МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

# МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к практическим занятиям

по дисциплине «**ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ В ИНФОКОММУНИКАЦИЯХ**»

для студентов направления подготовки

11.04.02 Инфокоммуникационные технологии и системы связиНаправленность (профиль)

«Технологии построения программно-управляемых систем и компьютерных сетей»

Ставрополь, 2026

Методические рекомендации содержат теоретическое обоснование, задания, методику и порядок выполнения работ, контрольные вопросы, указания по технике безопасности, а также список рекомендуемой литературы.

## Содержание

# Введение

В настоящее время во всем мире резко повысилось внимание к проблеме информационной безопасности. Это обусловлено процессами стремительного расширения потоков информации, пронизывающих все сферы жизни общества.

Информация давно перестала быть просто необходимым для производства вспомогательным ресурсом или побочным проявлением всякого рода деятельности. Она приобрела ощутимый стоимостной вес, который четко определяется реальной прибылью, получаемой при ее использовании, или размерами ущерба, с разной степенью вероятности наносимого владельцу информации. Однако создание индустрии переработки информации порождает целый ряд сложных проблем. Одной из таких проблем является надежное обеспечение сохранности и установленного статуса информации, циркулирующей и обрабатываемой в инфокоммуникационных системах и сетях.

В данных методических указаниях разработаны лабораторные работы, направленные на исследование принципов построения и параметров основных узлов, обеспечивающих защиту информации.

# 1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ
## 1.1. Проблемы защиты информации

В настоящее время во всем мире резко повысилось внимание к проблеме информационной безопасности. Это обусловлено процессами стремительного расширения потоков информации, пронизывающих все сферы жизни общества.

Информация давно перестала быть просто необходимым для производства вспомогательным ресурсом или побочным проявлением всякого рода деятельности. Она приобрела ощутимый стоимостной вес, который четко определяется реальной прибылью, получаемой при ее использовании, или размерами ущерба, с разной степенью вероятности наносимого владельцу информации. Однако создание индустрии переработки информации порождает целый ряд сложных проблем. Одной из таких проблем является надежное обеспечение сохранности и установленного статуса информации, циркулирующей и обрабатываемой в информационно-вычислительных системах и сетях.

Появление глобальных компьютерных сетей сделало простым получение доступа к информации, как для отдельных пользователей, так и для больших организаций. Но легкость и высокая скорость доступа к данным при помощи компьютерных сетей, таких как *Internet,* также сделали значительными следующие угрозы безопасности данных при отсутствии мер их защиты:

- неавторизованный доступ к информации;
- неавторизованное изменение информации
- неавторизованный доступ к сетям и сервисам;
- другие сетевые атаки, например, повтор перехваченных ранее транзакций и атаки типа "отказ в обслуживании".

При обработке любой значимой информации при помощи отдельного компьютера, а тем более в сети, возникает вопрос о ее защите от

несанкционированного доступа и использования. Наиболее распространенный в компьютерных системах способ защиты – использование паролей – более пригоден для защиты доступа к вычислительным ресурсам, нежели для защиты информации. Это своеобразный экран, отгораживающий законных пользователей системы от посторонних, пройдя сквозь который, квалифицированный пользователь получает доступ практически ко всей информации.

В настоящее время исключительно важное значение в разных областях приобрели вопросы, связанные с сохранением и передачей конфиденциальной информации. Возникающие при этом задачи решает *криптография* – наука о методах преобразования информации в целях ее защиты от незаконных пользователей.

Ретроспективный взгляд на историю развития криптографии, как специфическую область человеческой деятельности, позволяет выделить три основных периода. Первый, наиболее продолжительный, – это эпоха "ручной криптографии". Ее начало теряется в глубокой древности, а окончание приходится на 30-е годы XX века. Криптография прошла путь от магического искусства до вполне прозаической прикладной специальности чиновников дипломатических и военных ведомств.

Второй период – создание и широкое внедрение в практику сначала механических, затем электромеханических и электронных устройств шифрования, организация целых сетей засекреченной связи. Его началом можно считать разработку Гилбертом Вернамом *(G.Vernam)* в 1917 году схемы телеграфной шифровальной машин, использующей *одноразовую гамму*, рис.1.1.

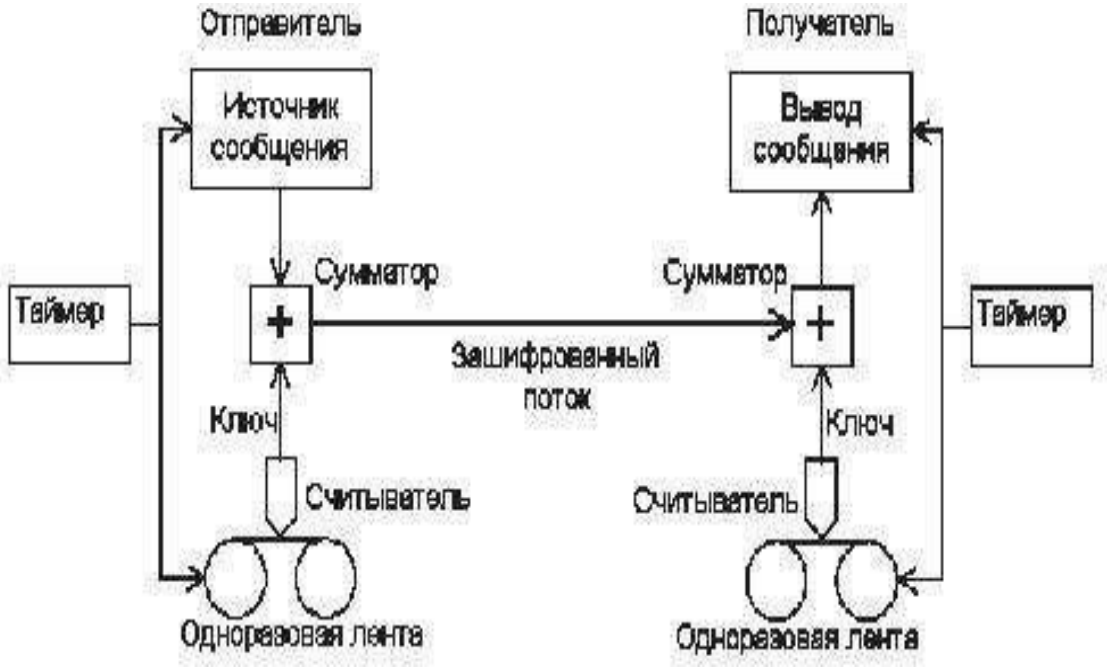



Рис. 1.1. Шифрование методом Вернама

К середине 70-х годов с развитием разветвленных коммерческих сетей связи, электронной почты и глобальных информационных систем на первый план вышли новые проблемы – проблемы снабжения ключами и проблемы подтверждения подлинности.

В 1976 году американские ученые Уитфилд Диффи *(W.Diffie)* и Мартин Хеллман *(M.Hellman)* предложили два новых принципа организации засекреченной связи без предварительного снабжения абонентов секретными ключами шифрования – принцип так называемого *открытого шифрования* и принцип *открытого распределения ключей*. Этот момент можно считать началом нового периода в развитии криптографии. В настоящее время это направление современной криптографии очень интенсивно развивается.

