки всех блоков исходного сообщения для каждой переменной выполняется обратная перестановка байт.

Поиск коллизий.

В 2004 г. китайские исследователи Ван Сяоюнь (Wang Xiaoyun), Фен Дэнгуо (Feng Dengguo), Лай Сюэцзя (Lai Xuejia) и Юй Хунбо (Yu Hongbo) объявили об обнаруженной ими уязвимости в алгоритме, позволяющей за небольшое время (1 час на кластере IBM p690) находить коллизии.

Применение шифрования для получения хеш-образа

Для выработки устойчивого к коллизиям хеш-образа могут применяться специальные режимы, предусмотренные в блочных шифрах (например, сцепление блоков шифра у DES), или в самой хеш-функции, как составная часть, может использоваться один из режимов блочного шифра.

Напомним что в случае, когда значение хеш-функции зависит не только от прообраза, но и закрытого ключа, то хеш-образ называют **кодом проверки подлинности сообщений (Message Authentication Code, MAC), кодом проверки подлинности данных (Data Authentication Code, DAC)** или **имитовставкой**.

В качестве примера приведем режим DES-CBC (сцепление блоков шифра - Cipher Block Chaining).

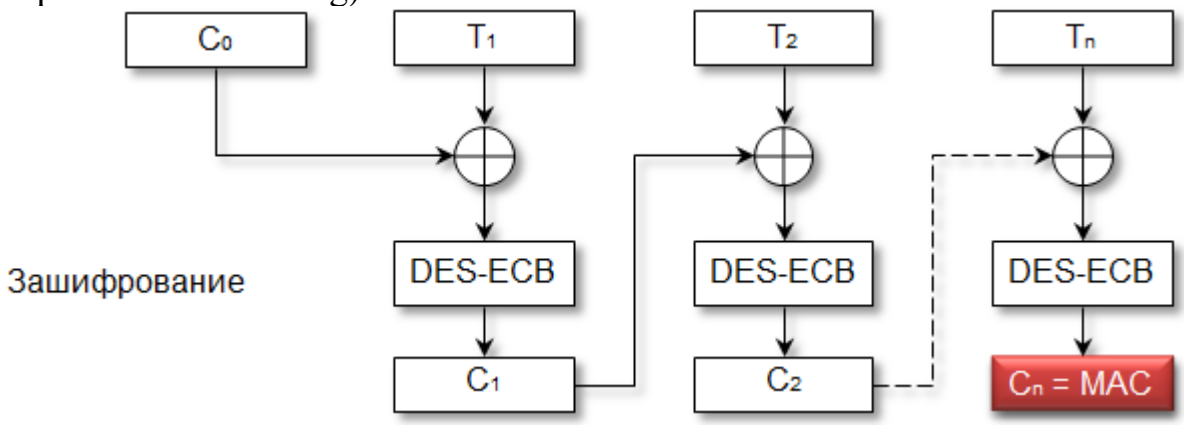


Рис.4. Схема алгоритма DES в режиме сцепления блоков шифра

Последний зашифрованный блок $\mathbf{C_n}$ и есть хеш-образ сообщения $\mathbf{T = \{T_1, T_2, \ldots, T_n\}}$.

Вопросы для самоконтроля

1. Виды шифрования данных.
2. Симметричное шифрование.
3. Ассиметричное шифрование.

Практическое занятие №6. Идентификация и аутентификация

Цель: изучить процедуры идентификации и аутентификации.

Ход работы:

Необходимо привести последовательность выполнения процедур идентификации/аутентификации с использованием следующих способов:

- на основе алгоритма RSA;
- по схеме Шнорра;
- по схеме Фейге-Фиата-Шамира.

При выполнении задания необходимо привести таблицы генерации ключей и аутентификации. В качестве случайного числа (k или r) принять коды, соответственно, 1-ой, 2-ой и 3-ей буквы своей фамилии согласно их положению в алфавите.

Общие сведения

Идентификация (англ. identification) - процесс распознавания сущностей путем присвоения им уникальных меток (идентификаторов, логинов).

Аутентификация (англ. authentication) - проверка соответствия (подлинности) сущности предъявленному ею идентификатору. (Заметим, что происхождение русскоязычного термина «аутентификация» не совсем понятно. Английское «authentication» скорее можно прочитать как «аутентикация»; трудно сказать, откуда в середине взялось еще «фи» – может, из идентификации? Тем не менее, термин устоялся и закреплен в РД Гостехкомиссии РФ).

Для полноты картины приведем определение термина авторизация, который не следует путать с двумя вышеприведенными. **Авторизация** (англ. authorization) - предоставление сущности возможностей в соответствии с положенными ей правами или проверка наличия прав при попытке выполнить какое-либо действие.

Идентификация и аутентификация – это первая линия обороны, «входная дверь» в информационное пространство организации.

Идентификация позволяет субъекту (пользователю, процессу, действующему от имени определенного пользователя, или иному аппаратно-программному компоненту) назвать себя (сообщить свое имя). Посредством аутентификации вторая сторона убеждается, что субъект действительно тот, за кого он себя выдает.

Идентификация сродни присвоении имени ребенку (не совсем точное сравнение, но все же). В любой информационной системе должны быть определены все субъекты, участвующие в информационном обмене. Часть из них может быть сгруппирована, если они наделены одинаковыми (схожими) правами и обладают одинаковыми (схожими) характеристиками. Каждый субъект (группа субъектов) должен обладать уникальным именем (обозначением).

Аутентификация бывает **односторонней** (обычно клиент доказывает свою подлинность серверу) и **двусторонней** (взаимной). Пример односторонней аутентификации – процедура входа пользователя в систему.

Субъект может подтвердить свою подлинность, предъявив один из следующих **аутентификаторов**:

- нечто, что он знает (пароль, личный идентификационный номер, криптографический ключ и т.п.);

- нечто, чем он владеет (паспорт, личную карточку или иное устройство аналогичного назначения);

- нечто, что есть часть его самого (голос, отпечатки пальцев, образец ДНК и т.п.).

В том случае, если в ходе процедуры аутентификации клиент должен предъявить сразу несколько аутентификаторов, аутентификация называется **многофакторной**. Например, в ходе двухфакторной аутентификации клиент должен знать пароль и воспользоваться личной карточкой.

Рассмотрим основные программно-технические способы реализации идентификации и аутентификации:

- пароли;
- с использованием хеш-функции;
- на основе шифрования с открытым ключом;
- сервер аутентификации Kerberos;

Парольная идентификация/аутентификация

Введенный пользователем пароль сравнивается с паролем, имеющимся в БД, хранящейся в защищаемой информационной системе, и если они совпадают, то дается разрешение на использование защищаемых ресурсов.

Главное **достоинство** парольной аутентификации – простота и привычность. Пароли давно встроены в ОС, СУБД и программные продукты. При правильном использовании пароли могут обеспечить приемлемый для многих организаций уровень безопасности. Тем не менее, по совокупности характеристик их следует признать самым слабым средством проверки подлинности.

Парольная аутентификация имеет массу **недостатков**:

- как правило, пароль генерируется в одном месте (например, на сервере) и должен быть передан во второе (например, клиенту). При передаче пароль может быть перехвачен злоумышленником;

- многие ОС и приложения имеют пароли, указанные производителем по умолчанию. После установки такой системы очень часто забывают их удалить. БД стандартных паролей можно найти в Интернете;

- злоумышленник может получить БД паролей, хранящихся в зашифрованном виде, и воспользоваться ей:

  - в Windows NT/2000/XP учетные записи (пользователи и пароли) хранятся в файле «%System Root% \ System32 \ Config \ sam». При работающей ОС пользователь не может выполнять операции чтения/записи с данным файлом (блокируется процессом lsass.exe, «убить» который невозможно). Получить доступ к файлу можно, загрузив ОС с другого носителя. Другой вариант заключается в использовании файла «%System Root% \ Repair \ sam». Он доступен для чтения/записи, но, как правило, содержит пароли «столетней» давности;

  - в ранних версиях Unix файл с учетными записями «/etc/passwd» был доступен для чтения любым желающим. В современных разновидностях

Unix файл с паролями «/etc/shadow» или «etc/secure» доступен только с привилегиями супервизора. Другой способ получения доступа к паролям – обрушения процесса, обращающегося к файлу с паролями. При этом Unix создает файл «core dump», содержащий дамп памяти (с паролями);

- после получения файла с зашифрованными паролями можно воспользоваться многочисленными программами-взломщиками. Одними из самых популярных взломщиков являются: для Windows – L0phtCrack, для Unix – John the Ripper. Время, требуемое для взлома пароля, зависит от его качества. Так, например, взлом пароля для L0phtCrack на компьютере с процессором Xeon 400 МГц при использовании:

- цифр и латиницы – 5,5 часов;

- всех символов – 480 часов.

- кроме перечисленных выше приемов взлома паролей, их можно подсмотреть (например, с помощью оптических приборов), сообщить другу/подруге (если секрет знают двое – это уже не секрет), записать на бумажке и приклеить на клавиатуру или монитор и т.п.

Тем не менее, так как парольная защита используется во многих продуктах и системах, можно порекомендовать следующие **меры, позволяющие повысить надежность парольной защиты**:

- наложение технических ограничений (пароль должен быть не слишком коротким, он должен содержать буквы, цифры, знаки пунктуации и т.п.). Еще лучше воспользоваться программами - генераторами паролей (ключей);

- ограничение доступа к файлу с паролями;

- удаление резервных копий файлов с паролями («%System Root% \ Repair \ sam»);

- использование защищенных протоколов обмена ключами (например, основанные на протоколе обмена ключами Диффи-Хеллмана);

- ограничение числа неудачных попыток входа в систему (это затруднит применение «метода грубой силы»). В Windows 2000 и XP этот параметр устанавливается по пути «Администрирование / Локальная политика безопасности / Политики учетных записей / Политика блокировки учетной записи / Пороговое значение блокировки». Там же («Политики учетных записей») можно настроить срок блокировки учетной записи, минимальную длину пароля, сроки его действия и т.п.;

- управление сроком действия паролей, их периодическая смена, использование сеансовых ключей;

- удаление паролей уволенных или лишенных полномочий пользователей.

Протокол идентификации/аутентификации с использованием хеш-функции

Напомним, что хеш-функция – легко вычислимая функция, преобразующая исходное сообщения произвольной длины (прообраз) в сообщение фиксированное длины (хеш-образ), для которой не существует эффективного алгоритма поиска коллизий.

При идентификации/аутентификации пользователь вводит пароль, а по каналу связи высылается его хеш-образ. Проверяющая система сравнивает

введенный хеш-образ с образом, хранящемся в информационной системе для этого пользователя и в случае их совпадения разрешает доступ. Таким образом, система не хранит паролей, что повышает ее защищенность (**достоинство**).

Недостаток приведенной схемы заключается в том, что все равно необходимо как-то передавать хеш-образ для хранения в системе или для аутентификации и на этом пути его может перехватить злоумышленник, а затем воспользоваться им.

Протокол идентификации/аутентификации на основе шифрования с открытым ключом

Широкое распространение при идентификации и аутентификации получили протоколы на базе асимметричного шифрования. Существует десятки разновидностей таких протоколов, наиболее известными из которых являются протоколы на основе алгоритмов RSA, схемы Фейге-Фиата-Шамира, Эль-Гамаля, Шнорра и т.д.

I. Протокол аутентификации на основе алгоритма RSA.

Этап 1. Генерация ключей.

1. **А** генерирует открытый {e=5, n=91} и закрытый {d=29} ключи.
2. **А** передает открытый ключ **Б**.

Этап 2. Аутентификация.

Таблица 1. Аутентификация на основе алгоритма RSA

№ п/п	Описание операции	Пример
1	**Б** выбирает случайное число **k** ($0 < k < n$), вычисляет **r** = $k^e \bmod n$ и посылает **r А**.	k = 23 r = $23^5 \bmod 91 = 4$
2	**А** вычисляет **k'** = $r^d \bmod n$ и посылает **k' Б**.	k' = $4^{29} \bmod 91 = 23$
3	**Б** проверяет соотношение **k** = **k'** и, если оно истинно, принимает доказательство, в противном случае - отвергает.	23 = 23

II. Схема аутентификации Клауса Шнорра.

Этап 1. Генерация ключей (выполняет **А**).

Таблица 2. Генерация ключей по схеме Клауса Шнорра

№ п/п	Описание операции	Пример
1	Выбираются два простых числа **p** и **q** такие, что $(p - 1) \bmod q = 0$.	p = 23, q = 11
2	Выбирается секретный ключ **x** ($0 < x < q$).	x = 8
3	Выбирается **g**, для которго $g^q \bmod p = 1$.	g = 3 [$3^{11} \bmod 23 = 1$]

4	Выбирается **y**, для которого (g^x * y) mod p = 1.	y = 4 [(3^8 * 4) mod 23 = 26244 mod 23 = 1]
5	Публикация открытого ключа {**y**, **g**, **p**}.	

Этап 2. Аутентификация.

Таблица 3. Аутентификация по схеме Клауса Шнорра

№ п/п	Описание операции	Пример
1	**А** выбирает случайное число **k** (0 < k < q), вычисляет **r** = g^k mod p и посылает **r Б**.	k = 6 r = 3^6 mod 23 = 16
2	**Б** выбирает случайное число **e** (0 < e < 2^t), где t - некоторый параметр, и посылает **e А**.	e = 4
3	**А** вычисляет **s** = (k + x * e) mod q и посылает **s Б**.	s = (6 + 8 * 4) mod 11 = 5
4	**Б** проверяет соотношение **r** = (g^s * y^e) mod p и, если оно выполняется, принимает доказательство, в противном случае - отвергает.	16 = (3^5 * 4^4) mod 23

Для обеспечения стойкости протокола в 1989 г. Шнорр рекомендовал использовать **p** длиной 512 бит, **q** длиной 140 бит и **t** = 52.

III. Упрощенная схема аутентификации Фейге-Фиата-Шамира.

Данная схема базируется на протоколе доказательства с нулевым разглашением. **Протокол доказательства с нулевым разглашением (секрета)** - интерактивный (многораундовый) вероятностный протокол обмена информацией, который позволяет одной стороне доказать второй стороне знание секрета без раскрытия самого секрета. Вероятностный характер протокола говорит о том, что в результате обмена информацией у доказывающей стороны, если она не знает секрета, есть все-таки малый шанс убедить вторую сторону в знании секрета.

Суть доказательства с нулевым разглашением популярно можно объяснить на примере «пещеры Аладдина» (авторы - Жан-Жак Кискатер и Луи Гийу).

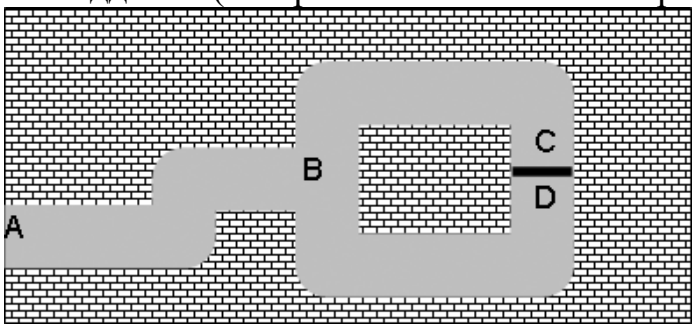


Рис.1. Пещера Аладдина

В пещере имеется потайная дверь **C-D**, открыть которую может только тот, кто знает волшебные слова. Алиса хочет доказать Бобу, что знает волшебные слова, но не хочет их раскрыть Бобу. Тогда Алиса может убедить Боба следующим образом.

1. Боб стоит в точке **A**.
2. Алиса проходит к точке **C** или **D**.
3. Боб проходит к точке **B** и предлагает Алисе появиться с левого прохода или с правого.
4. Алиса выполняет просьбу, используя, если необходимо, волшебные слова.
5. Алиса и Боб **t** раз повторяют шаги 1-4.

Если Алиса не знает секрета, то вероятность правильно выйти у нее в каждом раунде 50%. Если шаги повторить **t** раз, то вероятность правильного выхода во всех случаях 1 шанс на 2^t. Например, при t=16 у Алисы всего 1 шанс из 65536 (= 2^{16}).

Практическая реализация протокола с нулевым разглашением на примере схемы аутентификации Фейге-Фиата-Шамира.

Этап 1. Генерация ключей (выполняет Посредник).

Таблица 4. Генерация ключей по схеме Фейге-Фиата-Шамира

№ п/п	Описание операции	Пример
1	Выбирает модуль **n**, равный произведению двух простых чисел **p** и **q**.	p = 5, q = 7, n = 35
2	Выбирает число **v**, являющееся квадратичным вычетом по модулю **n** и имеется обратное значение $\mathbf{v^{-1}}$ по модулю **n**. **Квадратичный вычет** – число, удовлетворяющее выражению $x^2 \bmod n = v$, где $1 \le x \le n$. Для модуля n = 35, квадратичными вычетами являются 1 (x = 1, 6, 29, 34), 4, 9, 11, 14, 15, 16, 21, 25, 29, 30. Обратное значение вычисляется по формуле $(v * v^{-1}) \bmod n = 1$. У квадратичных вычетов 14, 15, 21, 25 и 30 нет обратных значений по модулю. Таким образом, $v \in \{1, 4, 9, 11, 16, 29\}$.	x = 4 $v = 4^2 \bmod 35 = 16$ $v^{-1} = 11$ $[(16 * 11) \bmod 35 = 176 \bmod 35 = 1]$
3	Определяет закрытый ключ **s** - наименьшее значение, удовлетворяющее выражению $s^2 \bmod n = v^{-1}$.	s = 9 $[9^2 \bmod 35 = 11]$
4	Публикация открытого ключа {**v**, **n**}. Передача А закрытого ключа **s.**	

Этап 2. Аутентификация.

Таблица 5. Аутентификация по схеме Фейге-Фиата-Шамира

№ п/п	Описание операции	Пример	
1	**A** выбирает случайное число **r** (0 < r < n), вычисляет **z** = r^2 mod n и посылает **z Б**.	r = 8 z = 8^2 mod 35 = 29	
2	**Б** посылает **A** случайный бит **b**.	b = 0	b = 1
3	Если **b**=0, то **A** посылает **Б r**, иначе - **y** = (r * s) mod n.	r = 8	y = (8 * 9) mod 35 = 2
4	Если **b**=0, то **Б** проверяет, что **z** = r^2 mod n, иначе - **z** = (y^2 * v) mod n.	29 = 8^2 mod 35	29 = (2^2 * 16) mod 35

Рассмотренный порядок операций, выполненный 1 раз называется **аккредитацией** (раундом). Если первую операцию поменять местами со второй, то **A**, даже не зная закрытого ключа **s**, может подобрать такое значение **r**, которое будет приводить к успешной аккредитации в обоих случаях (b=0 и b=1). Подобрать же такое **r**, которое будет приводить к успешной аккредитации в обоих случаях одновременно невозможно. Таким образом, если **A** не знает закрытого ключа **s**, то вероятность успешной аккредитации (подбора **r**) равна 100% / 2. Аккредитация повторяется **t** раз, пока не будет достигнута требуемая вероятность 100% / 2^t, что **A** не знает закрытого ключа **s**.

Следует отметить, что приведенные выше протокол на базе RSA и схема Шнорра, по своей сути, тоже можно отнести к протоколам доказательства с нулевым разглашением. Ведь аутентифицируемая сторона доказывает знание секрета (закрытого ключа), не разглашая его. При этом процедура аутентификации носит детерминированный характер и выполняется всего за один раунд.

Сервер аутентификации Kerberos

Kerberos – программный продукт, разработанный в середине 1980-х годов в Массачусетском технологическом институте и претерпевший с тех пор ряд принципиальных изменений. Клиентские компоненты Kerberos присутствуют в большинстве современных ОС.

Kerberos предназначен для решения следующей задачи. Имеется открытая (незащищенная) сеть, в узлах которой сосредоточены субъекты – пользователи, а также клиентские и серверные программные компоненты. Каждый субъект обладает секретным ключом. Чтобы субъект **C** мог доказать свою подлинность субъекту **S** (без этого **S** не станет обслуживать **C**), он должен не только назвать себя, но и продемонстрировать знание секретного ключа. **C** не может просто послать **S** свой секретный ключ, во-первых, потому, что сеть открыта (доступна для пассивного и активного прослушивания), а, во-вторых, потому, что **S** не знает

(и не должен знать) секретный ключ **C**. Требуется менее прямолинейный способ демонстрации знания секретного ключа.