### 1.2. Из истории криптографии

Понятие “безопасность” охватывает широкий круг интересов, как отдельных лиц, так и целых государств. Во все исторические времена

существенное внимание уделялось проблеме информационной безопасности, обеспечению защиты конфиденциальной информации от озна-

комления, кражи, модификации, подмены. Решением этих вопросов занимается *криптография.*

Рис. 1.2. Джон Валлис

*Криптография* – тайнопись. Термин ввел Джон Валлис *(John Wallis)* (1616-1703), английский математик. Потребность шифровать и передавать шифрованные сообщения возникла очень давно. Так еще в V-VI вв до н.э. греки применяли специальное шифрующее устройство. По описанию Плутарха, оно состояло из двух цилиндрических стержней одинаковой длины и толщины. Один оставляли себе, а другой отдавали отъезжающему. Эти стержни называли *сциталами.*

При необходимости передачи сообщения, длинную ленту папируса наматывали на сциталу, не оставляя на ней никакого промежутка. Затем, оставляя папирус на сцитале, писали на нем все, что необходимо, а, написав, снимали папирус и без стержня отправляли адресату. Так как буквы оказывались разбросанными в беспорядке, то прочитать сообщение мог только тот, кто имел свою сциталу такой же длины и толщины, намотав на нее папирус.

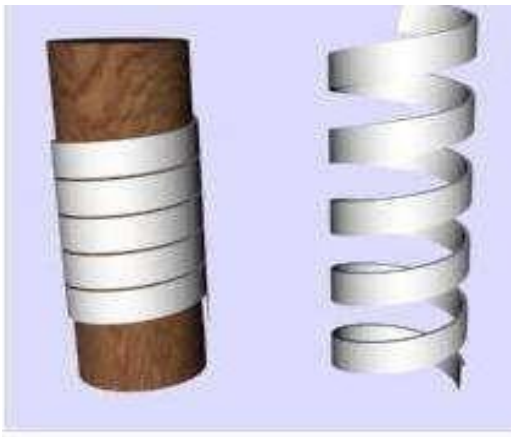


Рис 1.3. Сцитала

***Квадрат Полибия***

В Древней Греции (II в. до н.э.) был известен шифр, называемый "*квадрат Полибия*".

Это устройство представляло собой квадрат 5*5, столбцы и строки которого нумеровали цифрами от 1 до 5. В каждую клетку записывалась одна буква (в греческом варианте одна клетка оказывалась пустой, а в латинском – в одну клетку помещали две буквы *I,J*). В результате каждой букве отвечала пара чисел по номеру строки и столбца.

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

| 13 | 34 | 22 | 24 | 44 | 34 | 15 | 42 | 22 | 34 | 43 | 45 | 32 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

**Cogito ergo sum -** лат. "Я мыслю, следовательно, существую"
Р.Декарт

***Код Цезаря***

В I в.н.э. Ю.Цезарь во время войны с галлами, переписываясь со своими друзьями в Риме, заменял в сообщении первую букву латинского алфавита *(A)* на четвертую *(D)*, вторую *(B)* – на пятую *(E)*, наконец последнюю – на третью.

[1] Полибий (200-120 гг. до н.э.) древнегреческий историк

| | |
|---|---|
| ABCDEFGHIJKLMNOPQRSTUVWXYZ<br>DEFGHIJKLMNOPQRSTUVWXYZABC | YHQL YLGL YLFL<br>Veni vidi vici - "Пришел, увидел победил"<br>Ю.Цезарь Донесение Сенату о победе над понтийским царем |

Шифр Цезаря относится к так называемому классу *моноалфавитных подстановок* и имеет множество модификаций.

***Решетка Кардано***

Широко известны шифры, принадлежащие к классу *перестановка,* в частности "*решетка Кардано*"[2]. Это прямоугольная карточка с отверстиями, чаще всего квадратная, которая при наложении на лист бумаги оставляет открытыми лишь некоторые его части. Число строк и столбцов на карточке четно. Карточка сделана так, что при последовательном ее поворачивании каждая клетка лежащего под ней листа окажется занятой. Карточку поворачивают сначала вдоль вертикальной оси симметрии на 180 □, а затем вдоль горизонтально оси также на 180□. И вновь повторяют ту же процедуру.

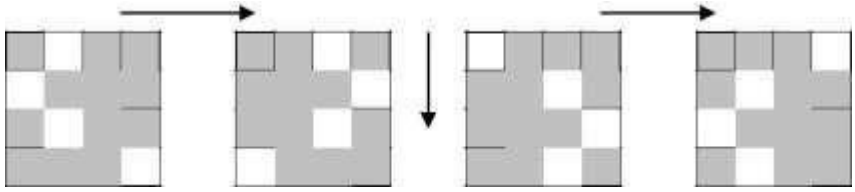


Рис. 1.4. Решетка Кардано

[2] Кардано Джероламо (1501-1576) - итальянский математик, философ и врач

***Диск Альберти***

Итальянец Альберти (XVI в.) впервые выдвинул идею "двойного шифрования" – текст, полученный в результате первого шифрования, подвергался повторному зашифрованию. В трактате Альберти был приведен его собственный шифр, который он назвал "шифром, достойным королей". Он утверждал, что этот шифр недешифруем. Реализация шифра осуществлялась с помощью шифровального диска, положившего начало целой серии *многоалфавитных подстановок*. Устройство представляло собой пару дисков – внешний, неподвижный (на нем были нанесены буквы в естественном порядке и цифры от 1 до 4) и внутренний – подвижный – на нем буквы были переставлены. Процесс шифрования заключался в нахождении буквы открытого текста на внешнем диске и замену ее на соответствующую (стоящую под ней) букву шифрованного текста. После шифрования нескольких слов внутренний диск сдвигался на один шаг. Ключом данного шифра являлся порядок расположения букв на внутреннем диске и его начальное положение относительно внешнего диска.

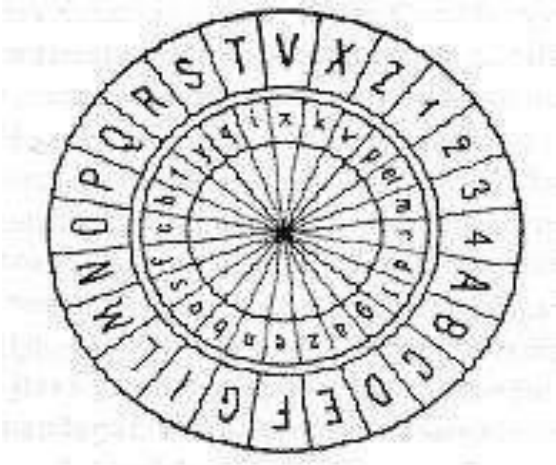



Рис. 1.5. Диск Альберт

### *Таблица Виженера*

Неудобство рассмотренных выше шрифтов моноалфавитных подстановок очевидно, так как в случае использования стандартного алфавита таблица частот встречаемости букв алфавита позволяет определить один или несколько символов, а этого иногда достаточно для вскрытия шифра (“Пляшущие человечки” Конан Дойля или “Золотой жук” Эдгара По). Поэтому использовались различные приемы для того чтобы затруднить дешифрование, например использование “*таблицы Виженера”*, которая представляет собой квадратную таблицу с числом строк и столбцов равным количеству букв алфавита. Чтобы зашифровать какое-либо сообщение выбирают слово – лозунг (например, “монастырь”) и надписывают его над сообщением с необходимым повторением.

Чтобы получить шифрованный текст, находят очередной знак лозунга, начиная с первого в вертикальном алфавите, а ему соответствующий знак сообщения в горизонтальном алфавите. На пересечении выделенных столбца и строки находим первую букву шифра. Очевидно, что ключом к такому шифру является используемый лозунг.

**м о н а с т ы р ь м о н а с т ы р ь м о н**

**р а с к и н у л о с ь м о р е ш и р о к о**

**э о я к щ а п ы й ю й щ о в ч ф ш л ь ш ы**

***Таблица Виженера***[3]

**АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮЯ**
**БВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮЯА**
**ВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮЯАБ**
**ГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮЯАБВ**
**ДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮЯФБВГ**
**ЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮЯФБВГД**
**ЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮЯФБВГДЕ**
**ЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖ**
**ИЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗ**
**ЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИ**
**КЛМНОПРСТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙ**
**ЛМНОПРСТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙК**
**МНОПРСТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛ**
**НОПРСТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМ**
**ОПРСТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМН**
**ПРСТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНО**
**РСТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОП**
**СТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОПР**
**ТУФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОПРС**
**УФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОПРСТ**
**ФХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОПРСТУ**
**ХЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОПРСТУФ**
**ЦЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОПРСТУФХ**
**ЧШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОПРСТУФХЦ**
**ШЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧ**
**ЩЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШ**
**ЬЫЭЮЯАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩ**
**ЫЭЮЯАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЬ**
**ЭЮЯАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧЩШЬЫ**
**ЮЯАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЭ**
**ЯАБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮ**

[3]Блез де Виженер (1523-1596) - французский посол в Риме, написал большой труд о шифрах. Квадратный шифр Виженера на протяжении почти 400 лет не был дешифрован, считался недешифруемым шифром

**Одноразовый шифровальный блокнот**

Примером нераскрываемого шифра может служить так называемый *одноразовый шифровальный блокнот* – шифр, в основе которого лежит та же идея, что в шифре Цезаря. Назовем *расширенным алфавитом* множество букв алфавита и знаков препинания { *. , : ; ! ? () – “ <пробел>*}, число символов расширенного кириллического алфавита в данном варианте будет равно *44*. Занумеруем символы расширенного алфавита числами от *0* до *43*. Тогда любой передаваемый текст можно рассматривать как последовательность $\{a_n\}$ чисел множества $A=\{0,1,2,\ldots,43\}$.

Предположим, что имеем случайную последовательность $\{c_n\}$ из чисел множества *A* той же длины, что и передаваемый текст – *ключ*.

Складывая по модулю *44* число из передаваемого текста $a_n$ с соответствующим числом из множества ключа $c_n$:

$a_n + c_n \equiv b_n \pmod{44},\ 0<=b_n<=43$

получим последовательность $\{b_n\}$ знаков шифрованного текста. Чтобы получить передаваемый текст, можно воспользоваться тем же ключом:

$a_n \equiv b_n - c_n \pmod{44},\ 0<=a_n<=43$

У двух абонентов, находящихся в секретной переписке, имеются два одинаковых блокнота. В каждом из них на нескольких листах напечатана случайная последовательность чисел множества *A*. Отправитель свой текст шифрует, указанным выше способом, при помощи первой страницы блокнота. Зашифровав сообщение, он уничтожает использованную страницу и отправляет текст сообщения второму абоненту, получатель шифрованного текста расшифровывает его и также уничтожает.

использованный лист блокнота. Нетрудно видеть, что одноразовый шифр не раскрываем в принципе, так как символ в тексте может быть заменен любым другим символом и этот выбор совершенно случаен.

### 1.3. Методы шифрования

#### 1.3.1.Одноалфавитный метод

Данный метод, пожалуй, самый древний из всех известных методов. В его основе лежит простой способ шифрования: отправитель и получатель зашифрованного документа заранее договариваются об определенном смещении букв относительно их обычного местоположения в алфавите. Например, для кириллицы, если смещение равно 1, то “А” соответствует букве “Б”, “Б” – “В”, и так далее, а когда алфавит подходит к концу, то начинают брать буквы из начала списка. И выходит, например, следующее: из слова “КОДИРОВАНИЕ” получается “ЛПЕЙСПГБОЙЖ”.

Частным случаем данного метода является ранее рассмотренный шифр Цезаря. Очевидно, что произвольный шифр из класса одноалфавитных методов не является шифром Цезаря (если мощность алфавита текста равна $n$, то число шифров Цезаря равно $n$, а число всех одноалфавитных шифров равно $n!$).Однако и для таких методов легко предложить способы дешифрования, основанные на статистических свойствах шифрованных текстов поскольку открытый и закрытый тексты имеют одинаковые статистические характеристики.

*В лабораторной работе № 1 рассматриваются два варианта одноалфавитного метода с фиксированным смещением и с произвольным (задаваемым) смещении.*

### 1.3.2. Шифрование методом перестановки символов

Суть этого метода заключается в том, что символы текста переставляются по определенным правилам, при этом используются только символы исходного (незашифрованного) текста.

Перестановки в классической криптографии обычно получаются в результате записи исходного текста и чтения шифрованного текста по разным путям геометрической фигуры.

Простейшим примером перестановки является запись исходного текста по строкам некоторой матрицы и чтение его по столбцам этой матрицы.

Последовательность заполнения строк и чтения столбцов может быть любой и задается ключом. Таким образом, для матрицы размером 8*8 (длина блока 64 символа) возможно $1.6*10^9$ ключей, что позволяет на современных компьютерах путем перебора дешифровать заданный текст. Однако для матрицы размером 16*16 (длина блока 256 символов) имеется $1.4*10^{26}$ ключей, и перебор их с помощью современных вычислительных средств весьма затруднителен.

Примером применения метода перестановки может быть также восьмиэлементная таблица, обладающая совокупностью маршрутов, носящих название маршрутов Гамильтона. Последовательность заполнения таблицы каждый раз соответствует нумерации ее элементов. Если длина шифруемого текста не кратна числу элементов, то при последнем заполнении в свободные элементы заносится произвольный символ. Выборка из таблицы для каждого заполнения может выполняться по своему маршруту, при этом маршруты могут использоваться как последовательно, так и в порядке, задаваемом ключом.

Для методов перестановки характерны простота алгоритма, возможность программной реализации и низкий уровень защиты, так как при большой длине исходного текста в его зашифрованном варианте проявляются статистические закономерности ключа, что и позволяет его быстро раскрыть. Другой недостаток этих методов – легкое раскрытие, если уда-

ется направить в систему для шифрования несколько специально подобранных сообщений. Так, если длина блока исходном тексте равна *К* символам, то для раскрытия ключа достаточно пропустить через шифровальную систему *К-1* блоков исходного текста, в которых все символы, кроме одного, одинаковы.

### 1.3.3. Шифрование инверсными символами (по дополнению до 255)

Данный метод шифрования, является частным случаем одноалфавитной замены в алфавите мощности 256. Суть метода заключается в замене символа *ASCII*-кодировки с номером *i* на символ с номером *255-i.* Аналогично проводится и операция расшифрования.

### 1.3.4. Многоалфавитные методы шифрования

Многоалфавитное шифрование (многоалфавитная замена) заключается в том, что для последовательных символов шифруемого текста используются одноалфавитные методы с различными ключами.