Система Kerberos представляет собой доверенную третью сторону (т.е. сторону, которой доверяют все), владеющую секретными ключами обслуживаемых субъектов и помогающую им в попарной проверке подлинности.

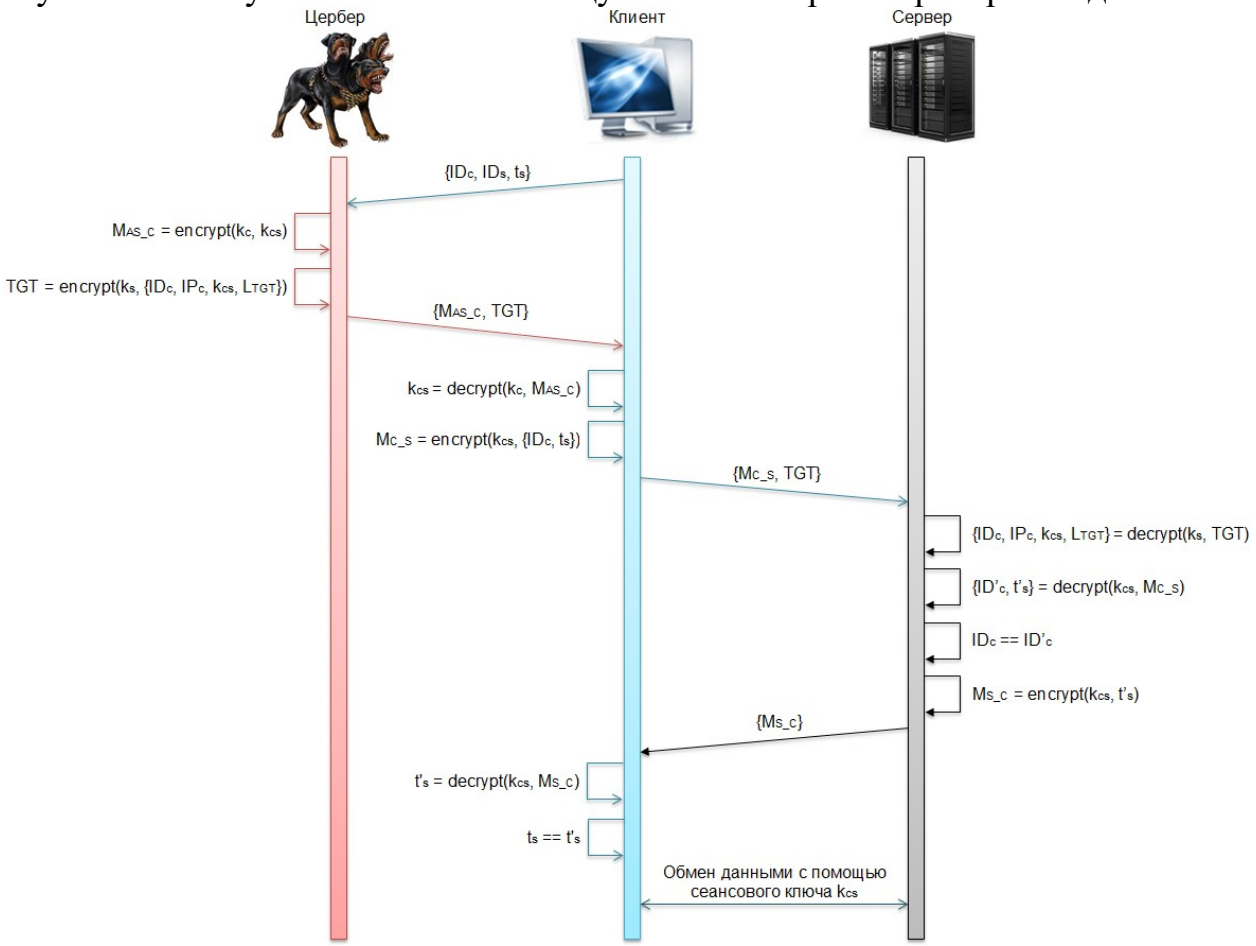


Рис.2. Проверка сервером S подлинности клиента C

Упрощенно последовательность идентификации/аутентификации с помощью Kerberos версии 5.0 на основе спецификации RFC 1510 ["Request for Comments: 1510. The Kerberos Network Authentication Service (V5)" - "Запрос для комментариев: 1510. Сетевой сервис аутентификации Kerberos"] выглядит следующим образом.

(1) Клиент посылает Kerberos запрос, содержащий свой идентификатор $\mathbf{ID_c}$ и идентификатор сервера (запрашиваемой услуги) $\mathbf{ID_s}$, а также метку времени $\mathbf{t_s}$.

(2) Kerberos:

• по метке времени $\mathbf{t_s}$ проверяет синхронизацию своих часов с часами клиента;

• формирует сообщение $\mathbf{M_{AS\_C}}$ - зашифровывает сеансовый ключ $\mathbf{k_{cs}}$ ключом клиента $\mathbf{k_c}$;

• формирует билет **TGT** - зашифровывает идентификатор клиента $\mathbf{ID_c}$, его IP-адрес $\mathbf{IP_c}$, сеансовый ключ $\mathbf{k_{cs}}$ и время жизни билета (сеансового ключа) $\mathbf{L_{TGT}}$ ключом сервера $\mathbf{k_s}$;

• отсылает сообщение $\mathbf{M_{AS\_C}}$ и билет **TGT** клиенту.

(3) Клиент:

• расшифровывает сообщение $\mathbf{M_{AS\_C}}$ с помощью своего ключа $\mathbf{k_c}$ для получения сеансового ключа $\mathbf{k_{cs}}$;

• формирует сообщение $\mathbf{M_{C\_S}}$ - зашифровывает свой идентификатор $\mathbf{ID_c}$ и метку времени $\mathbf{t_s}$ сеансовым ключом $\mathbf{k_{cs}}$;

• отсылает сообщение $\mathbf{M_{C\_S}}$ и билет **TGT** серверу.

(4) Сервер:

• расшифровывает билет **TGT** с помощью своего ключа $\mathbf{k_s}$ для получения идентификатора клиента $\mathbf{ID_c}$, его IP-адреса $\mathbf{IP_c}$, сеансового ключа $\mathbf{k_{cs}}$ и времени жизни билета $\mathbf{L_{TGT}}$;

• расшифровывает сообщение $\mathbf{M_{C\_S}}$ с помощью сеансового ключа $\mathbf{k_{cs}}$ для получения идентификатора клиента $\mathbf{ID'_c}$ и метки времени $\mathbf{t'_s}$;

• для аутентификации клиента выполняет сравнение идентификаторов $\mathbf{ID_c}$ и $\mathbf{ID'_c}$, полученных, соответственно, из билета **TGT** и сообщения $\mathbf{M_{C\_S}}$;

• формирует сообщение $\mathbf{M_{S\_C}}$ - зашифровывает метку времени $\mathbf{t'_s}$ сеансовым ключом $\mathbf{k_{cs}}$;

• отсылает сообщение $\mathbf{M_{S\_C}}$ клиенту.

(5) Клиент:

• расшифровывает сообщение $\mathbf{M_{S\_C}}$ с помощью сеансового ключа $\mathbf{k_{cs}}$ для получения метки времени $\mathbf{t'_s}$;

• для аутентификации сервера выполняет сравнение меток времени $\mathbf{t_s}$ и $\mathbf{t'_s}$.

(6) Обмен данными между клиентом и сервером выполняется с помощью сеансового ключа $\mathbf{k_{cs}}$ в течение времени жизни билета (сеансового ключа) $\mathbf{L_{TGT}}$.

Как видно из данного протокола, помимо идентификации/аутентификации, параллельно решается вопрос с обменом сеансовым ключом. Согласно RFC 1510 все ключи, включая сеансовый, используются для симметричного шифрования (в частности DES). В RFC 1510 рассмотрены дополнительные детали: применение хеширования для контроля целостности (в частности MD4 или MD5) сообщения в случае ошибочных действий сторон и т.д. В отдельных спецификациях рассмотрены расширения для Kerberos:

- RFC 3962 ["Request for Comments: 3962. Advanced Encryption Standard (AES) Encryption for Kerberos 5" - "Запрос для комментариев: 3962. Расширенный стандарт шифрования для Kerberos 5"] - применение современного стандарта США симметричного шифрования AES;

- RFC 4556 ["Request for Comments: 4556. Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)" - "Запрос для комментариев: 4556. Шифрование с открытым ключом для предварительной аутентификации в Kerberos (PKINIT)"] - применение асимметричного шифрования (в частности RSA) для предварительной аутентификации клиента перед Kerberos;

- и др.

Вопросы для самоконтроля

1. Криптоанализ.
2. Частотный анализ.
3. Метод полного перебора.
4. Атака по ключам.
5. Криптоанализ симметричных шифров.
6. Криптоанализ асимметричных шифров.
7. Криптоанализ по побочным каналам.

Практическое занятие №7. Электронная цифровая подпись (RSA, ГОСТы 34.10-94 и 34.10-2001)

Цель: изучить способы генерации и проверки ЭЦП.

Ход работы:

Необходимо ознакомиться с теоретической частью и привести последовательность выполнения процедур генерации и проверки ЭЦП с использованием следующих способов:

- на базе алгоритма RSA;
- по ГОСТ 34.10-94;
- по ГОСТ 34.10-2001.

При выполнении задания необходимо привести таблицы генерации ключей, отправки сообщения с ЭЦП и получения сообщения с ЭЦП. В качестве хеш-образа исходного сообщения h(T) принять коды, соответственно, 1-ой, 2-ой и 3-ей буквы своей фамилии согласно их положению в алфавите.

Общие сведения

Протоколы ЭЦП с одной стороны относят к протоколам аутентификации, т.к. гарантируют, что сообщение поступило от достоверного отправителя, а с другой стороны к протоколам контроля целостности, т.к. гарантируют, что сообщение пришло в неискаженном виде. Более того, получатель в дальнейшем может использовать ЭЦП как доказательство достоверности сообщения третьим лицам (арбитру) в том случае, если отправитель впоследствии попытается отказаться от него.

Электронная цифровая подпись – реквизит электронного документа, предназначенный для защиты данного документа от подделки, полученный в результате криптографического преобразования информации с использованием закрытого ключа ЭЦП и позволяющий *идентифицировать владельца* сертификата ключа подписи, а также *установить отсутствие искажения информации* в электронном документе (Федеральный закон № 1-ФЗ "Об электронной цифровой подписи" от 10.01.2002г.).

Электронная цифровая подпись – информация в электронной форме, которая присоединена к другой информации в электронной форме (подписываемой информации) или иным образом связана с такой информацией и которая используется для *определения лица, подписывающего информацию* (Федеральный закон № 63-ФЗ "Об электронной подписи" от 06.04.2011г.).

[Электронная цифровая] подпись – строка бит, полученная в результате процесса формирования подписи (ISO/IEC 14888-1:2008 "Информационные технологии. Методы защиты. Цифровые подписи с приложением. Часть 1. Общие положения" и ГОСТ Р 34.10-2012 "Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи"). По ГОСТ понятия "Электронная цифровая подпись", "электронная подпись" и "цифровая подпись" являются синонимами.

Говоря о схеме электронной подписи, обычно имеют в виду следующую **классическую ситуацию**:

- отправитель знает содержание сообщения, которое он подписывает;
- получатель, зная открытый ключ проверки подписи, может проверить правильность подписи полученного сообщения в любое время без какого-либо разрешения и участия отправителя;
- безопасность схемы подписи гарантируется.

При создании электронной подписи по классической схеме отправитель:

- применяет к исходному сообщению **T** хеш-функцию **h(T)** и получает хеш-образ сообщения **r**;
- вычисляет электронную подпись **s** по хеш-образу **r** с использованием своего закрытого ключа;
- посылает сообщение **T** вместе с электронной подписью **s** получателю.

Получатель, отделив электронную подпись от сообщения, выполняет следующие действия:

- применяет к полученному сообщению **T** хеш-функцию **h(T)** и получает хеш-образ сообщения **r**;
- расшифровывает хеш-образ **r'** из электронной подписи **s** с использованием открытого ключа отправителя;
- проверяет соответствие хеш-образов **r** и **r'** и если они совпадают, то отправитель действительно является тем, за кого себя выдает, и сообщение при передаче не подверглось искажению.

Как видно из этой схемы, порядок использования ключей обратный тому, который используется при передаче секретных сообщений. Вначале отправитель использует свой закрытый ключ, а затем получатель применяет открытый ключ отправителя.

Существует несколько схем ЭЦП, которые, как правило, применяются совместно с определенными хеш-функциями. Некоторые из них приведены в таблице.

Таблица 1. Схемы ЭЦП

Схема цифровой подписи	Задача, лежащая в основе стойкости	Хеш-функция
RSA	Разложение числа на множители	MD4 или MD5 (Message Digest Algorithm - алгоритм краткого изложения сообщения, Р. Ривест)
DSS (NIST[1]. FIPS Publication 186: Digital Signature Standard (DSS). May 1994) DSS – Федеральный стандарт цифровой подписи США	Дискретное логарифмирование по схеме Эль-Гамаля	SHA-1 (NIST. FIPS Publication 180: Secure Hash Standard (SHS). May 1993) SHS – стандарт хэш-функции США

		SHA - Secure Hash Algorithm – алгоритм хеш-функции
ECDSA (Elliptic Curve Digital Signature Algorithm) - алгоритм цифровой подписи на эллиптических кривых. Принят в качестве стандарта ISO[2] 14888-3 в 1998 г., ANSI[3] X9.62 – 1999 г., IEEE[4] 1363 – 2000 г. и NIST 186-2 – 2000 г. (последняя редакция – NIST. FIPS Publication 186-3: Digital Signature Standard (DSS). June 2009)	Дискретное логарифмирование в группе точек эллиптической кривой	SHA (NIST. FIPS 180-3: Secure Hash Standard (SHS). October 2008)
ГОСТ 34.10-94 (Информационная технология. Криптографическая защита информации. Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма)	Дискретное логарифмирование по схеме Эль-Гамаля	ГОСТ 34.11-94 (Информационная технология. Криптографическая защита информации. Функция хэширования)
ГОСТ Р 34.10-2001 (Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи)	Дискретное логарифмирование в группе точек эллиптической кривой	ГОСТ 34.11-94 (Информационная технология. Криптографическая защита информации. Функция хэширования)
ГОСТ Р 34.10-2012 (Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи)	Дискретное логарифмирование в группе точек эллиптической кривой	ГОСТ Р 34.11-2012 (Информационная технология. Криптографическая защита информации. Функция хэширования)

Примечания.

[1]NIST - Национальный Институт стандартов и технологий, США (The National Institute of Standards and Technology).

[2]ISO - Международная организация по стандартизации (International Organization for Standardization).

[3]ANSI - Американский национальный институт стандартов (American National Standards Institute).

[4]IEEE - Институт инженеров по электротехнике и электронике (Institute of Electrical and Electronics Engineers).

Протокол на базе алгоритма RSA

Этап 1. Отправитель **А** генерирует открытый {e=5, n=91} и закрытый {d=29} ключи, публикует открытый ключ.

Этап 2. Отправка сообщения и электронной подписи (выполняет отправитель **А**).

Таблица 2. Отправка сообщения и ЭЦП на базе алгоритма RSA

№ п/п	Описание операции	Пример
1	Вычисление хеш-образа **h** = h(T), где **T** – исходное сообщение; **h(T)** – хеш-функция (для MD5 длина хеш-образа 128 бит).	h = 7
2	Выработка электронной подписи **s** = h^d mod n, где **d** – закрытый ключ отправителя **А**; **n** – часть открытого ключа отправителя **А**.	s = 7^{29} mod 91 = 63
3	Отправка получателю **Б** исходного сообщения **T** и электронной подписи **s**.	

Этап 3. Получение сообщения и проверка электронной подписи (выполняет получатель **Б**).

Таблица 3. Получение сообщения и проверка ЭЦП на базе алгоритма RSA

№ п/п	Описание операции	Пример
1	Вычисление хеш-образа по полученному сообщению **h'** = h(T'), где **T'** – полученное сообщение.	h' = 7
2	Вычисление хеш-образа из электронной подписи **h** = s^e mod n, где **e** и **n** – открытый ключ отправителя **А**.	h = 63^5 mod 91 = 7
3	Если h' = h, то получатель **Б** делает вывод, что полученное сообщение T' = T и оно действительно отправлено **А**.	7 = 7

Алгоритм цифровой подписи ГОСТ 34.10-94

Алгоритм цифровой подписи ГОСТ 34.10-94 похож на DSS-94 и является вариацией схемы Эль-Гамаля.

Этап 1. Выработка ключей (выполняет отправитель **А**).

Таблица 4. Выработка ключей для ЭПЦ по ГОСТ 34.10-94

№	Описание операции	Пример

п/п		
1	Выбор простого числа **p** (по ГОСТ – $2^{509} < p < 2^{512}$ либо $2^{1020} < p < 2^{1024}$).	p = 79
2	Выбор простого числа **q**, являющегося делителем (p – 1) (по ГОСТ – $2^{254} < q < 2^{256}$).	q = 13 [(79 – 1) mod 13 = 0]
3	Выбор **a** (0 < a < p – 1), для которого $a^q \bmod p = 1$.	a = 8 [$8^{13} \bmod 79 = 1$]
4	Выбор закрытого ключа **x** (0 < x < q).	x = 4
5	Вычисление **y** = $a^x \bmod p$.	y = $8^4 \bmod 79 = 67$
6	Публикация открытого ключа {**p**, **q**, **a**, **y**}. Первые три параметра **p**, **q** и **a** - открыты и могут совместно использоваться пользователями сети, **y** – персональный открытый ключ для одного пользователя (отправителя **A**).	

Этап 2. Отправка сообщения и электронной подписи (выполняет отправитель **A**).

Таблица 5. Отправка сообщения и ЭЦП по ГОСТ 34.10-94

№ п/п	Описание операции	Пример
1	Вычисление хеш-образа **h** = h(T) (по ГОСТ длина хеш-образа 256 бит).	h = 7
2	Выбор **k** (0 < k < q).	k = 11
3	Вычисление двух значений: **w** = $a^k \bmod p$ и **w'** = w mod q (по ГОСТ длина w' 256 бит). Если w' = 0, перейти к этапу 2 и выбрать другое значение числа k.	w = $8^{11} \bmod 79 = 21$ w' = 21 mod 13 = 8
4	Вычисление **s** = (x w' + k h) mod q (по ГОСТ длина s 256 бит). Если s = 0, перейти к этапу 2 и выбрать другое значение числа k.	s = (4*8 + 11*7) mod 13 = 5
5	Отправка получателю **Б** исходного сообщения **T** и электронной подписи {**w'**, **s**}.	

Этап 3. Получение сообщения и проверка электронной подписи (выполняет получатель **Б**).

Таблица 6. Получение сообщения и проверка ЭЦП по ГОСТ 34.10-94

№ п/п	Описание операции	Пример
1	Вычисление хеш-образа по полученному сообщению **h'** = h(T'), где T' – полученное сообщение. Если T = T', то должно быть h = h'.	h' = 7
2	Вычисление **v** = h' $^{q-2}$ mod q.	v = 7^{11} mod 13 = 2
3	Вычисление двух значений: $\mathbf{z_1}$ = (s v) mod q и $\mathbf{z_2}$ = ((q – w') v) mod q.	z_1 = (5 * 2) mod 13 = 10 z_2 = ((13 – 8) * 2) mod 13 = 10
4	Вычисление **u** = ((a^{z1} * y^{z2}) mod p) mod q.	u = ((8^{10} * 67^{10}) mod 79) mod 13 = 8
5	Если w' = u, то получатель **Б** делает вывод, что полученное сообщение T' = T и оно действительно отправлено **А**.	8 = 8

Алгоритм цифровой подписи ГОСТ Р 34.10-2001 и ГОСТ Р 34.10-2012

Алгоритмы цифровой подписи ГОСТ Р 34.10-2001 и ГОСТ Р 34.10-2012 совпадают и похожи на ECDSA.

Этап 1. Отправитель **А** генерирует открытый {(A,B)=(3,7), P(x_p,y_p)=P(7,17), q=47, Q(x_q,y_q)=Q(36,20)} и закрытый {d=10} ключи, публикует открытый ключ.

Этап 2. Отправка сообщения и электронной подписи (выполняет отправитель **А**).