Например, первый символ заменяется по методу Цезаря со смещением 14, второй – со смещением 10, и так далее до конца заданного ключа. Затем процедура продолжается периодически. Более общей является ситуация, когда используется не шифр Цезаря, а последовательность произвольных подстановок, соответствующих одноалфавитным методам.

Более наглядным примером подобного шифрования является метод гаммирования. Данный способ преобразования заключается в том, что символы закрываемого текста последовательно складываются с символами некоторой специальной последовательности именуемой гаммой. Такое преобразование иногда называют наложением гаммы на открытый текст.

*Собственно процедура наложения может осуществляться одним из двух способов:*

1) Символы закрываемого текста и гаммы заменяются цифровыми эквивалентами а затем складываются по модулю *К*, где *К* – количество символов алфавита,

$T_ш = (T_o \square T_г) \ mod \ K$ где $T_ш$ – шифротекст,

$T_o$ – открытый текст,

$T_г$ – гамма.

2) Символы текста и гаммы представляются в двоичных кодах, а затем каждая пара двоичных разрядов складывается по модулю 2.

Стойкость шифрования методом гаммирования определяется, главным образом, качеством гаммы, которое определяется двумя характеристиками: длиною периода и случайностью распределения по периоду.

Длиною периода гаммы называется минимальное количество символов, после которого последовательность начинает повторяться. Случайность распределения символов по периоду означает отсутствие закономерностей между появлением различных символов в пределах периода.

*В лабораторной работе № 1 рассматриваются три варианта многоалфавитного метода – с фиксированным ключом, с ключом фиксированной длины и с ключом произвольной длины.*

### 1.3.5. Основные требования, которые предъявляются к методам шифрования информации

1) Сложность и трудоемкость процедур шифрования и расшифрования должны определяться в зависимости от степени секретности защищаемых данных.

2) Надежность закрытия должна быть такой, чтобы секретность не нарушалась даже в том случае, когда злоумышленнику известен способ закрытия.

3) Способ закрытия и набор используемых служебных данных (ключевых установок) не должны быть слишком сложными. Затраты на защитные преобразования должны быть приемлемые при заданном уровне сохранности информации.

4) Выполнение процедур прямого и обратного преобразования должно быть формальным и как можно проще.

5) Процедуры прямого и обратного преобразования не должны зависеть от длины сообщения.

6) Ошибки, возникающие в процессе преобразования, не должны распространяться по системе и вызывать потерю информации. Из-за появления ошибок передачи зашифрованного сообщения по каналам связи не должна исключаться возможность надежной расшифровки текста на приемном конце.

7) Избыточность сообщений, вносимая закрытием должна быть как можно меньшей.

8) Объем ключа не должен затруднять его запоминание и пересылку.

### 1.3.6. Гистограмма текста

Одним из наиболее известных методов криптоанализа является изучение статистических характеристик шифрованных текстов. Графическое отображение совокупности частот встречаемости символов в тексте называют гистограммой этого текста.

Предположим, что мы имеем дело с методом одноалфавитного шифрования. Зная частоту встречаемости букв в алфавите, можно предположить, какая буква была заменена на данную. Например, часто встречаемая буква “О” заменена на редко встречающуюся букву “Щ”.

*Для наглядности, в лабораторной работе № 1 используются двойные гистограммы, отображающие частоту встречаемости символов в исходном и зашифрованном текстах.*

Следует иметь в виду, что вид гистограммы для стандартного распределения зависит от вида исходного текста следующим образом: если

исходный текст содержит символы кириллицы и латинского алфавита, то выводится статистическое распределение для кириллицы и латиницы, если только кириллицы (латиницы) то выводится статистическое распределение для кириллицы (латиницы).

## Практическая работа № 1
## Использование классических криптоалгоритмов подстановки и перестановки для защиты текстовой информации

**Цель работы:** формирование у студентов практических умений исследования классических криптографических алгоритмов многоалфавитной подстановки, многоалфавитной подстановки и перестановки для защиты текстовой информации и формирование гистограмм, отображающих частоту встречаемости символов в тексте для криптоанализа классических шифов.

### 1. Описание лабораторной работы

Для выполнения лабораторной работы необходимо запустить программу *L_LUX.EXE.* На экране дисплея появляется окно, с размещенным в его центре текстовым редактором (для отображения зашифрованных и расшифрованных текстов), в верхней строке окна расположено главное меню, позволяющее пользователю выполнить требуемое действие, чуть ниже основного меню расположена панель инструментов (для управления быстрыми командными кнопками и другими “горячими” элементами управления), а в самом низу окна, под текстовым редактором, находится строка состояния, в которой указывается подсказка и выводится дополнительная информация. Клавиши панели инструментов, для удобства, снабжены всплывающими подсказками.

Для того чтобы попасть в основное меню, необходимо нажать клавишу *F10.* Передвижение по главному меню осуществляется клавишами перемещения курсора. Чтобы вызвать пункт меню, нужно нажать клавишу “ENTER”, вернуться в главное меню или вовсе выйти из него – “ESC”.

А теперь более подробно рассмотрим каждый из пунктов главного меню.

### 1.1. Редактор

Данный пункт основного меню содержит подпункты: создать документ, открыть файл, сохранить файл, выход из программы.

Предварительно, сразу после запуска программы, текстовый редактор недоступен, также недоступными являются почти все пункты главного меню (кроме создания документа, открытия файла, выхода из программы, информации о программе) и большая часть клавиш панели управления (за исключением создания документа, открытия файла и выхода из программы).

**Создать документ (Ctrl+N) –** данный подпункт делает доступным для работы тестовый редактор (пользователь получает право создать свой текстовый файл, который впоследствии можно будет использовать при работе с программой), также появляется возможность использовать все недоступные до этого пункты основного меню и клавиши панели управления.

**Открыть файл (Ctrl+L) –** при выборе этого пункта появляется диалоговое окно, предоставляющее возможность выбора файла для загрузки. При этом содержимое файла будет отображено в окне редактора текстов.

Аналогично пункту “Создать документ” доступным для работы становится текстовый редактор с отображаемым текстом, а также появляется возможность использовать все недоступные до этого пункты основного меню и клавиши панели управления.

**Сохранить файл (Ctrl+S) -** при выборе этого пункта появляется диалоговое окно, позволяющее сохранить на диск содержимое редактора текстов.

**Выход из программы (Ctrl+X) -** при выборе этого пункта появляется диалоговое окно, позволяющее выйти из программы.

### 1.2. Гистограмма

Вывод на экран двух гистограмм, отображающих частоту встречаемости символов в тексте.

***Внимание !*** *До выполнения шифрования и дешифрования вызывать гистограмму не имеет смысла, так как еще не сформированы тексты, для которых будет просматриваться гистограмма.*

Имеется возможность просмотра следующих сочетаний гистограмм:

- гистограммы исходного и зашифрованного файла,
- гистограммы зашифрованного и расшифрованного файла,
- гистограммы стандартного распределения и зашифрованного текста,
- гистограммы стандартного распределения и расшифрованного текста.

В гистограммах с целью масштабирования используются левая и правая клавиши мыши.

Например, после шифрования текста большого объема пользователь хочет посмотреть гистограммы исходного и зашифрованного файла. Поскольку размеры текста достаточно большие, то на экран будут выведены две гистограммы с большим количеством столбцов в каждой (столбец соответствует одному символу текста), однако трудно будет сказать, какой из этих столбцов соответствует тому или иному символу текста, и какова частота встречаемости данного символа. Поэтому у пользователя есть возможность увеличить масштаб любой из двух гистограмм с целью более точного определения требуемого значения частоты встречаемости конкретного символа. Для этого необходимо навести указатель мыши на левую границу того участка гистограммы, который требуется увеличить, затем нажать левую клавишу мыши, и не отпуская ее растянуть прямоугольник так, чтобы его нижний правый угол совпал с правой границей увеличиваемого участка гистограммы. После этого следует отпустить левую клавишу мыши и на экране появится увеличенное изображение нужного участка.

Причем, нажав и не отпуская правую клавишу мыши, можно перемещать гистограмму в любом направлении с целью изучения всего полученного распределения в увеличенном масштабе.

Для того, чтобы от увеличенного масштаба вернуться к исходному виду, нужно навести указатель мыши на гистограмму, затем нажать левую клавишу мыши, и, не отпуская ее, снизу вверх растянуть небольшой по размерам прямоугольник, после этого следует отпустить левую клавишу мыши и на экране появится исходное изображение гистограммы.

### 1.3. Шифрование

Выполнение шифрования текстового файла осуществляется одним из семи методов, рассматриваемых в лабораторной работе.

1. Одноалфавитный метод (с фиксированным смещением).
2. Одноалфавитный метод с задаваемым смещением (от 2 до 20).
3. Перестановка символов.
4. По дополнению до 255 (инверсный метод).
5. Многоалфавитный метод (с фиксированным ключом).
6. Многоалфавитный метод с ключом фиксированной длины.
7. Многоалфавитный метод с ключом произвольной длины.

Выбор метода шифрования производится как мышкой, так и клавишами перемещения курсора и клавишей “ENTER”.

Затем появляется окно в котором в зависимости от метода шифрования требуется указать те или иные параметры, и либо подтвердить процесс кодировки, либо отказаться от него. После этого в окне редактора будет выдан зашифрованный текст.

### 1.4. Расшифрование

Аналогично предыдущему пункту выбирается метод расшифрования (должен соответствовать методу, которым был зашифрован файл).

Снова появляется окно, в котором в зависимости от метода расшифрования требуется указать те или иные параметры, и либо подтвердить процесс расшифрования, либо отказаться от него.

После этого в окно редактора будет выдан расшифрованный текст. При правильном расшифровании, полученный текст совпадает с исходным.

### 1.5. Дополнительная информация

Программа предусматривает возможность посмотреть дополнительную информацию ('Помощь Ctrl+F9'), справочную информацию об используемых методах шифрования ('О методах Ctrl+F10'), сведения о программе ('О программе Ctrl+F11').

## 2. Формируемые компетенции

Данная Практическая работа направлена на формирование следующих компетенций выпускника бакалавриата:

ОПК-1: способностью инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем;

ПК-5: способностью сопрягать аппаратные и программные средства в составе информационных и автоматизированных систем.

## 3. Теоретическая часть

Для выполнения лабораторной работы необходимо запустить программу L_LUX. На экране дисплея появится окно с размещенным в его центре текстовым редактором. В верхней строке окна расположено главное меню, позволяющее пользователю выполнить требуемое действие. Чуть ниже основного меню расположена панель инструментов, а в самом низу окна, под текстовым редактором находится строка состояния, в которой указывается подсказка и выводится дополнительная информация. Кла-

виши панели инструментов для удобства снабжены всплывающими подсказками.

Для того чтобы попасть в основное меню, не обходимо нажать клавишу F10. Передвижение по главному меню осуществляется мышью.

Пример работы с программой.

Рассмотрим одноалфавитное шифрование с фиксированным ключом.

Загрузите в окно редактора исходный текст (любой) длиной от 10 до 15 предложений. Сохраните его, это необходимо для последующей работы с этим файлом. Затем вызовите пункт меню ШИФРОВАНИЕ, выберите одноалфавитный метод (с фиксированным смещением). В появившемся окне нажмите клавишу ЗАШИФРОВАТЬ. После того как шифрование выполнено, можно в редакторе просмотреть зашифрованный текст.

Перейдите к пункту меню ГИСТОГРАММА. Выберите тип гистограмм, отображающий гисторраммы исходного и шифрованного текста. Проанализируйте гистограммы. Они должны иметь примерно одинаковый вид.

Чтобы узнать ключ шифра (смещение второго алфавита относительно первого), необходимо по гистограммам найти символы, имеющие одинаковую частоту встречаемости. Например, самый частый символ в первой гистограмме при шифровании должен перейти в самый частый символ во второй гистограмме. Таким образом, найдя два самых часто встречаемых символа в обеих гистограммах, можно по стандартной таблице ASCII-кодов вычислить смещение. Зная смещение и таблицу кодировки символов, текст можно легко дешифровать. Вызвав меню ДЕШИФРОВАНИЕ, можно провести те же действия в автоматическом режиме.

Примечание. При шифровании и дешифровании из таблицы кодировки не используются символы с кодами 176-223 и 240-255, то есть при ручной расшифровке эти символы следует пропускать и считать, что, например, символ Я имеет код 159, а 223, аналогично П не 175, а 239.

Иногда в гистограммах под столбцами, показывающими частоту встречаемых символов, изображены не сами символы, а их табличные коды в квадратных скобках.

**4. Оборудование и материалы**

К аппаратному обеспечению лабораторной работы относятся компьютерные классы с установленным сетевым оборудованием, а также

- операционная система Windows;

- программа L_LUX в среде C++.

**5. Указания по технике безопасности**

Требования безопасности перед началом работ:

– ЗАПРЕЩАЕТСЯ: переодеваться, пользоваться огнем, курить, принимать пищу в лаборатории;

– убедиться в целостности электрических розеток и разъемов. Запрещается работать с неисправным электроинструментом, разбивать и самостоятельно ремонтировать его;

– в лаборатории необходимо быть в сменной обуви;

– включение компьютера производить только после получения допуска по выполняемой работе и разрешения преподавателя или лаборанта.

Требования безопасности во время работы:

– выполняя лабораторную работу, студенты обязаны использовать только вычислительную технику, периферийное оборудование, соединительные кабели, измерительное оборудование и дискеты, непосредственно относящиеся к данному лабораторному занятию;

– подключение и отсоединение составляющих вычислительного комплекса производить только при полном снятии напряжения со всех составляющих вычислительного комплекса.

Требования безопасности по окончании работы:

– по команде преподавателя студенты обязаны выключить компьютер и составляющие вычислительного комплекса;

– рабочее место привести в порядок.

**6. Задания**

**Задание №1**. Для одноалфавитного метода с фиксированным смещением определить установленное в программе смещение. Для этого следует:

- просмотреть предварительно созданный с помощью редактора свой текстовый файл;

- выполнить для этого файла шифрование;

- просмотреть в редакторе зашифрованный файл;

- просмотреть гистограммы исходного и зашифрованного текстов;

- описать гистограммы (в чем похожи, в чем отличаются) и определить, с каким смещением было выполнено шифрование;

- расшифровать зашифрованный текст:

- С помощью гистограммы, после чего проверить в редакторе правильность расшифрования;

- Вручную с помощью гистограмм, описать и объяснить процесс дешифрования.