Таблица 7. Отправка сообщения и ЭЦП по ГОСТ 34.10-2001 (34.10-2012)

№ п/п	Описание операции	Пример
1	Вычисление хеш-образа **h** = h(T) (по ГОСТ длина хеш-образа 256 бит).	h = 7
2	Вычисление **e** = h mod q.	e = 7 mod 47 = 7
3	Выбор **k** (0 < k < q).	k = 11
4	Определение точки эллиптической кривой **C(x_c, y_c)** = k	C = 11 * P(7, 17) =

	$P(x_p, y_p)$.	(16, 16)
5	Вычисление **r** = x_c mod q. Если r = 0, перейти к этапу 2 и выбрать другое значение числа k.	r = 16 mod 47 = 16
6	Вычисление **s** = (r d + k e) mod q. Если s = 0, перейти к этапу 2 и выбрать другое значение числа k.	s = (16*10 + 11*7) mod 47 = 2
7	Отправка получателю **Б** исходного сообщения **Т** и электронной подписи (**r**, **s**).	

Этап 3. Получение сообщения и проверка электронной подписи (выполняет получатель **Б**).

Таблица 8. Получение сообщения и проверка ЭЦП по ГОСТ 34.10-2001 (34.10-2012)

№ п/п	Описание операции	Пример
1	Вычисление хеш-образа по полученному сообщению **h'** = h(T'), где T' – полученное сообщение. Если T = T', то должно быть h = h'.	h' = 7
2	Вычисление **e'** = h' mod q.	e' = 7 mod 47 = 7
3	Вычисление **v** = e'^{-1} mod q, где e'^{-1} – обратное число к e' по модулю q.	e'^{-1} = 27 [(7 * 27) mod 47 = 1] v = 27 mod 47 = 27
4	Вычисление двух значений: $\mathbf{z_1}$ = (s v) mod q и $\mathbf{z_2}$ = ((q – r) v) mod q.	z_1 = (2 * 27) mod 47 = 7 z_2 = ((47 – 16) * 27) mod 47 = 38
5	Определение точки эллиптической кривой $\mathbf{C'(x_{c'}, y_{c'})}$ = z_1 P(x_p, y_p) + z_2 Q(x_q, y_q).	C' = 7 P(7, 17) + 38 Q(36, 20) = (22, 26) + (11, 31) = (16, 16)
6	Вычисление **r'** = $x_{c'}$ mod q.	r' = 16 mod 47 = 16
7	Если r' = r, то получатель **Б** делает вывод, что полученное сообщение T' = T и оно действительно отправлено **А**.	16 = 16

Разновидности ЭЦП

Кроме классической схемы ЭЦП различают еще несколько **специальных**:

- схема "конфиденциальной" (неотвергаемой) подписи – подпись не может быть проверена без участия сгенерировавшего ее лица;

- схема подписи "вслепую" ("затемненной" подписи) - отправитель не знает подписанного им сообщения;

- схема "мультиподписи" - вместо одного отправителя сообщение подписывает группа из нескольких участников;

- схема "групповой" подписи - получатель может проверить, что подписанное сообщение пришло от члена некоторой группы отправителей, но не знает, кем именно из членов группы оно подписано. В то же время, в случае необходимости, отправитель может быть определен;

- и др.

Юридические основания использования ЭЦП

10 января 2002 г. Президент Российской Федерации В.В. Путин подписал Федеральный закон № 1-ФЗ "Об электронной цифровой подписи". **Цель Федерального закона № 1-ФЗ** - обеспечение правовых условий использования ЭЦП в электронных документах, при соблюдении которых ЭЦП в электронном документе признается равнозначной собственноручной подписи в документе на бумажном носителе.

В настоящий момент действует Федеральный закон № 63-ФЗ "Об электронной подписи" от 06.04.2011 г. **Сфера действия (цель) Федерального закона № 63-ФЗ** - 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 системах, где число пользователей исчисляется сотнями и тысячами, для проверки ЭЦП используются так называемые сертификаты ЭЦП (ЭП).

Сертификат ЭЦП – открытый ключ с некоторой дополнительной информацией о его владельце (регистрационный номер сертификата, ФИО владельца, срок действия и т.д.), подписанный ключом **Центра сертификации** (ЦС, Certificate Authority, CA, Удостоверяющий центр, УЦ).

В Федеральном законе "Об электронной подписи" даны следующие определения.

Сертификат ключа проверки электронной подписи – электронный документ или документ на бумажном носителе, выданные УЦ либо доверенным лицом УЦ и подтверждающие принадлежность ключа проверки ЭП владельцу сертификата ключа проверки ЭП.

Квалифицированный сертификат ключа проверки электронной подписи – сертификат ключа проверки ЭП, выданный аккредитованным УЦ или доверенным лицом аккредитованного УЦ либо федеральным органом исполнительной власти, уполномоченным в сфере использования ЭП.

При получении документа, подписанного ЭЦП, вначале подается запрос в ЦС, который высылает сертификат ЭЦП, информацию об окончании срока его действия или информацию об отсутствии сертификата. Если ЦС выслал сертификат, то считается, что документ послал именно тот, кто указан в

сертификате. Для автоматизации деятельности ЦС применяется системы, называемые **системы поддержки инфраструктуры открытых ключей** (Public Key Infrastructure, PKI).

Впервые ссуда под ЭЦП (на покупку дома) была выдана в США 25 июля 2000г.

Вопросы для самоконтроля

1. Компьютерный вирус.
2. Червь.
3. Троянские программы.
4. Логические бомбы.
5. Файловые вирусы.
6. Стелс вирус.
7. Полиморфные вирусы.
8. Антивирусные программы.

Практическое занятие №8. Контроль целостности (биты четности, контрольные цифры)

Цель: изучить способы контроля целостности данных.

Ход работы:

Необходимо ознакомиться с теоретической частью и определить контрольные данные с использованием следующих способов:

- битов четности. В качестве исходных данных принять битовое представление букв фамилии в соответствии с кодировкой Windows 1251;

Буква	Битовая строка	Паритетный бит	
		четный (even)	нечетный (odd)

- контрольных цифр. В качестве исходных данных принять необходимое количество цифр (за исключением контрольной) из строки, состоящей из кодов букв фамилии, имени и отчества согласно их положению в алфавите:
- по алгоритму Луна (15 цифр);
- для штрихкода по стандарту EAN-13 (12 цифр);
- для ИНН физического лица (10 цифр);
- для кодов станций на железнодорожном транспорте (5 цифр);

Общие сведения

Как было отмечено в первой лекции, целостность является одним из трех ключевых свойств информации (доступность, целостность и конфиденциальность). При этом под **целостностью** понимается свойство информации, характеризующее ее устойчивость к случайному или преднамеренному разрушению или несанкционированному изменению.

Рассмотрим некоторые способы и методы контроля целостности.

Проверка четности

Представляет собой самый простой способ обеспечения целостности при хранении или передаче данных. Битовая строка (обычно длиной 7-8 бит), контроль которой необходимо выполнить, дополняется одним, так называемым **паритетным битом** (англ. parity bit). Существует две разновидности проверки четности: с **четным (even)** и **нечетным (odd)** паритетным битом. В первом случае при записи или пересылке данных паритетный бит устанавливается равным 1, если количество единиц в контролируемой строке нечетное, и 0 – если четное. В случае нечетного паритетного бита поступают наоборот.

Таблица 1. Примеры установки бита четности

Битовая строка	Паритетный бит	
	четный (even)	нечетный (odd)
1100 1011	1	0

1001 1001	0	1
1111 1111	0	1
0000 0000	0	1

Недостатки:

- исправление ошибки невозможно;

- в случае изменения состояния четного количества бит (например, двух), вычисленный паритетный бит совпадет с записанным. Т.е. ошибка не будет обнаружена. В то же время, согласно статистики, приблизительно 90% всех ошибок памяти происходит именно с одиночным разрядом. Таким образом, проверки четности бывает достаточно для большинства ситуаций.

Использование контрольных цифр

В отличие от предыдущего способа для контроля целостности используется не бит, а цифра. Обычно, контролируемый набор цифр вначале по определенным правилам складывается, а затем берется остаток от деления по модулю, который и является контрольной цифрой. Ниже рассматриваются некоторые системы кодирования с использованием контрольной цифры:

- алгоритм Луна;
- штрихкод по стандарту EAN-13;
- заграничный паспорт гражданина РФ с биометрическими данными;
- идентификационный номер налогоплательщика;
- коды станций на железнодорожном транспорте.

Алгоритм Луна (англ. Luhn algorithm) - алгоритм вычисления контрольной цифры в соответствии со стандартом ISO/IEC 7812 «Идентификационные карты. Идентификация эмитентов». Алгоритм разработан сотрудником фирмы IBM Гансом Питером Луном в 1954 г. Используется для подсчета контрольной цифры:

- номеров всех банковских карт;
- номеров некоторых дисконтных карт;
- кодов полисов обязательного медицинского страхования;
- единого 8-значного номера железнодорожного вагона на РЖД;
- IMEI-кодов (англ. International Mobile Equipment Identity - международный идентификатор мобильного оборудования);
- ICCID-кодов (англ. Integrated Circuit Card ID - идентификатор карты с интегрированной микросхемой);
- т.д.

а) банковская карта

б) полис медицинского страхования

в) цистерна

г) мобильный телефон (IMEI)

д) SIM-карта (ICCID)

Рис.1. Номера и коды с контрольными цифрами

В следующей таблице приведен порядок вычисления контрольной цифры на примере кода полиса медицинского страхования (рис. 1б).

Таблица 2. Вычисление контрольной цифры по алгоритму Луна
(если количество цифр, включая контрольную, в коде четное)

№ п/п	Описание операции	Пример
1	Каждая из цифр, стоящая в нечетной позиции, умножается на 2, после чего вычисляется остаток от деления на 9.	(2 * 2) mod 9 = 4 (5 * 2) mod 9 = 1 (6 * 2) mod 9 = 3 (0 * 2) mod 9 = 0 (4 * 2) mod 9 = 8 (0 * 2) mod 9 = 0 (0 * 2) mod 9 = 0 (1 * 2) mod 9 = 2
2	Вычисляется сумма остатков $\mathbf{S_н}$.	$S_н$ = 4 + 1 + 3 + 0 + 8 + 0 + 0 + 2 = 18
3	Вычисляется сумма цифр $\mathbf{S_ч}$, стоящих в четных позициях, за исключением последней (контрольной).	$S_ч$ = 7 + 8 + 2 + 8 + 2 + 0 + 2 = 29
4	Вычисляется контрольная (последняя) цифра **cd** из	cd = 3

	уравнения ($S_н$ + $S_ч$ + cd) mod 10 = 0.	[(18 + 29 + 3) mod 10 = 0]

Если количество цифр в коде нечетное (например, для IMEI-кодов), то 1 и 2 операция выполняются для цифр, стоящих в четных позициях, 3 операция – для цифр, стоящих в нечетных позициях.

Штрихкод по стандарту EAN-13 - одна из вариаций Европейского стандарта штрихкода, предназначенного для кодирования идентификатора товара и производителя. Регламентируется ГОСТ ИСО/МЭК 15420-2001 «Автоматическая идентификация. Кодирование штриховое. Спецификация символики EAN/UPC (ЕАН/ЮПиСи)».

Рис.2. Штрихкод EAN-13

В следующей таблице приведен порядок вычисления контрольной цифры по стандарту EAN-13.

Таблица 3

№ п/п	Описание операции	Пример
1	Вычисляется сумма цифр **$S_н$**, стоящих в нечетных позициях, за исключением последней (контрольной).	$S_н$ = 5 + 0 + 2 + 4 + 2 + 4 = 17
2	Вычисляется утроенная сумма цифр **$S_ч$**, стоящих в четных позициях.	$S_ч$ = 3 * (9 + 1 + 3 +1 + 3 + 5) = 66
3	Вычисляется контрольная (последняя) цифра **cd** из уравнения ($S_н$ + $S_ч$ + cd) mod 10 = 0.	cd = 7 [(17 + 66 + 7) mod 10 = 0]

В России с 2009 г. во всех субъектах РФ действуют пункты выдачи паспортно-визовых документов нового поколения - **заграничных паспортов гражданина РФ с биометрическими данными**.

В документе используются различные способы защиты, в т.ч. защита целостности за счет контрольных цифр. В пластиковой странице с фотографией владельца и встроенным внутри чипом имеется т.н. машиносчитываемая зона (МСЗ).

Рис.3. Машиносчитываемая зона паспорта
(синим контуром выделены контрольные цифры)

В нижней строке МСЗ используются пять контрольных цифр. Ее структура приведена в следующей таблице.

Таблица 4. Структура нижней строки МСЗ

Контрольная цифра	Позиции знаков, используемые для расчета контрольных цифр	Позиция контрольной цифры
Номер паспорта	1-9	10
Дата рождения	14-19	20
Дата истечения срока действия паспорта	22-27	28
Личный номер	29-42	43
Заключительная контрольная цифра	1-10, 14-20, 22-43 (позиции 11-13 и 23 исключаются из расчета)	44

Алгоритм расчета контрольных цифр заключается в перемножении каждой цифры соответствующего элемента данных на весовой показатель повторяющейся числовой последовательности "731 731 ...", суммировании полученных произведений и взятии остатка от деления на 10. Если в элементе данных встречаются буквы латинского алфавита, то при расчете они заменяются на числа от 10 (A) до 35 (Z); знак "<" соответствует 0. Ниже приведен пример расчета контрольных цифр.

Таблица 5. Пример расчета контрольных цифр нижней строки МСЗ

Назначение	Номер паспорта									КЦ				Дата рождения						КЦ		Дата истечения срока действия						КЦ	Личный номер			КЦ	КЦ
№ позиции	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	...	42	43	44
Нижн	7	2	5	4	3	6	2	3	9	**5**	R	U	S	7	3	0	1	0	7	**2**	M	2	3	0	7	1	2	**7**	<	..	<	**0**	**2**

яя строка МСЗ																													.			
Весовой показатель	7	3	1	7	3	1	7	3	1	7			7	3	1	7	3	1	7		7	3	1	7	3	1	7	7	.. .	3	1	
Расчет контрольной цифры в элементе данных	(7*7+2*3+5*1+4*7+3*3+6*1+2*7+3*3+9*1) mod 10 = 135 mod 10 = **5**												(7*7+3*3+0*1+1*7+0*3+7*1) mod 10 = 72 mod 10 = **2**								(2*7+3*3+0*1+7*7+1*3+2*1) mod 10 = 77 mod 10 = **7**								(0*7+ … +0*3) mod 10 = **0**			
Расчет заключительной контрольной цифры	[(135 + 5*7) + (72 + 2*7) + (77 + 7*7) + (0 + 0*1)] mod 10 = 382 mod 10 = **2**																															

Идентификационный номер налогоплательщика (ИНН) - уникальный идентификатор, присваиваемый юридическому или физическому лицу для учета уплаты налогов в Российской Федерации. При постановке на налоговый учет подотчетному лицу выдается свидетельство, в котором указывается его ИНН.

Министерство Российской Федерации по налогам и сборам

СВИДЕТЕЛЬСТВО

о постановке на учет в налоговом органе физического лица по месту жительства на территории Российской Федерации

Настоящее Свидетельство выдано в соответствии с положением части первой Налогового кодекса Российской Федерации, принятого Федеральным законом от 31 июля 1998 года N146-ФЗ, физическому лицу [redacted]
(фамилия, имя, отчество)

пол муж.

дата рождения 7 января 1973 года
(число, месяц, год)

место рождения БЕЛОГОРСК Г,,,643
(указывается в точном соответствии с записью в документе, удостоверяющем личность)

и подтверждает постановку физического лица на учет 22 марта 2000 года
(число,месяц,год постановки на учет)

В ИНСПЕКЦИИ МНС РОССИИ ПО ЖЕЛЕЗНОДОРОЖНОМУ РАЙОНУ г.ХАБАРОВСКА 2724
(наименование государственной налоговой инспекции и ее код)

Идентификационный номер налогоплательщика (ИНН) 272407031790

Дата выдачи Свидетельства 22 марта 2000 года
(число,месяц,год)

Свидетельство применяется во всех предусмотренных законодательством случаях и предъявляется вместе с документом, удостоверяющим личность физического лица и место его жительства на территории Российской Федерации .

Свидетельство подлежит замене в случае переезда физического лица на новое место жительства на территорию, подведомственную другой государственной налоговой инспекции, изменения приведенных в нем сведений, а также в случае порчи, утери.

Руководитель инспекции МНС России по Железнодорожному району г.Хабаровска Л.Г.Бурковская
(подпись, фамилия, имя, отчество)

МП

серия 27 № 0042973

Рис.4. Свидетельство о постановке на учет

Контрольная (контрольные) цифра ИНН определяется по следующим формулам:

- для десятизначного ИНН юридического лица:

$$n_{10} = ((2n_1 + 4n_2 + 10n_3 + 3n_4 + 5n_5 + 9n_6 + 4n_7 + 6n_8 + 8n_9) \bmod 11) \bmod 10; \quad (1)$$

- для двенадцатизначного ИНН физического лица:

$$n_{11} = ((7n_1 + 2n_2 + 4n_3 + 10n_4 + 3n_5 + 5n_6 + 9n_7 + 4n_8 + 6n_9 + 8n_{10}) \bmod 11) \bmod 10, \quad (2)$$

$$n_{12} = ((3n_1 + 7n_2 + 2n_3 + 4n_4 + 10n_5 + 3n_6 + 5n_7 + 9n_8 + 4n_9 + 6n_{10} + 8n_{11}) \bmod 11) \bmod 10, \quad (3)$$

где n_i - i-ая цифра ИНН.

Для ИНН физического лица, отображенного на рис. 4, контрольные цифры:

$$n_{11} = ((7*2 + 2*7 + 4*2 + 10*4 + 3*0 + 5*7 + 9*0 + 4*3 + 6*1 + 8*7) \bmod 11) \bmod 10 = (185 \bmod 11) \bmod 10 = 9 \bmod 10 = 9,$$

n_{12} = ((3*2 + 7*7 + 2*2 + 4*4 + 10*0 + 3*7 + 5*0 + 9*3 + 4*1 + 6*7 + 8*9) mod 11) mod 10 = (241 mod 11) mod 10 = 10 mod 10 = 0.

Коды станций на железнодорожном транспорте. В информационных системах железнодорожного транспорта приняты различные способы кодирования станций. В **АСУЖТ** используется код станции, состоящий из 6 цифр ($n_1n_2n_3n_4n_5n_6$). Последняя цифра кода (n_6) является контрольной и определяется по следующей формуле:

$$n_6 = (1n_1 + 2n_2 + 3n_3 + 4n_4 + 5n_5) \bmod 11. \quad (4)$$

Если остаток от деления меньше 10, то он является контрольной цифрой, иначе выполняют сдвиг весового ряда на две позиции и вычисления повторяют:

$$n_6 = (3n_1 + 4n_2 + 5n_3 + 6n_4 + 7n_5) \bmod 11. \quad (5)$$

Если новый остаток от деления вновь получится равным 10, то контрольная цифра принимается равной 0, иначе - остатку, вычисленному по формуле 5.

Первые четыре цифры АСУЖТ для станций, открытых для грузовых операций, называют кодом **Единой сетевой разметки (ЕСР)**. Вариация кода ЕСР с контрольной цифрой состоит из 5 знаков ($n_1n_2n_3n_4n_5$), последний из которых (n_5) определяется точно также, как и для кода станции в АСУЖТ. Отличие заключается в использовании сокращенных весовых рядов (1, 2, 3, 4) и (3, 4, 5, 6). Т.к. пятая цифра для грузовых станций в АСУЖТ принимается равной 0, то контрольные цифры кодов станций АСУЖТ и ЕСР совпадают. В частности, код станции Хабаровск-1 Дальневосточной железной дороги:

- АСУЖТ: код - 970406, контрольная цифра - n_6 = (1*9 + 7*2 + 0*3 + 4*4 + 5*0) mod 11 = 39 mod 11 = 6;
- ЕСР: код - 97046, контрольная цифра - n_5 = (1*9 + 7*2 + 0*3 + 4*4) mod 11 = 39 mod 11 = 6.

Вопросы для самоконтроля

1. Брандмауэр.
2. Firewall.
3. Аппаратная защита информации.
4. Виды аппаратных средств защиты.
5. Защита от НСД.

Практическое занятие №9. Тайные многосторонние вычисления и разделение секрета

Цель: изучить способы шифрования данных различными перестановками.

Ход работы:

Необходимо ознакомиться с теоретической частью и привести последовательность выполнения следующих протоколов:

- тайных многосторонних вычислений для расчета средней величины трех чисел. В качестве исходных данных принять коды 1-ой, 2-ой и 3-ей буквы своей фамилии согласно их положению в алфавите;
- разбиения секрета с использование гаммирования для трех участников. В качестве секрета принять первые 3 буквы фамилии, для гамм - любые трехбуквенные сочетания;
- разделения секрета по схеме Шамира для (3, 5)-пороговой схемы. В качестве секрета **S** принять код 1-ой буквы своей фамилии согласно ее положению в алфавите;
- разделения секрета по схеме Асмута-Блума для (3, 5)-пороговой схемы. В качестве секрета **S** принять код 1-ой буквы своей фамилии согласно ее положению в алфавите.

При выполнении задания необходимо привести исходные данные и таблицы, содержащие последовательность выполнения протоколов.

Тайные многосторонние вычисления

Суть протокола заключается в определении результата некоторых преобразований над информацией, которой владеют несколько людей (субъектов). При этом они не хотят раскрывать ее для других.

Пример протокола.

Задача – определение средней заработной платы трех сотрудников: Директора (Д), Секретаря (С) и Уборщицы (У).

Заработная плата, которую сотрудники не хотят раскрывать друг другу:

- Директор – 10 у.е.;
- Секретарь – 5 у.е.;
- Уборщица – 2 у.е.

Пары ключей асимметричного шифрования (RSA) для обмена информацией между собой:

- Директор: открытый ключ – $e_д = 5$ и $n_д = 91$; закрытый ключ - $d_д = 29$;
- Секретарь: открытый ключ – $e_c = 7$ и $n_c = 91$; закрытый ключ - $d_c = 31$;
- Уборщица: открытый ключ – $e_y = 17$ и $n_y = 91$; закрытый ключ - $d_y = 17$.

В рассмотренном примере подбор ключей некорректен, т.к. выбор простых чисел для определения модуля должен выполняться каждым индивидуально. Данное упрощение принято в целях рассмотрения работы протокола.

Таблица 1. Протокол тайных многосторонних вычислений

№ п/п	Описание операции	Пример

1	**У** прибавляет случайное секретное число **x** к сумме своей зарплаты, шифрует результат с помощью открытого ключа **С** и отсылает его **С**.	x = 3 $(2 + 3)^7 \bmod 91 = 47$
2	**С** расшифровывает результат, добавляет к нему свою зарплату, шифрует результат с помощью открытого ключа **Д** и отсылает его **Д**.	$47^{31} \bmod 91 = 5$ $(5 + 5)^5 \bmod 91 = 82$
3	**Д** расшифровывает результат, добавляет к нему свою зарплату, шифрует результат с помощью открытого ключа **У** и отсылает его **У**.	$82^{29} \bmod 91 = 10$ $(10 + 10)^{17} \bmod 91 = 76$
4	**У** расшифровывает результат, отнимает **x** и объявляет среднюю зарплату.	$76^{17} \bmod 91 = 20$ СЗ = (20 - 3)/3 = 5.(6)

В данной схеме поочередное шифрование/дешифрование необходимо для предотвращения вычисления мошенником информации, которой владеют участники протокола. Если пересылаемую информацию не шифровать, то мошенник, зная функцию преобразования и перехватив информацию, направленную одному из участников и высланную им следующему, может легко вычислить скрываемые исходные данные. Например, противник или недобросовестный участник протокола, перехватив информацию, высланную уборщицей секретарю (5) и секретарем директору (10), определит зарплату секретаря 10 – 5 = 5 у.е. Шифрование предотвращает подобную атаку.

Протоколы разбиения и разделения секрета

Сущность данных протоколов заключается в том, что владелец секрета распределяет его части (доли) между несколькими людьми (субъектами). Каждая часть сама по себе ничего не значит и не дает информации о секрете. Для восстановления секрета требуется собрать все или определенное количество его долей. В первом случае говорят о **разбиении** (англ. splitting), а во втором о **разделении** (англ. sharing) секрета.

Второе отличие протоколов разбиения от протоколов разделения заключается в распределении долей. При разбиении разные доли передаются разным людям, при разделении – один человек может владеть сразу несколькими долями.

Примерами подобных ситуаций может быть сохранение в тайне до определенного момента рецепта изготовления продукта, сущности изобретения, кодов запуска баллистических ракет и т.п.

Разбиение секрета с использование гаммирования

Простую и в то же время эффективную схему разбиения секрета можно построить на базе гаммирования по модулю 2. В этом случае секрет вначале кодируется в двоичном виде. Для его разбиения владелец генерирует несколько битовых строк (гамм), которые отдельно передает каждому из участников протокола. Кроме этого он складывает по модулю 2 битовое представление секрета со всеми гаммами и результат (шифрограмму) выкладывает в доступное для участников место. Для восстановления секрета необходимо сложить шифрограмму со всеми выданными гаммами, при чем не важно, в каком порядке.

В следующей таблице представлен пример разбиения секрета между тремя участниками.

Таблица 2. Протокол разбиения секрета на трех участников

№ п/п	Описание операции	Пример				
1	Кодирование секрета – слово «КОД».	Секрет	Буква	К	О	Д
			Bin-код	1100 1010	1100 1110	1100 0100
2	Генерация случайных гамм и передача их участникам.	$Гамма_1$	Буква	Ю	Л	Я
			Bin-код	1101 1110	1100 1011	1101 1111
		$Гамма_2$	Буква	Ш	А	Р
			Bin-код	1101 1000	1100 0000	1101 0000
		$Гамма_3$	Буква	С	У	К
			Bin-код	1101 0001	1101 0011	1100 1010
3	Получение шифрограммы и выкладывание её в доступное для участников место.	Секрет $Гамма_1$ $Гамма_2$ $Гамма_3$	⊕	1100 1010 1101 1110 1101 1000 1101 0001	1100 1110 1100 1011 1100 0000 1101 0011	1100 0100 1101 1111 1101 0000 1100 1010
		Шифрограмма	Bin-код	0001 1101	0001 0110	0000 0001
4	Восстановление секрета.	Шифрограмма $Гамма_2$ $Гамма_1$	⊕	0001 1101 1101	0001 0110 1100	0000 0001 1101

		Гамма$_3$		1000 1101 1110 1101 0001	0000 1100 1011 1101 0011	0000 1101 1111 1100 1010
		Секрет	Bin-код	1100 1010	1100 1110	1100 0100
			Буква	К	О	Д

Примечание. Бинарное представление символов в соответствии с кодировкой Windows 1251.

Для восстановления секрета участники протокола, которым выданы гаммы, должны собраться вместе и в любой последовательности сложить свои гаммы с шифрограммой. В том случае, если гаммирование выполняется поочередно и обмен информации между участниками выполняется по открытым каналам связи (например, первый участник складывает шифрограмму со своей гаммой и отсылает результат второму, второй складывает полученный результат со своей гаммой и отсылает третьему и т.д.), то пересылаемую между ними информацию необходимо шифровать, как при тайных многосторонних вычислениях. В противном случае мошенник может определить гаммы участников протокола.

В общем случае, в основе протокола разбиения секрета лежит шифрование с разными ключами (например, как в рассмотренном примере или тройном DES) или последовательное применение разных методов шифрования (например, как в комбинированных шифрах).

Разделение секрета по схеме Шамира (интерполяционных полиномов Лагранжа)

В отличие от разбиения секрета участнику (субъекту) может передаваться сразу несколько равных долей и для восстановления секрета необязательно иметь все доли. Например, код запуска баллистической ракеты разбивается на пять долей, которые передаются трем полковникам (по одной доле) и одному генералу (две доли). Если для восстановления кода запуска (секрета) необходимо собрать три доли, то это могут сделать три полковника или генерал с одним полковником. Такая схема, где секрет делится на **n** долей, а для его восстановления необходимо собрать не менее чем **m** долей, где m < n, называется **(m, n)–пороговой схемой**.

В 1979 г. Ади Шамир (один из авторов RSA) предложил протокол разделения секрета с использованием полиномов (многочленов), максимальная степень которых равна m-1. Для восстановления секрета используются формулы интерполяционного полинома Лагранжа.

Интерполяционный полином Лагранжа.

Пусть имеется некоторая исходная функция f(x), с помощью которой определены m точек – пар (x_i, y_i). Тогда можно подобрать полином степени m-1, который будет проходить через все точки и максимально близко описывать исходную функцию.

Интерполяционный полином L(x) определяется формулой

$$f(x) \approx L(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + a_2x^2 + a_1x + a_0 = \sum_{i=0}^{m-1} y_i l_i(x),$$

(1)

где a_i - коэффициенты интерполяционного полинома Лагранжа; y_i - значения исходной функции в i-ой точке; $l_i(x)$ - базисные полиномы, определяемые по формуле

$$l_i(x) = \prod_{j=0, j\neq i}^{m-1} \frac{x - x_j}{x_i - x_j}, \quad (2)$$

где x_i, x_j - значения аргумента функции в i-ой и j-ой точках.

Пример подбора интерполяционного полинома Лагранжа.

Исходная функция $f(x) = \mathrm{Sin}(x^2)$.

Точки исходной функции:

- $x_0 = -2$, $f(x_0) = -0.7568$;
- $x_1 = -1$, $f(x_1) = -0.8415$;
- $x_2 = 0$, $f(x_2) = 0$;
- $x_3 = 1$, $f(x_3) = 0.8415$;
- $x_4 = 2$, $f(x_4) = 0.7568$.

Определение базисных полиномов:

$$l_0(x) = \prod_{j=1}^{4} \frac{x - x_j}{x_0 - x_j} = \frac{x - x_1}{x_0 - x_1} \cdot \frac{x - x_2}{x_0 - x_2} \cdot \frac{x - x_3}{x_0 - x_3} \cdot \frac{x - x_4}{x_0 - x_4} = \frac{x+1}{-2+1} \cdot \frac{x-0}{-2-0} \cdot \frac{x-1}{-2-1} \cdot \frac{x-2}{-2-2} = \frac{x+1}{-1} \cdot \frac{x}{-2} \cdot \frac{x-1}{-3} \cdot \frac{x-2}{-4} = \frac{1}{24} \cdot (x^4 - 2x^3 - x^2 + 2x)$$

$$l_1(x) = \prod_{j=0, j\neq 1}^{4} \frac{x - x_j}{x_1 - x_j} = \frac{x - x_0}{x_1 - x_0} \cdot \frac{x - x_2}{x_1 - x_2} \cdot \frac{x - x_3}{x_1 - x_3} \cdot \frac{x - x_4}{x_1 - x_4} = \frac{x+2}{-1+2} \cdot \frac{x-0}{-1-0} \cdot \frac{x-1}{-1-1} \cdot \frac{x-2}{-1-2} = \frac{x+2}{1} \cdot \frac{x}{-1} \cdot \frac{x-1}{-2} \cdot \frac{x-2}{-3} = \frac{1}{-6} \cdot (x^4 - x^3 - 4x^2 + 4x)$$

$$l_2(x) = \prod_{j=0, j\neq 2}^{4} \frac{x - x_j}{x_2 - x_j} = \frac{x - x_0}{x_2 - x_0} \cdot \frac{x - x_1}{x_2 - x_1} \cdot \frac{x - x_3}{x_2 - x_3} \cdot \frac{x - x_4}{x_2 - x_4} = \frac{x+2}{0+2} \cdot \frac{x+1}{0+1} \cdot \frac{x-1}{0-1} \cdot \frac{x-2}{0-2} = \frac{x+2}{2} \cdot \frac{x+1}{1} \cdot \frac{x-1}{-1} \cdot \frac{x-2}{-2} = \frac{1}{-4} \cdot (x^4 - 5x^2 + 4)$$

$$l_3(x) = \prod_{j=0, j\neq 3}^{4} \frac{x - x_j}{x_3 - x_j} = \frac{x - x_0}{x_3 - x_0} \cdot \frac{x - x_1}{x_3 - x_1} \cdot \frac{x - x_2}{x_3 - x_2} \cdot \frac{x - x_4}{x_3 - x_4} = \frac{x+2}{1+2} \cdot \frac{x+1}{1+1} \cdot \frac{x-0}{1-0} \cdot \frac{x-2}{1-2} = \frac{x+2}{3} \cdot \frac{x+1}{2} \cdot \frac{x}{1} \cdot \frac{x-2}{-1} = \frac{1}{-6} \cdot (x^4 + x^3 - 4x^2 - 4x)$$

$$l_4(x) = \prod_{j=0}^{3} \frac{x - x_j}{x_4 - x_j} = \frac{x - x_0}{x_4 - x_0} \cdot \frac{x - x_1}{x_4 - x_1} \cdot \frac{x - x_2}{x_4 - x_2} \cdot \frac{x - x_3}{x_4 - x_3} = \frac{x+2}{2+2} \cdot \frac{x+1}{2+1} \cdot \frac{x-0}{2-0} \cdot \frac{x-1}{2-1} = \frac{x+2}{4} \cdot \frac{x+1}{3} \cdot \frac{x}{2} \cdot \frac{x-1}{1} = \frac{1}{24} \cdot (x^4 + 2x^3 - x^2 - 2x)$$

.

Определение интерполяционного полинома Лагранжа:

$$L(x) = \sum_{i=0}^{m-1} y_i l_i(x) = \frac{y_0}{24} \cdot (x^4 - 2x^3 - x^2 + 2x) + \frac{y_1}{-6} \cdot (x^4 - x^3 - 4x^2 + 4x) + \frac{y_2}{-4} \cdot (x^4 - 5x^2 + 4) + \frac{y_3}{-6} \cdot (x^4 + x^3 - 4x^2 - 4x) + \frac{y_4}{24} \cdot (x^4 + 2x^3 - x^2 - 2x) =$$

$$\frac{-0.7568}{24} \cdot (x^4 - 2x^3 - x^2 + 2x) + \frac{0.8415}{-6} \cdot (x^4 - x^3 - 4x^2 + 4x) + \frac{0}{-4} \cdot (x^4 - 5x^2 + 4) + \frac{0.8415}{-6} \cdot (x^4 + x^3 - 4x^2 - 4x) + \frac{-0.7568}{24} \cdot (x^4 + 2x^3 - x^2 - 2x) = -0.3436x^4 + 1.1851x^2$$

.

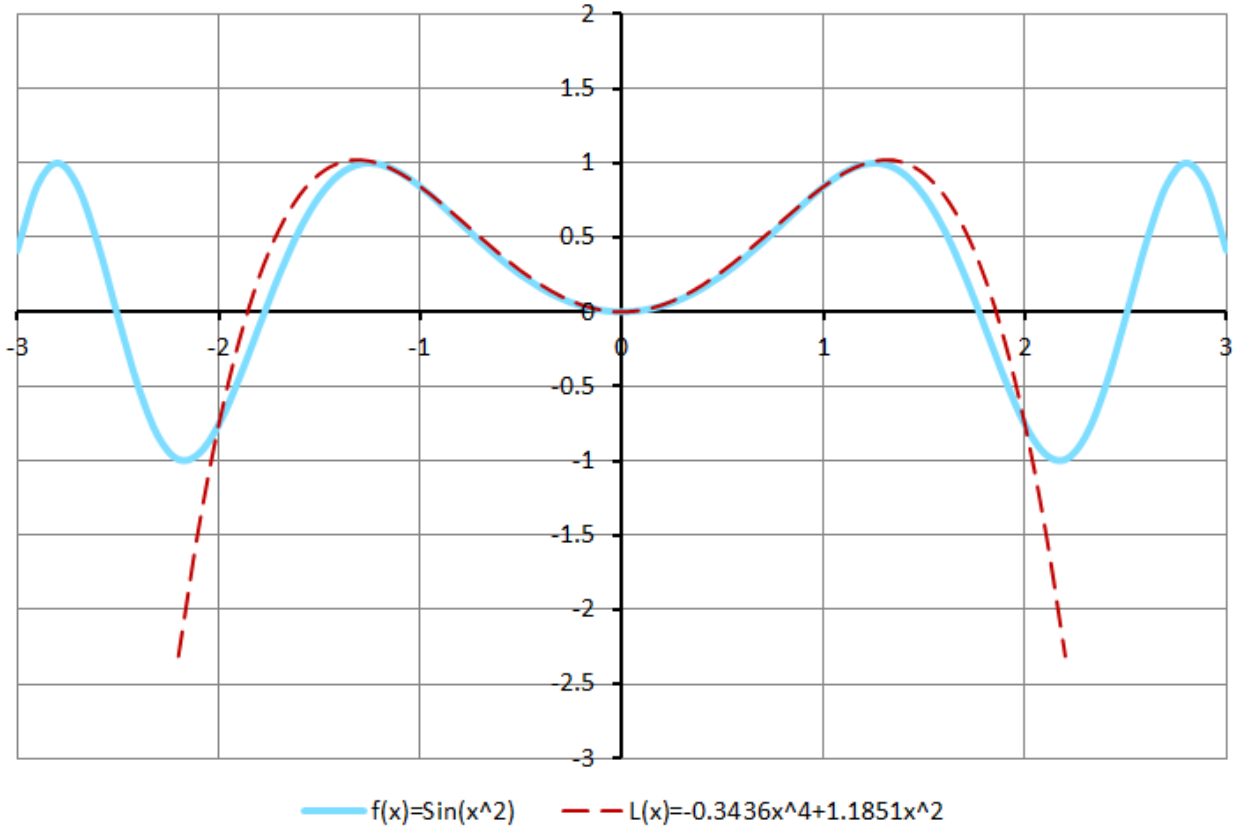


Рис.1. Графики исходной функции и интерполяционного полинома

Как видно на рисунке, интерполяционный полином довольно хорошо аппроксимирует исходную функцию в диапазоне x ∈ [-2; 2].

Протокол разделения секрета на основе интерполяционных полиномов Лагранжа.

Для разделения секрета S, восстанавливаемого с помощью m долей, используется полином степени m-1 по модулю p

$$f(x) = L(x) = (a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + a_2x^2 + a_1x + S) \bmod p,$$

(3)

где f(x) = L(x) - исходная функция и интерполяционный полином Лагранжа;
a_i - целочисленные коэффициенты полинома;
$S = a_0$ - разделяемый секрет, закодированный в виде числа;
p - простое число.

Коэффициенты полинома a_i выбираются произвольно, за исключением $a_0 = S$. Модуль p должен быть простым числом, большим секрета S и общего количества долей n. Владелец секрета для $x_i = 1..n$ определяет значения полинома $y_i = f(x_i)$ и передает пары (x_i, y_i) участникам согласно определенному для каждого количеству долей. Для восстановления секрета необходимо собрать m долей (пар (x_i, y_i)) и найти значения коэффициентов интерполяционного полинома, включая секрет $S = a_0$.

Т.к. исходная функция и интерполяционный полином выражаются одинаковой формулой, то определенные по известным точкам (долям) коэффициенты интерполяционного полинома совпадут с коэффициентами исходной функции.

Пример протокола.
Секрет S = 11.
Количество долей, необходимых для восстановления секрета, m = 3.
Общее количество долей n = 5.