В отчете для каждого метода шифрования описывается последовательность выполняемых действий, имена всех использованных файлов, полученные гистограммы, указывается найденное смещение, описывается процесс дешифрования.

Преподавателю предоставляется отчет о проделанной работе и все использованные и созданные файлы.

**Задание №2.** Для одноалфавитного метода с заданным смещением (шифр Цезаря) следует:

- выполнить шифрование с произвольным смещением для своего исходного текста;

- просмотреть и описать гистограммы исходного и зашифрованного

текстов, определить смещение для нескольких символов;

- расшифровать текст с помощью программы;

- дешифровать зашифрованный шифром Цезаря текст с помощью программы методом перебора смещения; указать в отчете с каким смещением был зашифрован файл.

**Задание №3**. Для метода перестановки символов дешифровать зашифрованный файл. Для этого необходимо определить закон перестановки символов открытого текста. Создайте небольшой файл длиной в несколько символов с известным вам текстом, зашифруйте его. Посмотрите гистограммы, опишите их, ответьте можно ли извлечь из них полезную для дешифрования инфориацию. Сравните с помощью редактора ваш исходный текст и зашифрованный текст и определите закон перестановки символов.

Дешифруйте файл:

- вручную и объясните ваши действия;

- с помощью программы.

**Задание №4**. Для инверсного кодирования (по дополнению до 255)

- выполните шифрование для своего произвольного файла;

- посмотрите гистограммы исходного и зашифрованного текстов, опишите гистограммы и определите смещение для нескольких символов;

- дешифруйте зашифрованный текст, проверьте в редакторе правильность дешифрования.

**Задание №5**. Для многоалфавитного шифрования с фиксированным ключом определите, сколько одноалфавитных методов и с каким смещением используется в программе. Для этого нужно создать свой файл, состоящий из строки одинаковых символов, выполнить для него шифрование и по гистограмме определить способ шифрования и набор смещений.

**Задание №6.** Для многоалфавитного шифрования с ключом фиксированной длины:

- выполните шифрование и определите по гистограмме, какое смещение получает каждый символ для файла, состоящего из строки одинако-

вых символов;

- выполните шифрование и расшифрование для файла произвольно текста;

- просмотрите и опишите гистограммы исходного и зашифрованного текстов; ответьте, какую информацию можно получить и гистограмм.

**Задание №7**. Для многоалфавитного шифрования с произвольным паролем задание полностью аналогично заданию 6.

**7. Содержание отчета**

Отчет должен содержать:

1. наименование работы;
2. цель работы;
3. рабочие схемы;
4. основные соотношения и расчетные формулы;
5. таблицы;
6. выводы по работе.

**8. Контрольные вопросы**

Привести в отчете ответы на контрольные вопросы в соответствии с номером варианта, указанным преподавателем (табл. 1.1).

Таблица 1.1 – Контрольные вопросы

| Номер варианта | Контрольные вопросы |
|---|---|
| 1, 5, 7, 3, 9, 18, 28 | Какие вы знаете методы криптографической защиты файлов? |
| 2, 4, 6, 8, 20, 22, 24, 26, 30 | В чем преимущества и недостатки одноалфавитных методов? |
| 11, 13, 15, 10, 17, 19, 27 | Если необходимо зашифровать текст, содержащий важную информацию, какой метод из рассмотренных вы выберете? Обоснуйте свой выбор |
| 12, 14, 16, 21, 23, 25, 29 | Целесообразно ли повторно применять для уже зашифрованного текста: |

| | |
|---|---|
| | а) метод многоалфавитного шифрования;<br>б) метод Цезаря? |

**9. Список литературы, рекомендованный к использованию по данной теме**

1. Жук, А.П., Жук, Е.П., Лепешкин, О.М., Тимошкин, А.И. Защита информации: Учебное пособие для бакалавров [Текст] / А.П. Жук и др. – М.: ИНФРА-М, 2015. -392с.

2. Шаньгин, В.Ф. Защита компьютерной информации. Эффективные методы и средства [Текст] / В.Ф. Шаньгин – М.: ДМК Пресс, 2012. – 544 с.

## Практическая работа № 2
## Исследование различных методов защиты текстовой информации и их стойкости на основе подбора ключей

**Цель лабораторной работы:** исследования методов шифрования/расшифрования перестановкой символов, подстановкой, гаммированием, использованием таблицы Виженера, сравнение стойкости различных методов на основе атак путем перебора всех возможных ключей.

*В лабораторной работе рассматривается способ вскрытия шифра, основанный на переборе всех вариантов ключа. Критерием правильности варианта служит наличие в тексте «вероятного слова».*

*Перебирается множество всех возможных ключей, шифрованный текст расшифровывается на каждом ключе. В получившемся «псевдооткрытом» тексте ищется вероятное слово. Если такого слова нет, текущий текст бракуется и осуществляется переход к следующему ключу. Если такое слово найдено, на экран выводится вариант ключа. Затем перебор ключей продолжается до тех пор, пока не исчерпается все множество вариантов. Возможно обнаружение нескольких ключей, при которых в «псевдооткрытых текстах» имеется вероятное слово.*

*После завершения перебора необходимо расшифровать текст на найденных ключах. «Псевдооткрытый текст» выводится на экран для визуального контроля.*
*Если оператор признает текст открытым, то работа по вскрытию заканчивается. Иначе, данный вариант ключа бракуется и осуществляется переход к следующему ключу.*

### 1. Формируемые компетенции

Данная Практическая работа направлена на формирование следующих компетенций выпускника бакалавриата:

ОПК-1: способностью инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем;

ПК-5: способностью сопрягать аппаратные и программные средства в составе информационных и автоматизированных систем.

## 2. Теоретическая часть

Для выполнения лабораторной работы необходимо запустить программу *LAB_RAB.exe,* используемую для шифрования/расшифрования, а также дешифрования (методом протяжки вероятного слова) файлов. Система реализует следующие функции:

– ввод, удаление и селекция ключей пользователя;

– поддержка списка ключей;

– шифрование и расшифрование текста;

– дешифрование текста путем подбора ключей, методом протяжки вероятного слова.

Система поддерживает следующие методы криптографического преобразования информации:

– замена;

– перестановка;

– гаммирование;

– таблица Виженера.

При запуске утилит шифрования и расшифрования у пользователя запрашивается подтверждение на правильность выбранного метода для работы и соответствия заданного ключа целям пользователя (также всегда при изменении файла в текстовом редакторе выдается запрос на сохранение изменений при каждом шаге, дальнейшее развитие которого приведет к необратимым изменениям в файле и потере изначальной информации).

Описание интерфейса.

- Окно текстового редактора с широким набором дополнительных функций.
- Таблица всех ключей, введенных в систему с быстрым доступом для ввода, удаления или выбора текущего ключа.
- Список всех методов шифрования для быстрого и удобного переключения между ними.

- Основное меню (вверху экрана).
- Дополнительное меню (вызывается нажатием правой кнопки мыши).
- Набор вспомогательных кнопок для быстрого и удобного интерфейса.
- Поля вывода текущего состояния системы:
- текущий ключ;
- вероятное слово;
- сила ключа для протяжки.