Таблица 3. Процедура определения и распределения долей (выполняет владелец)

№ п/п	Описание операции	Пример
1	Выбор простого числа **p**, которое больше количества долей **n** и секрета **S**.	p = 59
2	Выбор произвольного многочлена степени **m-1** : **f(x)** = $(a_2x^2 + a_1x + S) \bmod p$, где значения $\mathbf{a_2}$ и $\mathbf{a_1}$ выбираются случайным образом, хранятся в тайне и отбрасываются после распределения долей.	$a_2 = 10$, $a_1 = 23$ $f(x) = (10x^2 + 23x + 11) \bmod 59$
3	Определение долей ($\mathbf{x_i}$, $\mathbf{y_i}$), где $\mathbf{y_i} = f(x_i)$ и $\mathbf{x_i} = i + 1$.	$y_0 = (10*1^2 + 23*1 + 11) \bmod 59 = 44$ $y_1 = (10*2^2 + 23*2 + 11) \bmod 59 = 38$ $y_2 = (10*3^2 + 23*3 + 11) \bmod 59 = 52$ $y_3 = (10*4^2 + 23*4 + 11) \bmod 59 = 27$ $y_4 = (10*5^2 + 23*5 + 11) \bmod 59 = 22$
4	Публикация **p** и распределение долей ($\mathbf{x_i}$, $\mathbf{y_i}$) между участниками.	p = 59 $(x_0, y_0) = (1, 44)$ $(x_1, y_1) = (2, 38)$ $(x_2, y_2) = (3, 52)$ $(x_3, y_3) = (4, 27)$ $(x_4, y_4) = (5, 22)$

Таблица 4. Процедура восстановления секрета (выполняют участники)

№ п/п	Описание операции	Пример
1	Сбор **m** долей.	$(x_1, y_1) = (2, 38)$ $(x_2, y_2) = (3, 52)$ $(x_4, y_4) = (5, 22)$

2	Определение базисных полиномов.	$l_1(x) = \frac{x-3}{2-3} \cdot \frac{x-5}{2-5} = \frac{x-3}{-1} \cdot \frac{x-5}{-3} = \frac{1}{3} \cdot (x^2 - 8x + 15)$ $l_2(x) = \frac{x-2}{3-2} \cdot \frac{x-5}{3-5} = \frac{x-2}{1} \cdot \frac{x-5}{-2} = \frac{1}{-2} \cdot (x^2 - 7x + 10)$ $l_4(x) = \frac{x-2}{5-2} \cdot \frac{x-3}{5-3} = \frac{x-2}{3} \cdot \frac{x-3}{2} = \frac{1}{6} \cdot (x^2 - 5x + 6)$
3	Определение интерполяционного полинома Лагранжа.	$L(x) = \left[\frac{38}{3} \cdot (x^2 - 8x + 15) + \frac{52}{-2} \cdot (x^2 - 7x + 10) + \frac{22}{6} \cdot (x^2 - 5x + 6)\right] mo$ $L(x) = \left[\frac{76}{6} \cdot (x^2 - 8x + 15) - \frac{156}{6} \cdot (x^2 - 7x + 10) + \frac{22}{6} \cdot (x^2 - 5x + 6)\right] m$ $L(x) = \left[\frac{1}{6} \cdot (-58x^2 + 374x - 288)\right] \ mod\ 59$
4	Определение обратного числа по модулю $\mathbf{b^{-1}}$ для дробного множителя полинома **1 / b**.	$\frac{1}{b} = \frac{1}{6}$ $b^{-1} = 10\ [(6 * 10) \bmod 59 = 1]$
5	Замена дробного множителя **1 / b** и умножение коэффициентов полинома на множитель $\mathbf{b^{-1}}$.	$L(x) = [10 * (-58x^2 + 374x - 288)] \bmod 59 = (-580x^2 + 3740x - 2880) \bmod 59$
6	Приведение коэффициентов полинома и определение секрета **S**.	$a_2 = -580 \bmod 59 = -49 \bmod 59 = 10$ $a_1 = 3740 \bmod 59 = 23$ $S = a_0 = -2880 \bmod 59 = -48 \bmod 59 = 11$ $L(x) = (10x^2 + 23x + 11) \bmod 59$

Примечание. Обратное число по модулю определяется из выражения $(b * b^{-1}) \bmod p = 1$ (например, с помощью расширенного алгоритма Евклида).

Как отмечает Брюс Шнайер применительно к пороговой схеме (6, n):

Наиболее впечатляющим моментом совместного использования секрета является то, что, если коэффициенты выбраны случайным образом, пять человек даже при помощи бесконечных вычислительных мощностей не смогут узнать ничего, кроме длины сообщения (которая и так им известна). Это также безопасно, как одноразовый блокнот. Попытка выполнить исчерпывающий поиск (т.е., перебор всех возможных шести коэффициентов) покажет, что любое возможное сообщение останется секретным.

Брюс Шнайер. «Практическая криптография»

Разделение секрета по схеме Асмута-Блума

В (m, n)-пороговой схеме Асмута-Блума для распределения долей используются простые[1] и взаимно простые числа[2], а для восстановления - китайская теорема об остатках.

Схема и пример протокола.

Секрет S = 11.

Количество долей, необходимых для восстановления секрета, m = 3.

Общее количество долей n = 5.

Таблица 5. Процедура определения и распределения долей (выполняет владелец)

№ п/п	Описание операции	Пример
1	Выбор простого числа **p**, которое больше секрета **S**.	p = 13
2	Выбор **n** взаимно простых чисел $\mathbf{d_i}$, удовлетворяющих условиям: - $d_i > p$; - $d_i < d_{i+1}$; - $d_1 * d_2 * \ldots * d_m < p * d_{n-m+2} * d_{n-m+3} * \ldots * d_n$.	$d_i \in \{17, 20, 23, 29, 37\}$ 17 * 20 * 23 < 13 * 29 * 37 7820 < 13949
3	Выбор произвольного числа **r**, удовлетворяющего условию $r < \frac{\prod_{i=1}^{m} d_i - S}{p}$. Вычисление **S'** = S + r p.	r = 30 $\left[30 < 600.7 = \frac{17 \cdot 20 \cdot 23 - 11}{13}\right]$ S' = 11 + 30 * 13 = 401
4	Определение долей ($\mathbf{d_i}$, $\mathbf{k_i}$), где $\mathbf{k_i}$ = S' mod d_i.	k_1 = 401 mod 17 = 10 k_2 = 401 mod 20 = 1 k_3 = 401 mod 23 = 10 k_4 = 401 mod 29 = 24 k_5 = 401 mod 37 = 31
5	Публикация **p** и распределение долей ($\mathbf{d_i}$, $\mathbf{k_i}$) между участниками.	p = 13 (d_1, k_1) = (17, 10) (d_2, k_2) = (20, 1) (d_3, k_3) = (23, 10) (d_4, k_4) = (29, 24) (d_5, k_5) = (37, 31)

Сущность **китайской теоремы об остатках** заключается в определении некоторого числа S' по набору его остатков k_i от деления на некоторые заданные взаимно простые числа d_i.

$$\begin{cases} S' \bmod d_1 = k_1 \\ S' \bmod d_2 = k_2 \\ \dots \\ S' \bmod d_m = k_m \end{cases}. \quad (4)$$

Например, для трех пар (d_i, k_i) – (3, 1), (5, 3) и (8, 3) – таким числом является S’ = 43.

$$\begin{cases} 43 \bmod 3 = 1 \\ 43 \bmod 5 = 3 \\ 43 \bmod 8 = 3 \end{cases}.$$

В следующей таблице, наряду с процедурой восстановления секрета, приведен алгоритм определения числа S’ (китайской теоремы об остатках) в пп. 2-5.

Таблица 6. Процедура восстановления секрета
(выполняют участники)

№ п/п	Описание операции	Пример
1	Сбор **m** долей.	(d_2, k_2) = (20, 1) (d_3, k_3) = (23, 10) (d_5, k_5) = (37, 31)
2	Вычисление произведения **D** взаимно простых чисел $\mathbf{d_j}$.	D = 20 * 23 * 37 = 17020
3	Вычисление сомножителей $\mathbf{D_j} = D / d_j$.	D_1 = 17020 / 20 = 851 D_2 = 17020 / 23 = 740 D_3 = 17020 / 37 = 460
4	Определение обратных чисел $\mathbf{D_j^{-1}}$ по модулям $\mathbf{d_j}$.	D_1^{-1} = 11 [(851 * 11) mod 20 = 1] D_2^{-1} = 6 [(740 * 6) mod 23 = 1] D_3^{-1} = 7 [(460 * 7) mod 37 = 1]
5	Вычисление **S’** = (Σ k_j D_j D_j^{-1}) mod D.	S’ = (1*851*11 + 10*740*6 + 31*460*7) mod 17020 = 153581 mod 17020 = 401
6	Определение секрета **S** = S’ mod p.	S = 401 mod 13 = 11

Примечание. Обратное число по модулю определяется из выражения ($b * b^{-1}$) mod p = 1 (например, с помощью расширенного алгоритма Евклида).

[1]**Простое число** – натуральное число, большее единицы и не имеющее других натуральных делителей, кроме самого себя и единицы.

[2]**Взаимно простые числа** – числа, не имеющие общих делителей, кроме 1, т.е. наибольший общий делитель которых равен 1.

Другие разновидности схем разделения секрета

Кроме классической пороговой схемы разделения секрета (владелец раздает доли и при восстановлении участники раскрывают свои доли) существуют и другие схемы.

1. Разделение секрета без владельца. Участники могут создать секрет и разделить его на доли так, что никто из них не узнает секрета, пока они совместно его не восстановят.

2. Разделение секрета без раскрытия долей при восстановлении. В этом смысле протокол представляет собой нечто среднее между разделением секрета и тайными многосторонними вычислениями.

3. Разделение секрета с возможностью проверки корректности отдельных долей. Каждый из участников независимо от других может проверить корректность своей доли без восстановления секрета.

4. Разделение секрета с возможностью блокирования восстановления секрета. Каждый из участников получает две доли: «да» и «нет». Если при восстановлении секрета число долей «нет» превышает некоторое пороговое значение, то его восстановление невозможно, даже, если количества долей «да» достаточно.

5. Разделение секрета с возможностью блокирования долей. Если после распределения долей, некоторые из участников теряют доверие, то можно блокировать их доли.

6. Разделение секрета с возможностью выявления фальшивых долей. При восстановлении секрета возможно выявление участников, предоставивших фальшивые доли.

7. Групповое разделение секрета. Секрет распределяется среди участников, объединенных в k групп. Для восстановления секрета необходимо собрать в каждой группе определенное количество долей. Т.е. имеет место $((m_1, n_1), (m_2, n_2), ..., (m_k, n_k))$-пороговая схема.

Вопросы для самоконтроля

1. Методы аутентификации.
2. Дактилоскопия.
3. Динамические методы биометрической аутентификации.
4. Верификация подписи.
5. Защита биометрических данных.

Практическое занятие №10. Представление чисел в двоичном виде
Цель: изучить способы представления чисел в двоичном виде.
Ход работы:

Необходимо ознакомиться с теоретической частью и представить:
- первых три числа в прямом, обратном и дополнительном двоичном коде;
- четвертое число в двоичном формате с плавающей запятой, включающем знаки порядка и мантиссы. В отчете привести исходное число в нормализованной научной записи по основанию 2 и порядок получения дробной части в двоичном виде;
- пятое число в двоичном формате одинарной точности (single) по стандарту IEEE 754.

Варианты заданий выбрать согласно таблице.

№ варианта	Исходные числа для представления				
	в прямом, обратном и дополнительном коде			в формате с плавающей запятой, включающего знаки порядка и мантиссы	в формате одинарной точности по стандарту IEEE 754
1	101	-102	-103	104	-104
2	102	-103	-104	105	-105
3	103	-104	-105	106	-106
4	104	-105	-106	107	-107
5	105	-106	-107	108	-108
6	106	-107	-108	109	-109
7	107	-108	-109	110	-110
8	108	-109	-110	111	-111
9	109	-110	-111	112	-112
10	110	-111	-112	113	-113
11	111	-112	-113	114	-114
12	112	-113	-114	115	-115
13	113	-114	-115	116	-116
14	114	-115	-116	117	-117
15	115	-116	-117	118	-118
16	116	-117	-118	119	-119

17	117	-118	-119	120	-120
18	118	-119	-120	121	-121
19	119	-120	-121	122	-122
20	120	-121	-122	123	-123
21	121	-122	-123	124	-124
22	122	-123	-124	125	-125
23	123	-124	-125	126	-126
24	124	-125	-126	127	-127
25	125	-126	-127	128	-128

Кодирование информации

Кодирование – представление информации в альтернативном виде. По своей сути кодовые системы (или просто коды) аналогичны шифрам однозначной замены, в которых элементам кодируемой информации соответствуют кодовые обозначения. Отличие заключается в том, что в шифрах присутствует переменная часть (ключ), которая для определенного исходного сообщения при одном и том же алгоритме шифрования может выдавать разные шифртексты. В кодовых системах переменной части нет. Поэтому одно и то же исходное сообщение при кодировании, как правило, всегда выглядит одинаково. Другой отличительной особенностью кодирования является применение кодовых обозначений (замен) целиком для слов, фраз или чисел (совокупности цифр). Замена элементов кодируемой информации кодовыми обозначениями может быть выполнена на основе соответствующей таблицы (наподобие таблицы шифрозамен) либо определена посредством функции или алгоритма кодирования.

В качестве **элементов кодируемой информации** могут выступать:

- буквы, слова и фразы естественного языка;
- различные символы, такие как знаки препинания, арифметические и логические операции, операторы сравнения и т.д. Следует отметить, что сами знаки операций и операторы сравнения – это кодовые обозначения;
- числа;
- аудиовизуальные образы;
- ситуации и явления;
- наследственная информация;
- и т.д.

Кодовые обозначения могут представлять собой:

- буквы и сочетания букв естественного языка;
- числа;
- графические обозначения;
- электромагнитные импульсы;
- световые и звуковые сигналы;

- набор и сочетание химических молекул;
- и т.д.

Кодирование может выполняться в **целях**:

- удобства хранения, обработки и передачи информации (как правило, закодированная информация представляется более компактно, а также пригодна для обработки и передачи автоматическими программно-техническими средствами);
- удобства информационного обмена между субъектами;
- наглядности отображения;
- идентификации объектов и субъектов;
- сокрытия секретной информации;
- и т.д.

Кодирование информации бывает **одно-** и **многоуровневым**. Примером одноуровневого кодирования служат световые сигналы, подаваемые светофором (красный – стой, желтый – приготовиться, зеленый – вперед). В качестве многоуровневого кодирования можно привести представление визуального (графического) образа в виде файла фотографии. Вначале визуальная картинка разбивается на составляющие элементарные элементы (пикселы), т.е. каждая отдельная часть визуальной картинки кодируется элементарным элементом. Каждый элемент представляется (кодируется) в виде набора элементарных цветов (RGB: англ. red – красный, green – зеленый, blue – синий) соответствующей интенсивностью, которая в свою очередь представляется в виде числового значения. Впоследствии наборы чисел, как правило, преобразуются (кодируются) с целью более компактного представления информации (например, в форматах jpeg, png и т.д.). И наконец, итоговые числа представляются (кодируются) в виде электромагнитных сигналов для передачи по каналам связи или областей на носителе информации. Следует отметить, что сами числа при программной обработке представляются в соответствии с принятой системой кодирования чисел.

Кодирование информации может быть **обратимым** и **необратимым**. При обратимом кодировании на основе закодированного сообщения можно однозначно (без потери качества) восстановить кодируемое сообщение (исходный образ). Например, кодирование с помощью азбуки Морзе или штрихкода. При необратимом кодировании однозначное восстановление исходного образа невозможно. Например, кодирование аудиовизуальной информации (форматы jpg, mp3 или avi) или хеширование.

Различают **общедоступные** и **секретные** системы кодирования. Первые используются для облегчения информационного обмена, вторые – в целях сокрытия информации от посторонних лиц.

Представление чисел в двоичном виде (в компьютере). Как известно, информация, хранящаяся и обрабатываемая в компьютерах, представлена в двоичном виде. **Бит** (англ. **bi**nary digi**t** - двоичное число; также игра слов: англ. bit - кусочек, частица) - единица измерения количества информации, равная одному разряду в двоичной системе счисления. С помощью бита можно закодировать (представить, различать) два состояния (0 или 1; да или нет). Увеличивая

количество битов (разрядов), можно увеличить количество кодируемых состояний. Например, для байта (англ. byte), состоящего из 8 битов, количество кодируемых состояний составляет $2^8 = 256$.

Числа кодируются в т.н. форматах с фиксированной и плавающей запятой.

1. Формат с фиксированной запятой, в основном, применяется для целых чисел, но может применяться и для вещественных чисел, у которых фиксировано количество десятичных знаков после запятой. Для целых чисел подразумевается, что «запятая» находится справа после младшего бита (разряда), т.е. вне разрядной сетки. В данном формате существуют два представления: беззнаковое (для неотрицательных чисел) и со знаком.

Для **беззнакового** представления все разряды отводятся под представление самого числа. Например, с помощью байта можно представить беззнаковые целые числа от 0_{10} до 255_{10} (00000000_2 - 11111111_2) или вещественные числа с одним десятичным знаком от 0.0_{10} до 25.5_{10} (00000000_2 - 11111111_2).
Для **знакового** представления, т.е. положительных и отрицательных чисел, старший разряд отводится под знак (0 – положительное число, 1 – отрицательное).

Различают прямой, обратный и дополнительный коды записи знаковых чисел.

В **прямом** коде запись положительного и отрицательного числа выполняется так же, как и в беззнаковом представление (за исключение того, что старший разряд отводится под знак). Таким образом, числа 5_{10} и -5_{10} записываются, как 00000101_2 и 10000101_2. В прямом коде имеются два кода числа 0: «положительный нуль» 00000000_2 и «отрицательный нуль» 10000000_2.

При использовании **обратного** кода отрицательное число записывается в виде инвертированного положительного числа (0 меняются на 1 и наоборот). Например, числа 5_{10} и -5_{10} записываются, как 00000101_2 и 11111010_2. Следует отметить, что в обратном коде, как и в прямом, имеются «положительный нуль» 00000000_2 и «отрицательный нуль» 11111111_2. Применение обратного кода позволяет вычесть одно число из другого, используя операцию сложения, т.е. вычитание двух чисел X – Y заменяется их суммой X + (-Y). При этом используются два дополнительных правила:

- вычитаемое число инвертируется (представляется в виде обратного кода);
- если количество разрядов результата получается больше, чем отведено на представление чисел, то крайний левый разряд (старший) отбрасывается, а к результату добавляется 1_2.

В следующей таблице приведены примеры вычитания.

Таблица 1. Примеры вычитания двух чисел с использованием обратного кода

X – Y	5 – 5	6 – 5	5 – 6	5 – (-6)
X_2	00000101	00000110	00000101	00000101
Y_2	00000101	00000101	00000110	11111001
Замена сложением	5 + (-5)	6 + (-5)	5 + (-6)	5 + 6
Обратный код для вычитаемого	11111010	11111010	11111001	00000110

($-Y_2$)				
Сложение	00000101 + 11111010 11111111	00000110 + 11111010 100000000	00000101 + 11111001 11111110	00000101 + 00000110 00001011
Отбрасывание старшего разряда и добавление 1_2	не требуется	00000000 + 00000001 00000001	не требуется	не требуется
Результат	-0	1	-1	11

Несмотря на то, что обратный код значительно упрощает вычислительные процедуры, а соответственно и быстродействие компьютеров, наличие двух «нулей» и другие условности привели к появлению **дополнительного** кода. При представлении отрицательного числа его модуль вначале инвертируется, как в обратном коде, а затем к инверсии сразу добавляется 1_2.

В следующей таблице приведены некоторые числа в различном кодовом представлении.

Таблица 2. Представление чисел в различных кодах

Десятичное представление	Код двоичного представления (8 бит)		
	прямой	обратный	дополнительный
127	01111111	01111111	01111111
6	00000110	00000110	00000110
5	00000101	00000101	00000101
1	00000001	00000001	00000001
0	00000000	00000000	00000000
-0	10000000	11111111	---
-1	10000001	11111110	11111111
-5	10000101	11111010	11111011
-6	10000110	11111001	11111010
-127	11111111	10000000	10000001
-128	---	---	10000000

При представлении отрицательных чисел в дополнительных кодах второе правило несколько упрощается - если количество разрядов результата получается

больше, чем отведено на представление чисел, то только отбрасывается крайний левый разряд (старший).

Таблица 3. Примеры вычитания двух чисел с использованием дополнительного кода

X – Y	5 – 5	6 – 5	5 – 6	5 – (-6)
X_2	00000101	00000110	00000101	00000101
Y_2	00000101	00000101	00000110	11111010
Замена сложением	5 + (-5)	6 + (-5)	5 + (-6)	5 + 6
Дополнительный код для вычитаемого ($-Y_2$)	11111011	11111011	11111010	00000110
Сложение	00000101 + 11111011 00000000	00000110 + 11111011 100000001	00000101 + 11111010 11111111	00000101 + 00000110 00001011
Отбрасывание старшего разряда и добавление 1_2	не требуется	00000001	не требуется	не требуется
Результат	-0	1	-1	11

Можно возразить, что представление чисел в дополнительных кодах требует на одну операцию больше (после инверсии всегда требуется сложение с 1_2), что может и не потребоваться в дальнейшем, как в примерах с обратными кодами. В данном случае срабатывает известный «принцип чайника». Лучше сделать процедуру линейной, чем применять в ней правила «Если А то В» (даже если оно одно). То, что с человеческой точки зрения кажется увеличением трудозатрат (вычислительной и временной сложности), с точки зрения программно-технической реализации может оказаться эффективней.

Еще одно из преимуществ дополнительного кода перед обратным заключается в возможности представления в единице информации на одно число (состояние) больше, за счет исключения «отрицательного нуля». Поэтому, как правило, диапазон представления (хранения) для знаковых целых чисел длиной один байт составляет от +127 до -128.