**Пример работы с программой**

***Внимание!*** *Будьте внимательны при установке параметров работы, так как в процессе вычисления по ходу работы эти параметры изменить уже не удастся. После запуска программы абсолютно все рабочие поля пустые и необходимо провести первоначальные настройки для работоспособности системы.*

1. Вводится список ключей.

2. Вводится вероятное слово (необязательно вначале, до его ввода все меню запуска протяжки все равно недоступны).

3. Выбирается необходимый метод шифрования.

4. Загружается исходный или зашифрованный файл (открываются соответствующие меню для шифрования и расшифрования).

5. Запускается необходимый процесс:

– шифрование;
– расшифрование;
– протяжка вероятного слова;
– конвертация текста.

6. Продолжение работы в любом порядке в соответствии с описанными пунктами.

7. При завершении работы не забудьте сохранить необходимые результаты (при закрытии и загрузке новых файлов система автоматически запрашивает подтверждение на запись).

**Шифрование**

1. Открыть файл.

2. Внести необходимые изменения.

3. Настроить соответствующие параметры:

– тип шифрования;

– ключ;

– прочие.

4. Запустить процесс шифрования через пункт меню УТИЛИТЫ \ ЗАШИФРОВАТЬ F5.

***Внимание!*** *При шифровании файла все внесенные пользователем изменения до текущего момента времени будут сохранены на жестком диске.*

**Расшифрование**

1. Открыть файл.

2. Произвести необходимые изменения.

3. Настроить соответствующие параметры:

– тип шифрования;

– прочие.

4. Запустить процесс расшифрования через пункт меню УТИЛИТЫ \ РАСШИФРОВАТЬ F6.

***Внимание!*** *При расшифровании файла все проведенные пользователем изменения до текущего момента времени будут сохранены на жестком диске.*

**Протяжка вероятного слова (дешифрование)**

***Внимание!*** *Мощность ключа задастся заранее в опции "сила ключа".*

*Длина ключа значительно влияет на время протяжки вероятного слова (в худшем случае имеем дело с логарифмическим алгоритмом).*

1. Вводится вероятное слово (длинной от 1(3) до 9)

2. Для отделения вновь найденных ключей от предыдущих между ними добавляется надпись "подбор".

3. Перебор ключей.

4. Расшифровывается первая вся строка текста по текущему ключу.

5. Порциями, равными длине вероятного слова, сравнивается содержимое этой строки со значением вероятного слова.

6. Если найдено хоть одно совпадение, запоминаем ключ.

7. Переходим к новому ключу.

9. Результаты должны содержаться в списке ключей. Если совпадений не найдено, в список ключей ничего не добавляется.

**Операции с ключами**

С базой ключей могут осуществляться следующие действия:

– добавить новый ключ;

– удалить одну запись;

– изменить активную запись;

– очистить всю таблицу введенных ключей .

*Примечание. Под словами "работа с таблицей ключей" имеются ввиду ключи, введенные для использования в двух методах (гаммирования и таблица Виженера).*

**Ключи для перестановки**

В каждый момент времени в системе может быть только один текущий ключ для перестановки.

Правила ввода ключа для перестановки:

1. При переключении в списке поддерживаемых системой методов шифрования на пункт "Перестановка" вызывается окно ввода ключа перестановки. Окно состоит из двух кнопок (Отмены и Выхода без изменений и кнопки Enter – подтверждение установленной длины ключа) и окна задания длины ключа для перестановки.

2. В окне задания длины ключа необходимо выбрать необходимую длину (параметры заменяются в пределах 1 … 9), и подтвердить желание использовать ключ именно такой длины.

3. После подтверждения в окне высветятся кнопки с цифрами на лицевой стороне (в количестве, равном длине ключа), при нажатии на кнопку происходит фиксация кнопки (ее обесцвечивание) для

невозможности ее дальнейшего использования (так как все цифры в ключе перестановки должны быть неповторяющимися).

4. После перебора всех кнопок система запоминает введенный ключ, выводит его в поле ввода ключей и выходит из окна ввода ключа перестановки в окно основной программы.

**4. Оборудование и материалы**

К аппаратному обеспечению лабораторной работы относятся компьютерные классы с установленным сетевым оборудованием, а также

- операционная система Windows;

- программа *LAB_RAB* в среде C++.

**5. Указания по технике безопасности**

Требования безопасности перед началом работ:

– ЗАПРЕЩАЕТСЯ: переодеваться, пользоваться огнем, курить, принимать пищу в лаборатории;

– убедиться в целостности электрических розеток и разъемов. Запрещается работать с неисправным электроинструментом, разбивать и самостоятельно ремонтировать его;

– в лаборатории необходимо быть в сменной обуви;

– включение компьютера производить только после получения допуска по выполняемой работе и разрешения преподавателя или лаборанта.

Требования безопасности во время работы:

– выполняя лабораторную работу, студенты обязаны использовать только вычислительную технику, периферийное оборудование, соединительные кабели, измерительное оборудование и дискеты, непосредственно относящиеся к данному лабораторному занятию;

– подключение и отсоединение составляющих вычислительного комплекса производить только при полном снятии напряжения со всех составляющих вычислительного комплекса.

Требования безопасности по окончании работы:

– по команде преподавателя студенты обязаны выключить компьютер и составляющие вычислительного комплекса;

– рабочее место привести в порядок.

**6. Задания**

Для выполнения лабораторной работы необходимо запустить программу *LAB_RAB.exe,* используемую для шифрования/расшифрования, а также дешифрования (методом протяжки вероятного слова) файлов.

Система реализует следующие функции:

– ввод, удаление и селекция ключей пользователя;

– поддержка списка ключей;

– шифрование и расшифрование текста;

– дешифрование текста путем подбора ключей, методом протяжки вероятного слова.

Система поддерживает следующие методы криптографического преобразования информации:

– замена;

– перестановка;

– гаммирование;

– таблица Виженера.

При запуске утилит шифрования и расшифрования у пользователя запрашивается подтверждение на правильность выбранного метода для работы и соответствия заданного ключа целям пользователя (также всегда при изменении файла в текстовом редакторе выдается запрос на сохранение изменений при каждом шаге, дальнейшее развитие которого приведет к необратимым изменениям в файле и потере изначальной информации).

Описание интерфейса:

1. Окно текстового редактора с широким набором дополнительных функций.

2. Таблица всех ключей, введенных в систему с быстрым доступом для ввода, удаления или выбора текущего ключа.

3. Список всех методов шифрования для быстрого и удобного переключения между ними.

4. Основное меню (вверху экрана).

5. Дополнительное меню (вызывается нажатием правой кнопки мыши).

6. Набор вспомогательных кнопок для быстрого и удобного интерфейса.

7. Поля вывода текущего состояния системы:

- текущий ключ;

- вероятное слово;

- сила ключа для протяжки.

Пример работы с программой

Внимание! Будьте внимательны при установке параметров работы, так как в процессе вычисления по ходу работы эти параметры изменить уже не удастся.

После запуска программы абсолютно все рабочие поля пустые и необходимо провести первоначальные настройки для работоспособности системы.

1. Вводится список ключей.

2. Вводится вероятное слово (необязательно вначале, до его ввода все меню запуска протяжки все равно недоступны).

3. Выбирается необходимый метод шифрования.

4. Загружается исходный или зашифрованный файл (открываются соответствующие меню для шифрования и расшифрования).