2. Формат с плавающей запятой, в основном, используется для вещественных чисел. Число в данном формате представляется в экспоненциальном виде

$X = e^n * m,$ (1)

где e - основание показательной функции;

n - порядок основания;

e^n - характеристика числа;

m - мантисса (лат. mantissa - прибавка) – множитель, на который надо умножить характеристику числа, чтобы получить само число.

Например, число десятичное число 350 может быть записано, как $3.5 * 10^2$, $35 * 10^1$, $350 * 10^0$ и т.д. В **нормализованной научной записи**, порядок **n** выбирается такой, чтобы абсолютная величина **m** оставалась не меньше единицы, но строго меньше десяти ($1 \leq |m| < 10$). Таким образом, в нормализованной научной записи число 350 выглядит, как $3.5 * 10^2$. При отображении чисел в программах, учитывая, что основание равно 10, их записывают в виде **m E ± n**, где E означает «*10^» («…умножить на десять в степени…»). Например, число 350 – 3.5E+2, а число 0.035 – 3.5E-2.

Так как числа хранится и обрабатывается в компьютерах в двоичном виде, то для этих целей принимается e = 2. Одной из возможных форм двоичного представления чисел с плавающей запятой является следующая.

$b_{n\pm}$	b_{n7}	b_{n6}	b_{n5}	b_{n4}	b_{n3}	b_{n2}	b_{n1}	$b_{m\pm}$	b_{m15}	b_{m14}	b_{m13}	b_{m12}	b_{m11}	b_{m10}	b_{m9}	b_{m8}	b_{m7}	b_{m6}	b_{m5}	b_{m4}	b_{m3}	b_{m2}	b_{m1}
знак	величина							знак	величина														
порядок								мантисса															

Рис. 1. Двоичный формат представления чисел с плавающей запятой

Биты bn± и bm±, означающие знак порядка и мантиссы, кодируются аналогично числам с фиксированной запятой: для положительных чисел «0», для отрицательных – «1». Значение порядка выбирается таким образом, чтобы величина целой части мантиссы в десятичном (и соответственно в двоичном) представлении равнялась «1», что будет соответствовать нормализованной записи для двоичных чисел. Например, для числа 350_{10} порядок $n = 8_{10} = 001000_2$ ($350 = 1.3671875 * 2^8$), а для 576_{10} – $n = 9_{10} = 001001_2$ ($576 = 1.125 * 2^9$). Битовое представление величины порядка может быть выполнено в прямом, обратном или дополнительном коде (например, для $n = 8_{10}$ бинарный вид 001000_2). Величина мантиссы отображает дробную часть. Для ее преобразования в двоичный вид, она последовательно умножается на 2, пока не станет равной 0. Например,

0	.3671875
x	2
0	.734375
x	2
1	.46875
x	2
0	.9375
x	2
1	.875
x	2
1	.75
x	2
1	.5
x	2
1	.0

Рис. 2. Пример получения дробной части в бинарном виде

Целые части, получаемые в результате последовательного перемножения, и представляют собой двоичный вид дробной части ($0.3671875_{10} = 0101111_2$). Оставшаяся часть разрядов величины мантиссы заполняется 0. Таким образом, итоговый вид числа 350 в формате с плавающей запятой с учетом представления мантиссы в нормализованной записи

0	0 0 0 1 0 0 0	0	0 1 0 1 1 1 1 0 0 0 0 0 0 0 0
знак +	величина 2^8	знак +	величина (1) + 0.3671875
порядок		мантисса	

Рис. 3. Двоичный вид числа 350

В программно-аппаратных реализациях арифметических действий широко распространен стандарт представления чисел с плавающей точкой **IEEE 754** (последняя редакция «754-2008 - IEEE Standard for Floating-Point Arithmetic»). Данный стандарт определяет форматы с плавающими запятыми для представления чисел **одинарной** (англ. single, float) и **двойной** (англ. double) точности. Общая структура форматов

$b_{n\pm}$	b_{ni}	...	b_{n1}	b_{mj}	...	b_{m1}
знак мантиссы	порядок			величина мантиссы		

Рис. 4. Общий формат представления двоичных чисел в стандарте IEEE 754

Форматы представления отличаются количеством бит (байт), отводимым для представления чисел, и, соответственно, точностью представления самих чисел.

Таблица 4. Характеристики форматов представления двоичных чисел в стандарте IEEE 754

Формат	single	double
Общий размер, бит (байт)	32 (4)	64 (8)
Число бит для порядка	8	11
Число бит для мантиссы (без учета знакового бита)	23	52
Величина порядка	2^{128} .. 2^{-127} ($\pm 3.4 * 10^{38}$ .. $1.7 * 10^{-38}$)	2^{1024} .. 2^{-1023} ($\pm 1.8 * 10^{308}$ .. $9.0 * 10^{-307}$)
Смещение порядка	127	1023
Диапазон представления чисел	$\pm 1.4 * 10^{-45}$ .. $3.4 * 10^{38}$	$\pm 4.9 * 10^{-324}$ .. $1.8 * 10^{308}$

(без учета знака)		
Количество значащих цифр числа (не более)	8	16

Особенностью представления чисел по стандарту IEEE является отсутствие бита под знак порядка. Несмотря на это, величина порядка может принимать как положительные значения, так и отрицательные. Этот момент учитывается т.н. «смещением порядка». После преобразования двоичного вида порядка (записанного в прямом коде) в десятичный от полученной величины отнимается «смещение порядка». В результате получается «истинное» значения порядка числа. Например, если для числа одинарной точности указан порядок 11111111_2 (= 255_{10}), то величина порядка на самом деле 128_{10} (= 255_{10} - 127_{10}), а если 00000000_2 (= 0_{10}), то -127_{10} (= 0_{10} - 127_{10}).

Величина мантиссы указывается, как и в предыдущем случае, в нормализованном виде.

С учетом вышеизложенного, число 350_{10} в формате одинарной точности стандарта IEEE 754 записывается следующим образом.

0	1 0 0 0 0 1 1 1 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
знак мантиссы +	порядок 2^8 (8 = 135 - 127)	величина мантиссы (1) + 0.3671875

Рис. 5. Двоичный вид числа 350 по стандарту IEEE

К другим особенностям стандарта IEEE относится возможность представления специальных чисел. К ним относятся значения NaN (англ. Not a Number - не число) и +/-INF (англ. Infinity - бесконечность), получающихся в результате операций типа деления на ноль. Также сюда попадают денормализованные числа, у которых мантисса меньше единицы.

В заключение по числам с плавающей запятой несколько слов о пресловутой «**ошибке округления**». Т.к. в двоичной форме представления числа хранится только несколько значащих цифр, она не может «покрыть» все многообразие вещественных чисел в заданном диапазоне. В результате, если число невозможно точно представить в двоичной форме, оно представляется ближайшим возможным. Например, если к числу типа double «0.0» последовательно добавлять «1.7», то можно обнаружить следующую «картину» изменения значений.

0.0
1.7
3.4
5.1
6.8

8.5
10.2
11.899999999999999
13.599999999999998
15.299999999999997
16.999999999999996
18.699999999999996
20.399999999999995
22.099999999999994
23.799999999999994
25.499999999999993
27.199999999999992
28.89999999999999
30.59999999999999
32.29999999999999
33.99999999999999
35.699999999999996
37.4
39.1
40.800000000000004
42.50000000000001
44.20000000000001
45.90000000000001
47.600000000000016
…

Рис. 6. Результат последовательного добавления числа 1.7 (Java 7)

Другой нюанс обнаруживается при сложении двух чисел, у которых значительно отличается порядок. Например, результатом сложения $10^{10} + 10^{-10}$ будет 10^{10}. Даже если последовательно триллион (10^{12}) раз добавлять 10^{-10} к 10^{10}, то результат останется прежним 10^{10}. Если же к 10^{10} добавить произведение $10^{-10} * 10^{12}$, что с математической точки зрения одно и то же, результат станет 10000000100 ($1.0000000100 * 10^{10}$).

Вопросы для самоконтроля

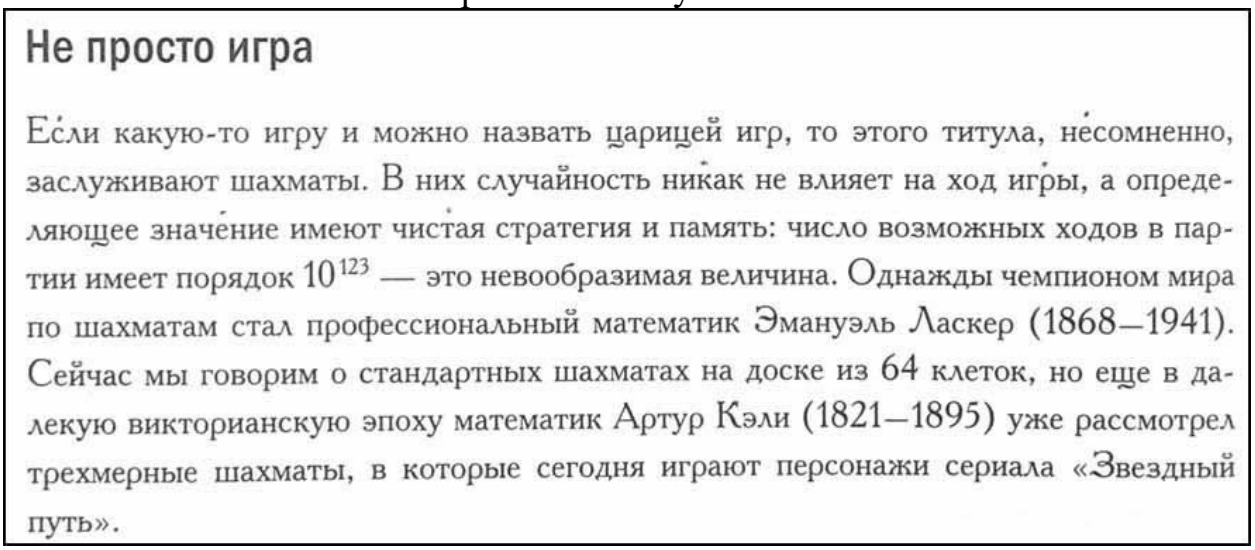
1. Идентификация.
2. Аутентификация.
3. Авторизация.
4. Многофакторная аутентификация.
5. Однофакторная двухэтапная аутентификация.

Практическое занятие №11. Стеганография
Цель: изучить методы стеганографии.
Ход работы:

1. Описание стеганографического метода

Компьютерные стеганографические методы как самостоятельно, так и совместно с криптографией, получили широкое распространение в целях защиты конфиденциальной информации. В лабораторной работе рассматривается стеганографическое сокрытие секретных сообщений в текстовых документах редактора Microsoft Word за счет специфического форматирования символов текста. Принципы сокрытия базируются на других известных стеганографических методах.

1. Микроточки. Использование микроточек для передачи секретных сообщений описал греческий ученый Эней Тактик в сочинении «Об обороне укреплённых мест». Суть предложенного им так называемого «книжного шифра» заключалась в прокалывании малозаметные дырок в книге или в другом документе над буквами секретного сообщения. Во время Первой мировой войны германские шпионы использовали аналогичный шифр, заменив дырки на точки, наносимые симпатическими чернилами на буквы газетного текста.



Не просто игра

Е́сли какую-то игру и можно назвать царицей игр, то этого титула, не́сомненно, заслуживают шахматы. В них случайность ни́как не влияет на ход игры́, а определяющее значе́ние имеют чиста́я стратегия и память: число возможных ходов в партии имеет порядок 10^{123} — это невообразимая величина. Однажды чемпионом мира по шахматам стал профессиональный математик Эмануэль Ласкер (1868–1941). Сейчас мы говорим о стандартных шахматах на доске из 64 клеток, но еще в далекую викторианскую эпоху математик Артур Кэли (1821–1895) уже рассмотрел трехмерные шахматы, в которые сегодня играют персонажи сериала «Звездный путь».

Рис.1. Сокрытие сообщения «секрет» в тексте за счет малозаметных точек (Хоакин Наварро. Тайная жизнь чисел. Мир математики – том 31)

По аналогии с микроточками скрываемая в тексте секретная информация специальным образом помечается (форматируется).

2. Использование особенностей человеческого зрения. Подобные методы широко используются для сокрытия информации в мультимедийных файлах (в частности, метод LSB, Least Significant Bit - наименьший значащий бит) за счет их избыточности. По аналогии с ними, в обычном тексте символы, составляющие секретное сообщение, могут форматируются так, что это будет незаметно для глаза неискушенного читателя текста. В частности, символы секретного сообщения могут выделяться другим цветом, незначительно отличающегося от цвета остальных символов.

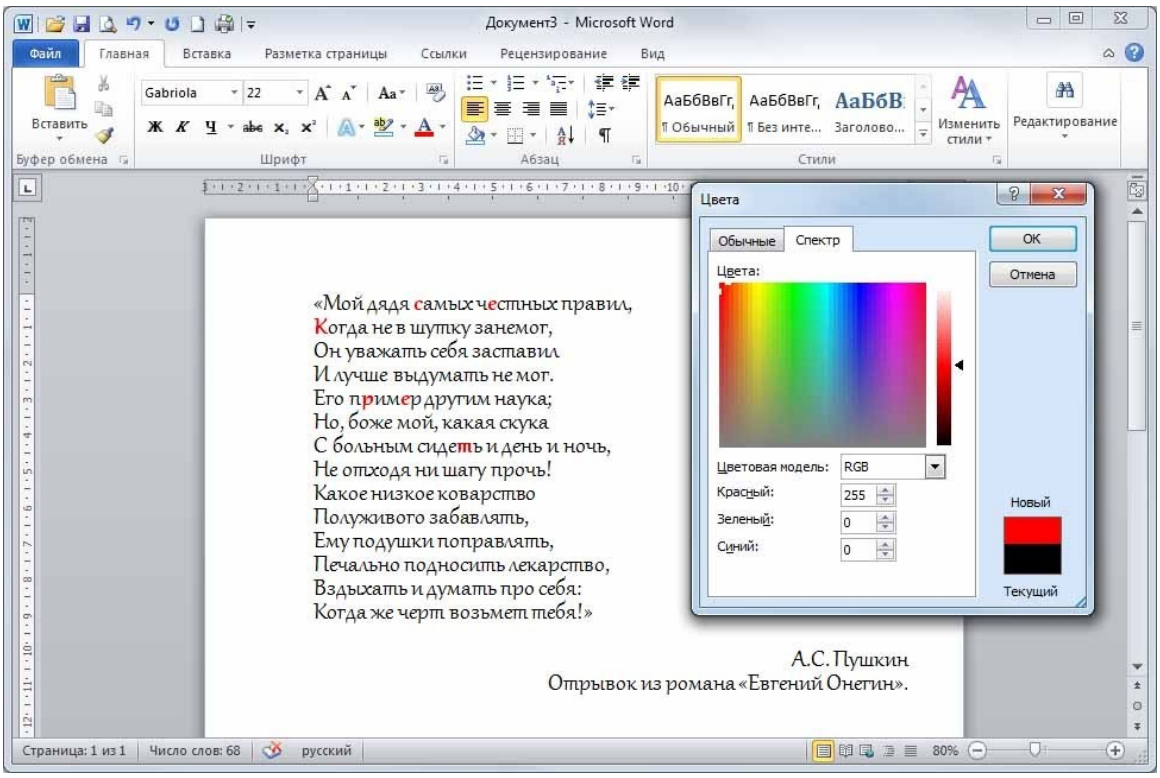


Рис.2. Принцип форматирования символов секретного сообщения «секрет» (цвет символов красный – RGB(255, 0, 0))

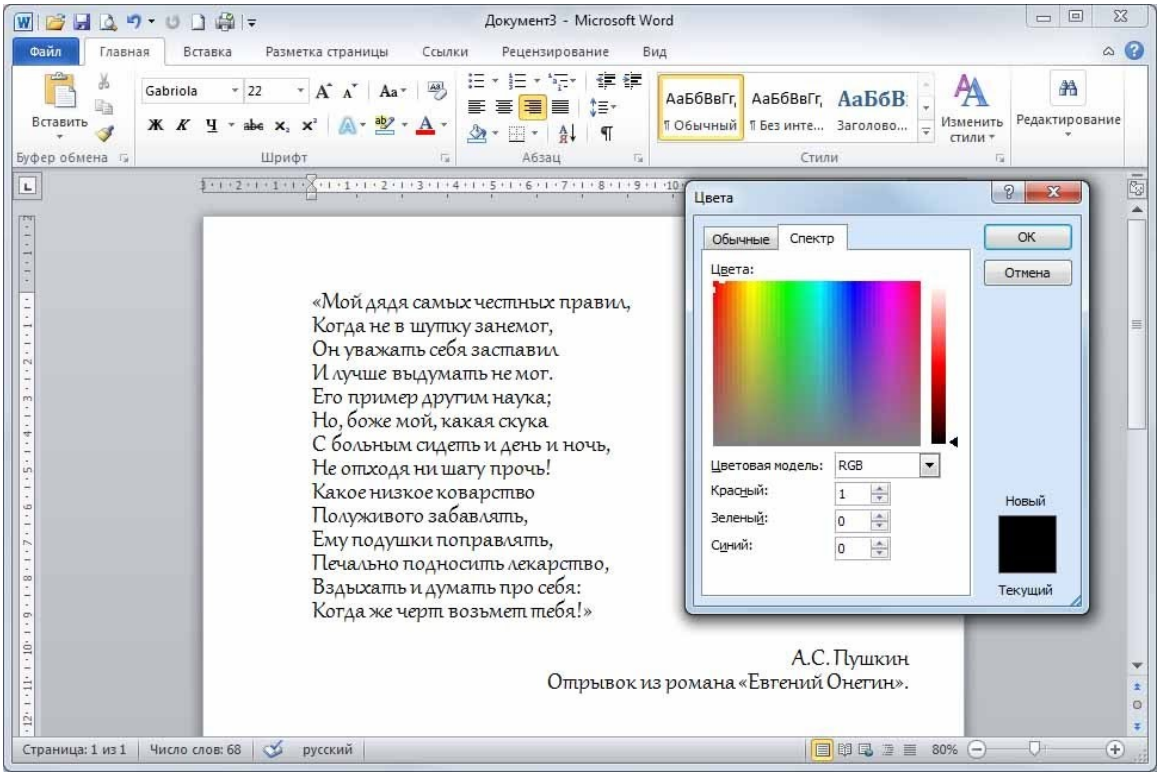


Рис.3. Стеганографическое сокрытие символов секретного сообщения «секрет» (цвет символов «почти черный» – RGB(1, 0, 0))

На рис.3 цвет символов секретного сообщения RGB(1, 0, 0) практически не отличается от цвета символов остального текста RGB(0, 0, 0).

3. Семаграммы и кодирование. Предыдущий метод можно усилить за счет использования предварительного кодирования символов секретного сообщения. Перед форматированием символы секретного сообщения вначале кодируются битовыми строками длиной n согласно принятой кодировке. В исходном тексте выбираются n первых символов, которые будут соответствовать битовому представлению первого символа секретного сообщения. Для нулей битовой строки оставляют исходное форматирование, для единиц – незначительно меняют (см. рис. 3). Процедуру последовательно повторяют для оставшихся символов секретного сообщения. Например, слово «секрет» согласно кодировке Windows 1251 в битовом представлении будет выглядеть 11110001 11100101 11101010 11110000 11100101 11110010_2.

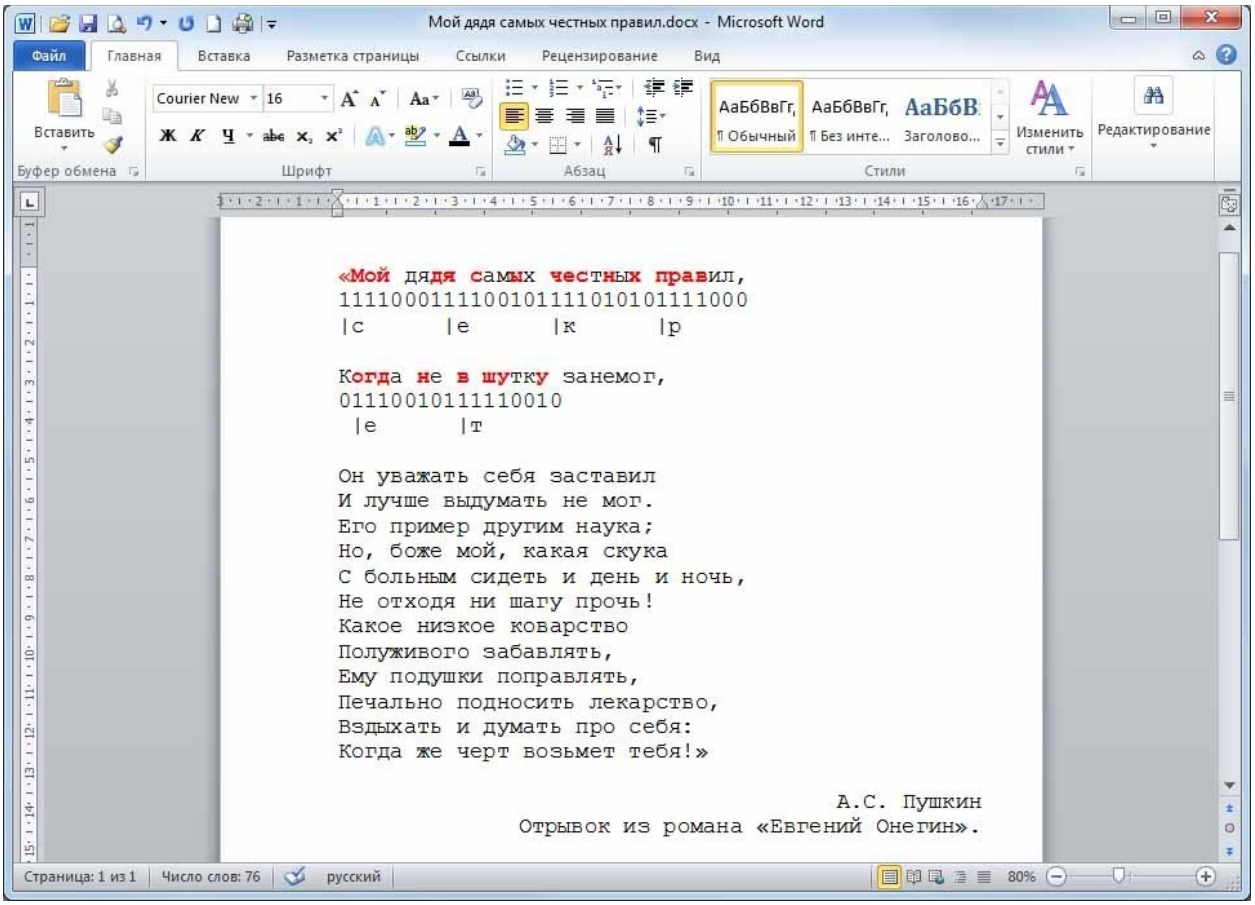


Рис.4. Принцип кодирования и форматирования символов секретного сообщения «секрет»
(цвет нулей черный – RGB(0, 0, 0); цвет единиц красный – RGB(255, 0, 0))

Стеганография

Как отмечалось ранее, разработкой средств и методов сокрытия факта передачи сообщения занимается **стеганография** (греч. στεγανός - скрытый и γράφω - пишу; буквально «тайнопись»). Наиболее эффективно ее применение совместно с криптографическими методами. Обычно стеганографию делят на два направления: классическую и компьютерную.

Компьютерная стеганография

Развитие компьютерной технологии и средств коммуникации придали новый импульс развитию и совершенствованию стеганографии. Сегодня каждый может воспользоваться теми преимуществами, которые дает стеганография как в области скрытой передачи информации, что особенно полезно в странах, где существует запрет на стойкие средства криптографии, так и в области защиты авторских прав. В настоящее время методы компьютерной стеганографии активно используются для решения следующих задач.

1. Защита конфиденциальной информации от несанкционированного доступа. Это область использования компьютерной стеганографии является наиболее эффективной при решении проблем защиты конфиденциальной информации. Так, например, объем секретного сообщения в звуковых и графических файлах может составлять до 25 - 30 % от размера файла. Причем аудиовизуальные изменения таковы, что не обнаруживаются при прослушивании и просмотре файлов большинством людей, даже если факт сокрытия известен.

2. Преодоление систем мониторинга и управления сетевыми ресурсами. Стеганографические методы позволяют противостоять попыткам контроля над информационным пространством при прохождении информации через серверы управления локальных и глобальных вычислительных сетей.

3. Камуфлирование программного обеспечения. Применяется в тех случаях, когда использование ПО незарегистрированными пользователями является нежелательным. ПО может быть закамуфлировано под стандартные универсальные программные продукты (например, текстовые редакторы) или скрыто в файлах мультимедиа и использоваться только лицами, имеющими на это права.

4. Защита авторских прав. Одним из наиболее перспективных направлений компьютерной стеганографии является технология использования **цифровых водяных знаков** ЦВЗ (digital watermarking) – в данном случае, создание невидимых глазу знаков защиты авторских прав на графические и аудио файлы. Такие ЦВЗ, помещенные в файл, могут быть распознаны специальными программами, которые извлекут из файла много полезной информации: когда создан файл, кто владеет авторскими правами, как вступить в контакт с автором и т.д. При том повальном воровстве, которое происходит в Интернете, польза от такой технологии очевидна.

Сегодня на рынке существует довольно много фирм, предлагающих продукты для создания и детектирования водяных знаков. Один из лидеров - фирма Digimarc. Ее продуктами, если верить предоставленной самой фирмой информации, пользуются более миллиона официальных клиентов: дизайнеры, художники, онлайновые галереи, журнал Playboy. Специальные поисковые агенты сканируют ресурсы Интернет, просматривая картинки на наличие ЦВЗ, и сообщают владельцам о фактах использования их собственности.

Несмотря на все заверения создателей соответствующих продуктов, ЦВЗ оказались нестойкими. Они могут перенести многое - изменение яркости и контраста, использование спецэффектов, даже печать и последующее сканирование, но они не могут перенести воздействие специальных программ-стирателей, которые появились в Интернете.

Наиболее известные методы компьютерной стеганографии и их характеристика приведены в следующей таблице.

Таблица 1. Методы компьютерной стеганографии и их характеристика

Стеганографические методы	Краткая характеристика методов	Примечания
1. Методы, основанные на использовании специальных свойств носителей данных		
1.1. Скрытие информации в неиспользуемых местах дисков	1. Используются дорожки, доступные для чтения, но не воспринимаемые ОС (например, в резервную область жесткого диска). 2. Запись в неиспользуемые места оптических дисков (CD, DVD, Blue-ray и т.д.)	1. Низкая степень скрытности. 2. Возможна передача больших объемов информации.
1.2. Нанесение дополнительных дорожек на гибкие магнитные диски (вышли из употребления)	Т.к. ширина дорожки в несколько раз меньше расстояния между дорожками (для гибких магнитных дисков), то на диск можно нанести дополнительные дорожки и записать туда информацию, не доступную ОС.	Возможна передача больших объемов информации.
1.3. Специальное форматирование дисков	Форматирование диска под размер секторов отличный от принятого в ОС.	1. Наличие программ как форматирующих подобным образом диски, так и читающих любое форматирование. 2. Возможна передача больших объемов информации.
2. Методы, основанные на использовании специальных свойств форматов данных		
2.1. Методы использования полей данных, зарезервированных для расширения	Поля расширения имеются во многих мультимедийных форматах. Они заполняются нулевой информацией и не учитываются программой.	1. Низкая степень скрытности. 2. Передача небольших объемов информации.
2.2. Методы специального форматирования в текстовых документах	1. Использование смещения символов, слов, предложений или абзацев в тексте (можно обеспечить вставкой	1. Слабая производительность методов. 2. Передача небольших

	дополнительных пробелов). 2. Выбор определенных позиций символов (например, акростих). 3. Использование дополнительных возможностей форматирования текстов (например, использование в MS Word: скрытого текста; специальных шрифтов; символов определенного шрифта, размера или цвета; белого цвета для символов и фона; одного пробела между словами для кодирования "0" и двух – для кодирования "1" и т.д.).	объемов информации. 3. Низкая степень скрытности.
2.3. Методы специального форматирования текстов при печати	1. Печать специальными шрифтами, символами определенного шрифта, размера или цвета. 2. Внесение малозаметных искажений информации при печати (Был использован при печати контрактов с клиентами в одной из московских компаний. Эта тайнопись выглядела как обычные незначительные дефекты печати и обеспечивала определенную степень подтверждения подлинности документа).	1. Слабая производительность методов. 2. Передача небольших объемов информации.
2.4. Скрытие информации в свободных областях диска	1. Использование свободной части последнего кластера файла. 2. Использование свободных кластеров без записи в таблицы размещения файлов информации о том, что в этих кластерах содержится информация.	1. Низкая степень скрытности. 2. Возможна передача больших объемов информации.

2.5. Использование особенностей файловой системы	1. Использование скрытых файлов. 2. Использование потоков в NTFS.	1. Низкая степень скрытности. 2. Возможна передача больших объемов информации.
3. Методы, основанные на использовании избыточности аудио- и видеоинформации		
3.1. Методы использования избыточности мультимедийных форматов	Младшие разряды байт, несущие информацию об интенсивности света и звука содержат очень мало полезной информации. Их заполнение практически не влияет на качество восприятия.	1. За счет введения дополнительной информации искажаются статистические характеристики цифровых потоков. 2. Для снижения компрометирующих признаков требуется коррекция статистических характеристик. 3. Возможна передача больших объемов информации.

Использование потоков в NTFS.

Любой файл в NTFS может содержать несколько потоков ("файлов"). Каждый файл NTFS содержит стандартный (default) или безымянный (unnamed) поток данных (data stream). Именно этот поток видит перед собой пользователь, открывающий файл в текстовом редакторе. И именно размер этого потока отображается в качестве размера файла. Альтернативный поток данных (alternate data stream) – файл, встраиваемый в другой. Ему может даваться любое имя и его размер не влияет на размер файла.

В частности, информация о файле с вкладки "Сводка" окна "Свойства" храниться в альтернативном потоке "♣SummaryInformation"

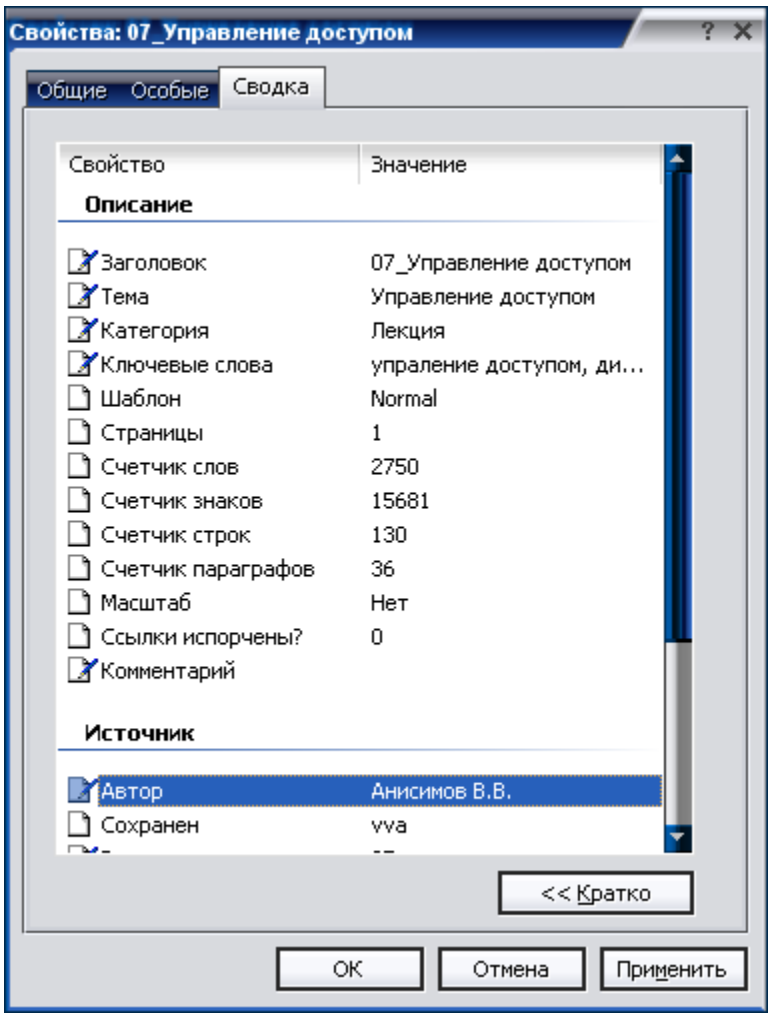


Рис 5. Свойства файла

Работа с потоками командами DOS.

Копирование файла "СовСек.txt" в альтернативный поток "ss" файла "НеСек.txt".

type СовСек.txt > НеСек.txt:ss

После копирования размер файла "НеСек.txt" не меняется.

Шутка. Таким образом, в файл размером 0 байт, можно записать несколько Гбайт мусора в потоки, заняв почти все место на диске, но файл по-прежнему будет иметь размер 0 байт.

Восстановление текстового файла из потока.

more < НеСек.txt:ss > СовСек.txt

Для выявления файлов, обладающих альтернативными потоками можно воспользоваться утилитой Streams (http://technet.microsoft.com/ru-ru/sysinternals, с исходным текстом утилиты на C).

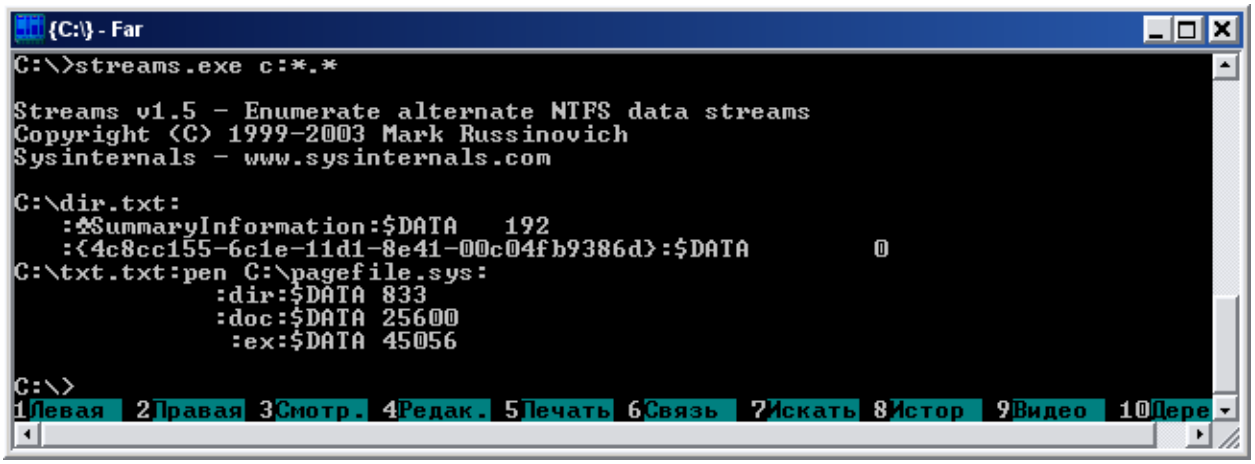


Рис 6. Отображение потоков утилитой Streams

Кроме легкого обнаружения, другим существенным недостатком является возможность использования данного способа только на дисках с NTFS. При копировании файлов на диски с другой файловой системой альтернативные потоки теряются.

Использование избыточности аудио- и видеофайлов.

Из всех приведенных в табл. 1 методов, этот является наиболее перспективным. Существуют различные его модификации, самый простой из которых LSB (Least Significant Bit, наименьший значащий бит). Суть этого метода заключается в замене последних значащих битов в контейнере (изображения, аудио или видеозаписи) на биты скрываемого сообщения. Допустим, имеется 8-битное изображение в градациях серого (0 (00000000_2) обозначает черный цвет, 255 (11111111_2) – белый). Всего имеется 256 градаций. Также предположим, что сообщение состоит из 1 байта – например, (01101011_2). При использовании 2 младших бит в описании пикселей, нам потребуется 4 пикселя. Допустим, они черного цвета. Тогда пиксели, содержащие скрытое сообщение, будут выглядеть следующим образом: (00000001 00000010 00000010 00000011_2). Тогда цвет пикселей изменится: первого - на 1/256, второго и третьего - на 2/256 и четвертого - на 3/256. Такие искажения исходного изображения, как правило, незаметны для человеческого зрения. Для разноцветных изображений искажения еще менее заметны, тем более, что в них биты исходного изображения могут совпадать с битами секретного сообщения.

Одной из лучших программ в своем классе является SecretBMP (http://www.bordak.fatal.ru/secretbmp/). В примере при скрытии файла give-me-too.zip (570 404 байта) в файле etr500.bmp (1 229 852 байта) размер результирующего файла 5.bmp стал 1 229 850 байта, а качество рисунка осталось неизменным для глаза.

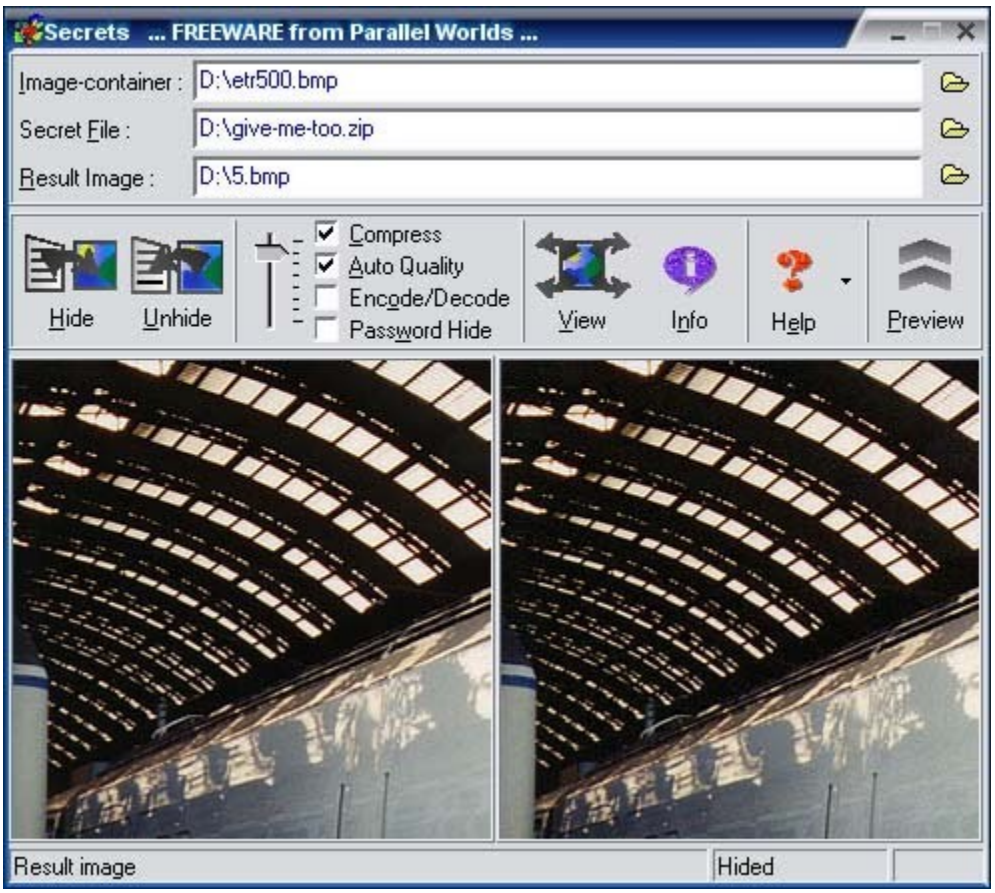


Рис 7. Окно программы SecretBMP
(слева – исходный файл-контейнер, справа – файл со вставленной секретной информацией)

Вопросы для самоконтроля

1. Электронная цифровая подпись.
2. Виды электронных цифровых подписей.
3. Применение разных видов подписей.

Практическое занятие №12. Технические условия и нормативы информационной безопасности

Цель: изучить технические условия и нормативы информационной безопасности.

Ход работы:

Создатель кибернетики Норберт Винер полагал, что информация обладает уникальными характеристиками и ее нельзя отнести ни к энергии, ни к материи. Особый статус информации как явления породил множество определений.

В словаре стандарта ISO/IEC 2382:2015 «Информационные технологии» приводится такая трактовка:

Информация (в области обработки информации) – любые данные, представленные в электронной форме, написанные на бумаге, высказанные на совещании или находящиеся на любом другом носителе, используемые финансовым учреждением для принятия решений, перемещения денежных средств, установления ставок, предоставления ссуд, обработки операций и т.п., включая компоненты программного обеспечения системы обработки.

Для разработки концепции обеспечения информационной безопасности (ИБ) под информацией понимают сведения, которые доступны для сбора, хранения, обработки (редактирования, преобразования), использования и передачи различными способами, в том числе в компьютерных сетях и других информационных системах.

Такие сведения обладают высокой ценностью и могут стать объектами посягательств со стороны третьих лиц. Стремление оградить информацию от угроз лежит в основе создания систем информационной безопасности.

Правовая основа

В декабре 2017 года в России принята новая редакция Доктрины информационной безопасности. В документ ИБ определена как состояние защищенности национальных интересов в информационной сфере. Под национальными интересами в данном случае понимается совокупность интересов общества, личности и государства, каждая группа интересов необходима для стабильного функционирования социума.

Доктрина – концептуальный документ. Правоотношения, связанные с обеспечением информационной безопасности, регулируются федеральными законами «О государственной тайне», «Об информации», «О защите персональных данных» и другими. На базе основополагающих нормативных актов разрабатываются постановления правительства и ведомственные нормативные акты, посвященные частным вопросам защиты информации.

Определение информационной безопасности

Прежде чем разрабатывать стратегию информационной безопасности, необходимо принять базовое определение самого понятия, которое позволит применять определенный набор способов и методов защиты.

Практики отрасли предлагают понимать под информационной безопасностью стабильное состояние защищенности информации, ее носителей и инфраструктуры, которая обеспечивает целостность и устойчивость процессов,

связанных с информацией, к намеренным или непреднамеренным воздействиям естественного и искусственного характера. Воздействия классифицируются в виде угроз ИБ, которые могут нанести ущерб субъектам информационных отношений.

Таким образом, под защитой информации будет пониматься комплекс правовых, административных, организационных и технических мер, направленных на предотвращение реальных или предполагаемых ИБ-угроз, а также на устранение последствий инцидентов. Непрерывность процесса защиты информации должна гарантировать борьбу с угрозами на всех этапах информационного цикла: в процессе сбора, хранения, обработки, использования и передачи информации.

Информационная безопасность в этом понимании становится одной из характеристик работоспособности системы. В каждый момент времени система должна обладать измеряемым уровнем защищенности, и обеспечение безопасности системы должно быть непрерывным процессом, которые осуществляется на всех временных отрезках в период жизни системы.

В теории информационной безопасности под субъектами ИБ понимают владельцев и пользователей информации, причем пользователей не только на постоянной основе (сотрудники), но и пользователей, которые обращаются к базам данных в единичных случаях, например, государственные органы, запрашивающие информацию. В ряде случаев, например, в банковских ИБ-стандартах к владельцам информации причисляют акционеров – юридических лиц, которым принадлежат определенные данные.

Поддерживающая инфраструктура, с точки зрения основ ИБ, включает компьютеры, сети, телекоммуникационное оборудование, помещения, системы жизнеобеспечения, персонал. При анализе безопасности необходимо изучить все элементы систем, особое внимание уделив персоналу как носителю большинства внутренних угроз.

Для управления информационной безопасностью и оценки ущерба используют характеристику приемлемости, таким образом, ущерб определяется как приемлемый или неприемлемый. Каждой компании полезно утвердить собственные критерии допустимости ущерба в денежной форме или, например, в виде допустимого вреда репутации. В государственных учреждениях могут быть приняты другие характеристики, например, влияние на процесс управления или отражение степени ущерба для жизни и здоровья граждан. Критерии существенности, важности и ценности информации могут меняться в ходе

жизненного цикла информационного массива, поэтому должны своевременно пересматриваться.

Информационной угрозой в узком смысле признается объективная возможность воздействовать на объект защиты, которое может привести к утечке, хищению, разглашению или распространению информации. В более широком понимании к ИБ-угрозам будут относиться направленные воздействия информационного характера, цель которых – нанести ущерба государству, организации, личности. К таким угрозам относится, например, диффамация, намеренное введение в заблуждение, некорректная реклама.

Три основных вопроса ИБ-концепции для любой организации

- Что защищать?
- Какие виды угроз превалируют: внешние или внутренние?
- Как защищать, какими методами и средствами?

Система ИБ

Система информационной безопасности для компании – юридического лица включает три группы основных понятий: целостность, доступность и конфиденциальность. Под каждым скрываются концепции с множеством характеристик.

Под **целостностью** понимается устойчивость баз данных, иных информационных массивов к случайному или намеренному разрушению, внесению несанкционированных изменений. Понятие целостности может рассматриваться как:

- статическое, выражающееся в неизменности, аутентичности информационных объектов тем объектам, которые создавались по конкретному техническому заданию и содержат объемы информации, необходимые пользователям для основной деятельности, в нужной комплектации и последовательности;
- динамическое, подразумевающее корректное выполнение сложных действий или транзакций, не причиняющее вреда сохранности информации.

Для контроля динамической целостности используют специальные технические средства, которые анализируют поток информации, например, финансовые, и выявляют случаи кражи, дублирования, перенаправления, изменения порядка сообщений. Целостность в качестве основной характеристики требуется тогда, когда на основе поступающей или имеющейся информации принимаются решения о совершении действий. Нарушение порядка расположения команд или последовательности действий может нанести большой ущерб в случае описания технологических процессов, программных кодов и в других аналогичных ситуациях.

Доступность – это свойство, которое позволяет осуществлять доступ авторизированных субъектов к данным, представляющим для них интерес, или обмениваться этими данными. Ключевое требование легитимации или авторизации субъектов дает возможность создавать разные уровни доступа. Отказ системы предоставлять информацию становится проблемой для любой организации или групп пользователей. В качестве примера можно привести

недоступность сайтов госуслуг в случае системного сбоя, что лишает множество пользователей возможности получить необходимые услуги или сведения.

Конфиденциальность означает свойство информации быть доступной тем пользователям: субъектам и процессам, которым допуск разрешен изначально. Большинство компаний и организаций воспринимают конфиденциальность как ключевой элемент ИБ, однако на практике реализовать ее в полной мере трудно. Не все данные о существующих каналах утечки сведений доступны авторам концепций ИБ, и многие технические средства защиты, в том числе криптографические, нельзя приобрести свободно, в ряде случаев оборот ограничен.

Равные свойства ИБ имеют разную ценность для пользователей, отсюда – две крайние категории при разработке концепций защиты данных. Для компаний или организаций, связанных с государственной тайной, ключевым параметром станет конфиденциальность, для публичных сервисов или образовательных учреждений наиболее важный параметр – доступность.

Объекты защиты в концепциях ИБ

Различие в субъектах порождает различия в объектах защиты. Основные группы объектов защиты:

- информационные ресурсы всех видов (под ресурсом понимается материальный объект: жесткий диск, иной носитель, документ с данными и реквизитами, которые помогают его идентифицировать и отнести к определенной группе субъектов);
- права граждан, организаций и государства на доступ к информации, возможность получить ее в рамках закона; доступ может быть ограничен только нормативно-правовыми актами, недопустима организация любых барьеров, нарушающих права человека;
- система создания, использования и распространения данных (системы и технологии, архивы, библиотеки, нормативные документы);
- система формирования общественного сознания (СМИ, интернет-ресурсы, социальные институты, образовательные учреждения).

Каждый объект предполагает особую систему мер защиты от угроз ИБ и общественному порядку. Обеспечение информационной безопасности в каждом случае должно базироваться на системном подходе, учитывающем специфику объекта.

Категории и носители информации

Российская правовая система, правоприменительная практика и сложившиеся общественные отношения классифицируют информацию по критериям доступности. Это позволяет уточнить существенные параметры, необходимые для обеспечения информационной безопасности:

- информация, доступ к которой ограничен на основании требований законов (государственная тайна, коммерческая тайна, персональные данные);
- сведения в открытом доступе;
- общедоступная информация, которая предоставляется на определенных условиях: платная информация или данные, для пользования которыми требуется оформить допуск, например, библиотечный билет;

- опасная, вредная, ложная и иные типы информации, оборот и распространение которой ограничены или требованиями законов, или корпоративными стандартами.

Информация из первой группы имеет два режима охраны. **Государственная тайна**, согласно закону, это защищаемые государством сведения, свободное распространение которых может нанести ущерб безопасности страны. Это данные в области военной, внешнеполитической, разведывательной, контрразведывательной и экономической деятельности государства. Владелец этой группы данных – непосредственно государство. Органы, уполномоченные принимать меры по защите государственной тайны, – Министерство обороны, Федеральная служба безопасности (ФСБ), Служба внешней разведки, Федеральной службы по техническому и экспортному контролю (ФСТЭК).

Конфиденциальная информация – более многоплановый объект регулирования. Перечень сведений, которые могут составлять конфиденциальную информацию, содержится в указе президента №188 «Об утверждении перечня сведений конфиденциального характера». Это персональные данные; тайна следствия и судопроизводства; служебная тайна; профессиональная тайна (врачебная, нотариальная, адвокатская); коммерческая тайна; сведения об изобретениях и о полезных моделях; сведения, содержащиеся в личных делах осужденных, а также сведения о принудительном исполнении судебных актов.

Персональные данные существует в открытом и в конфиденциальном режиме. Открытая и доступная всем пользователям часть персональных данных включает имя, фамилию, отчество. Согласно ФЗ-152 «О персональных данных», субъекты персональных данных имеют право:

- на информационное самоопределение;
- на доступ к личным персональным данным и внесение в них изменений;
- на блокирование персональных данных и доступа к ним;
- на обжалование неправомерных действий третьих лиц, совершенных в отношении персональных данных;
- на возмещение причиненного ущерба.

Право на обработку персональных данных закреплено в положениях о государственных органах, федеральными законами, лицензиями на работу с персональными данными, которые выдает Роскомнадзор или ФСТЭК. Компании, которые профессионально работают с персональными данными широкого круга лиц, например, операторы связи, должны войти в реестр, его ведет Роскомнадзор.

Отдельным объектом в теории и практике ИБ выступают носители информации, доступ к которым бывает открытым и закрытым. При разработке концепции ИБ способы защиты выбираются в зависимости от типа носителя. Основные носители информации:

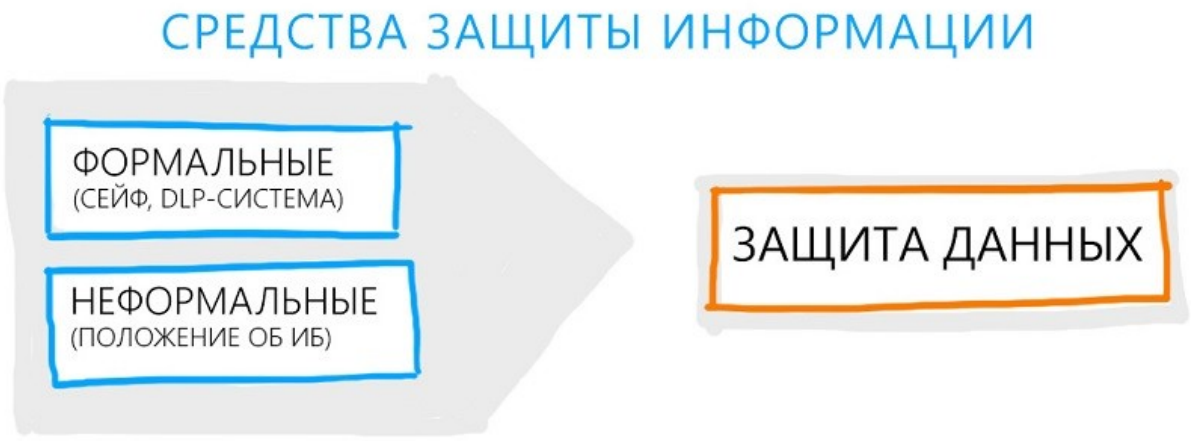
- печатные и электронные средства массовой информации, социальные сети, другие ресурсы в интернете;
- сотрудники организации, у которых есть доступ к информации на основании своих дружеских, семейных, профессиональных связей;
- средства связи, которые передают или сохраняют информацию: телефоны, АТС, другое телекоммуникационное оборудование;
- документы всех типов: личные, служебные, государственные;
- программное обеспечение как самостоятельный информационный объект, особенно если его версия дорабатывалась специально для конкретной компании;
- электронные носители информации, которые обрабатывают данные в автоматическом порядке.

Средства защиты информации

Для целей разработки концепций ИБ-защиты средства защиты информации принято делить на нормативные (неформальные) и технические (формальные).

Неформальные средства защиты – это документы, правила, мероприятия, формальные – это специальные технические средства и программное обеспечение. Разграничение помогает распределить зоны ответственности при создании ИБ-систем: при общем руководстве защитой административный персонал реализует нормативные способы, а IT-специалисты, соответственно, технические.

Основы информационной безопасности предполагают разграничение полномочий не только в части использования информации, но и в части работы с ее охраной. Подобное разграничение полномочий требует и нескольких уровней контроля.



Формальные средства защиты

Широкий диапазон технических средств ИБ-защиты включает:

Физические средства защиты. Это механические, электрические, электронные механизмы, которые функционируют независимо от информационных систем и создают препятствия для доступа к ним. Замки, в том числе электронные, экраны, жалюзи призваны создавать препятствия для контакта

дестабилизирующих факторов с системами. Группа дополняется средствами систем безопасности, например, видеокамерами, видеорегистраторами, датчиками, выявляющие движение или превышение степени электромагнитного излучения в зоне расположения технических средств снятия информации, закладных устройств.

Аппаратные средства защиты. Это электрические, электронные, оптические, лазерные и другие устройства, которые встраиваются в информационные и телекоммуникационные системы. Перед внедрением аппаратных средств в информационные системы необходимо удостовериться в совместимости.

Программные средства – это простые и системные, комплексные программы, предназначенные для решения частных и комплексных задач, связанных с обеспечением ИБ. Примером комплексных решений служат DLP-системы и SIEM-системы: первые служат для предотвращения утечки, переформатирования информации и перенаправления информационных потоков, вторые – обеспечивают защиту от инцидентов в сфере информационной безопасности. Программные средства требовательны к мощности аппаратных устройств, и при установке необходимо предусмотреть дополнительные резервы.

К **специфическим средствам** информационной безопасности относятся различные криптографические алгоритмы, позволяющие шифровать информацию на диске и перенаправляемую по внешним каналам связи. Преобразование информации может происходить при помощи программных и аппаратных методов, работающих в корпоративных информационных системах.

Все средства, гарантирующие безопасность информации, должны использоваться в совокупности, после предварительной оценки ценности информации и сравнения ее со стоимостью ресурсов, затраченных на охрану. Поэтому предложения по использованию средств должны формулироваться уже на этапе разработки систем, а утверждение должно производиться на том уровне управления, который отвечает за утверждение бюджетов.

В целях обеспечения безопасности необходимо проводить мониторинг всех современных разработок, программных и аппаратных средств защиты, угроз и своевременно вносить изменения в собственные системы защиты от несанкционированного доступа. Только адекватность и оперативность реакции на угрозы поможет добиться высокого уровня конфиденциальности в работе компании.

Неформальные средства защиты

Неформальные средства защиты группируются на нормативные, административные и морально-этические. На первом уровне защиты находятся нормативные средства, регламентирующие информационную безопасность в качестве процесса в деятельности организации.

- **Нормативные средства**

Эта категория средств обеспечения информационной безопасности представлена законодательными актами и нормативно-распорядительными документами, которые действуют на уровне организации.

В мировой практике при разработке нормативных средств ориентируются на стандарты защиты ИБ, основный – ISO/IEC 27000. Стандарт создавали две организации:

- ISO – Международная комиссия по стандартизации, которая разрабатывает и утверждает большинство признанных на международном уровне методик сертификации качества процессов производства и управления;
- IEC – Международная энергетическая комиссия, которая внесла в стандарт свое понимание систем ИБ, средств и методов ее обеспечения

Актуальная версия ISO/IEC 27000-2016 предлагают готовые стандарты и опробованные методики, необходимые для внедрения ИБ. По мнению авторов методик, основа информационной безопасности заключается в системности и последовательной реализации всех этапов от разработки до пост-контроля.

Для получения сертификата, который подтверждает соответствие стандартам по обеспечению информационной безопасности, необходимо внедрить все рекомендуемые методики в полном объеме. Если нет необходимости получать сертификат, в качестве базы для разработки собственных ИБ-систем допускается принять любую из более ранних версий стандарта, начиная с ISO/IEC 27000-2002, или российских ГОСТов, имеющих рекомендательный характер.

По итогам изучения стандарта разрабатываются два документа, которые касаются безопасности информации. Основной, но менее формальный – концепция ИБ предприятия, которая определяет меры и способы внедрения ИБ-системы для информационных систем организации. Второй документ, которые обязаны исполнять все сотрудники компании, – положение об информационной безопасности, утверждаемое на уровне совета директоров или исполнительного органа.

Кроме положения на уровне компании должны быть разработаны перечни сведений, составляющих коммерческую тайну, приложения к трудовым договорам, закрепляющий ответственность за разглашение конфиденциальных данных, иные стандарты и методики. Внутренние нормы и правила должны содержать механизмы реализации и меры ответственности. Чаще всего меры носят дисциплинарный характер, и нарушитель должен быть готов к тому, что за нарушением режима коммерческой тайны последуют существенные санкции вплоть до увольнения.

- **Организационные и административные меры**

В рамках административной деятельности по защите ИБ для сотрудников служб безопасности открывается простор для творчества. Это и архитектурно-планировочные решения, позволяющие защитить переговорные комнаты и кабинеты руководства от прослушивания, и установление различных уровней доступа к информации. Важными организационными мерами станут сертификация деятельности компании по стандартам ISO/IEC 27000, сертификация отдельных аппаратно-программных комплексов, аттестация субъектов и объектов на соответствие необходимым требованиям безопасности, получений лицензий, необходимых для работы с защищенными массивами информации.

С точки зрения регламентации деятельности персонала важным станет оформление системы запросов на допуск к интернету, внешней электронной почте, другим ресурсам. Отдельным элементом станет получение электронной цифровой подписи для усиления безопасности финансовой и другой информации, которую передают государственным органам по каналам электронной почты.

- **Морально-этические меры**

Морально-этические меры определяют личное отношение человека к конфиденциальной информации или информации, ограниченной в обороте. Повышение уровня знаний сотрудников касательно влияния угроз на деятельность компании влияет на степень сознательности и ответственности сотрудников. Чтобы бороться с нарушениями режима информации, включая, например, передачу паролей, неосторожное обращение с носителями, распространение конфиденциальных данных в частных разговорах, требуется делать упор на личную сознательность сотрудника. Полезным будет установить показатели эффективности персонала, которые будут зависеть от отношения к корпоративной системе ИБ.

Вопросы для самоконтроля

1. Регламенты информационной безопасности.
2. Процедуры информационной безопасности.

Рекомендуемая литература

Основная литература:

1. Горкуш, С. В. Защита конфиденциальной информации. Практикум: учебное пособие / С. В. Горкуш, О. Г. Савка. - Москва: РТУ МИРЭА, 2022. - 87 с. – Текст: электронный // Лань: электронно-библиотечная система. - URL: https://e.lanbook.com/book/311156

2. Игнатьев, Е. Б. Защита информации: криптоалгоритмы хеширования / Е. Б. Игнатьев. - 2-е изд., испр. - Санкт-Петербург: Лань, 2024. - 264 с. - ISBN 978-5-507-47433-2. - Текст: электронный // Лань: электронно-библиотечная система. - URL: https://e.lanbook.com/book/370928

3. Мартыненков, Б. В. Основы криптографической защиты информации: учебно-методическое пособие / Б. В. Мартыненков, В. А. Иванов, М. Ю. Конышев. - Москва: РТУ МИРЭА, 2023. - 95 с. - ISBN 978-5-7339-1807-5. - Текст: электронный // Лань: электронно-библиотечная система. - URL: https://e.lanbook.com/book/368762

Дополнительная литература:

1. Гончаренко, Ю. Ю. Теория информации для защиты информационных процессов : учебное пособие / Ю. Ю. Гончаренко. - Севастополь: СевГУ, 2023. - 147 с. - Текст: электронный // Лань: электронно-библиотечная система. - URL: https://e.lanbook.com/book/332207

2. Овчинников, А. А. Криптографические методы защиты информации: учебное пособие / А. А. Овчинников. - Санкт-Петербург: ГУАП, 2021. - 133 с. - ISBN 978-5-8088-1591-9. - Текст: электронный // Лань: электронно-библиотечная система. - URL: https://e.lanbook.com/book/216491

3. Прохорова, О. В. Информационная безопасность и защита информации: учебник для вузов / О. В. Прохорова. - 3-е изд., стер. - Санкт-Петербург: Лань, 2021. - 124 с. - ISBN 978-5-8114-7970-2. - Текст: электронный // Лань: электронно-библиотечная система. - URL: https://e.lanbook.com/book/169817