5. Запускается необходимый процесс:

– шифрование;

– расшифрование;

– протяжка вероятного слова;

– конвертация текста.

6. Продолжение работы в любом порядке в соответствии с описанными пунктами.

7. При завершении работы не забудьте сохранить необходимые результаты (при закрытии и загрузке новых файлов система автоматически запрашивает подтверждение на запись).

Шифрование

1. Открыть файл.

2. Внести необходимые изменения.

3. Настроить соответствующие параметры:

– тип шифрования;

– ключ;

– прочие.

4. Запустить процесс шифрования через пункт меню

УТИЛИТЫ \ ЗАШИФРОВАТЬ F5.

Внимание! При шифровании файла все внесенные пользователем изменения до текущего момента времени будут сохранены на жестком диске.

Расшифрование

1. Открыть файл.

2. Произвести необходимые изменения.

3. Настроить соответствующие параметры:

– тип шифрования;

– прочие.

4. Запустить процесс расшифрования через пункт меню

УТИЛИТЫ \ РАСШИФРОВАТЬ F6.

Внимание! При расшифровании файла все проведенные пользователем изменения до текущего момента времени будут сохранены на жестком диске.

Протяжка вероятного слова (дешифрование)

Внимание! Мощность ключа задастся заранее в опции "сила ключа".

Длина ключа значительно влияет на время протяжки вероятного слова (в худшем случае имеем дело с логарифмическим алгоритмом).

1. Вводится вероятное слово (длинной от 1(3) до 9)

2. Для отделения вновь найденных ключей от предыдущих между ними добавляется надпись "подбор".

3. Перебор ключей.

4. Расшифровывается первая вся строка текста по текущему ключу.

5. Порциями, равными длине вероятного слова, сравнивается содержимое этой строки со значением вероятного слова.

6. Если найдено хоть одно совпадение, запоминаем ключ.

7. Переходим к новому ключу.

8. Переходим к следующей строке.

9. Результаты должны содержаться в списке ключей. Если совпадений не найдено, в список ключей ничего не добавляется.

Операции с ключами

С базой ключей могут осуществляться следующие действия:

– добавить новый ключ;

– удалить одну запись;

– изменить активную запись;

– очистить всю таблицу введенных ключей.

Примечание. Под словами "работа с таблицей ключей" имеются ввиду ключи, введенные для использования в двух методах (гаммирования и таблица Виженера).

Ключи для перестановки

В каждый момент времени в системе может быть только один текущий ключ для перестановки.

Правила ввода ключа для перестановки:

1. При переключении в списке поддерживаемых системой методов шифрования на пункт "Перестановка" вызывается окно ввода ключа пере-

становки. Окно состоит из двух кнопок (Отмены и Выхода без изменений и кнопки Enter – подтверждение установленной длины ключа) и окна задания длины ключа для перестановки.

2. В окне задания длины ключа необходимо выбрать необходимую длину (параметры заменяются в пределах 1 … 9), и подтвердить желание использовать ключ именно такой длины.

3. После подтверждения в окне высветятся кнопки с цифрами на лицевой стороне (в количестве, равном длине ключа), при нажатии на кнопку происходит фиксация кнопки (ее обесцвечивание) для невозможности ее дальнейшего использования (так как все цифры в ключе перестановки должны быть неповторяющимися).

4. После перебора всех кнопок система запоминает введенный ключ, выводит его в поле ввода ключей и выходит из окна ввода ключа перестановки в окно основной программы.

**Задание №1.**

1. Ознакомиться с описанием лабораторной работы и заданием.

2. Выполнить настройку программы: выбрать метод шифрования; ввести ключи для всех методов; ввести вероятное слово; осуществить все остальные системные настройки.

**Задание №2.** Для метода замены (одноалфавитного метода):

- выбрать данный алгоритм в списке доступных методов шифрования;
- установить необходимое смещение;
- открыть произвольный файл;
- просмотреть содержимое исходного файла;
- выполнить для этого файла шифрование (при необходимости можно задать имя зашифрованного файла);
- просмотреть в редакторе зашифрованный файл;
- ввести вероятное слово;
- ввести вероятную длину ключа (кроме метода замены);

- подобрать ключ;
- выполнить расшифрование со всеми найденными ключами;
- найти в каком-либо из расшифрованных файлов правильно расшифрованное ключевое слово;
- расшифровать файл исходным ключом;
- проверить результат.

**Задание №3.** Для метода перестановки:

- выбрать метод перестановки;
- в открывшемся окне ввода ключа перестановки символов указать сначала длину этого ключа, а затем из появившихся кнопок составить необходимую комбинацию для ключа, нажимая на кнопки в заданном порядке; при этом уже использованные кнопки становятся недоступными для предотвращения их повторного ввода;
- далее действия полностью соответствуют изложенным в предыдущем пункте задания.

**Задание №4.** Для метода гаммирования:

- выбрать метод гаммирования;
- ввести ключ гаммирования;
- полностью повторить из задания 3 второй пункт.

**Задание №5.** Для таблицы Виженера все действия повторяются из зад. 4 (метод гаммирования).

В отчете для каждого метода шифрования описывается последовательность выполняемых действий, указываются имена всех использованных файлов, исходные и найденные ключи, описывается процесс дешифрования.

Преподавателю предоставляется отчет о проделанной работе и все использованные файлы.

### 7. Содержание отчета

В отчете для каждого метода шифрования описывается последовательность выполняемых действий, указываются имена всех использован-

ных файлов, исходные и найденные ключи, описывается процесс дешифрования.

Преподавателю предоставляется отчет о проделанной работе и все использованные файлы.

**8. Контрольные вопросы**

| Номер варианта | Контрольные вопросы |
|---|---|
| 1,5,7, 3,9,18,28 | Чем отличается "псевдооткрытый" текст (текст, полученный при расшифровке по ложному ключу) от настоящего открытого текста? |
| 2,4,6,8, 20,22,24, 26,30 | Как зависит время вскрытия шифра описанным выше способом подбора ключей от длины "вероятного" слова? |
| 11,13,15, 10,17,19, 27 | Зависит ли время вскрытия шифра гаммирования (или таблицы Виженера) от мощности алфавита гаммы? |
| 12,14,16 21,23.25, 29 | В чем недостатки метода дешифрования с использованием протяжки вероятного слова? |

**9. Список литературы, рекомендованный к использованию по данной теме**

1. Жук, А.П., Жук, Е.П., Лепешкин, О.М., Тимошкин, А.И. Защита информации: Учебное пособие для бакалавров [Текст] / А.П. Жук и др. – М.: ИНФРА-М, 2015. -392с.

2. Шаньгин, В.Ф. Защита компьютерной информации. Эффективные методы и средства [Текст] / В.Ф. Шаньгин – М.: ДМК Пресс, 2012. – 544 с.

## Практическая работа № 3

## Стандарт симметричного шифрования *Aes Rijndael*

**Цель работы:** исследования принципов шифрования, используемыми в алгоритме симметричного шифрования ***AES RIJNDAEL*** и умений делать самостоятельные выводы по результатам исследований.

### 2. Формируемые компетенции

Данная Практическая работа направлена на формирование следующих компетенций выпускника бакалавриата:

ОПК-1: способностью инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем;

ПК-5: способностью сопрягать аппаратные и программные средства в составе информационных и автоматизированных систем.

### 3. Теоретическая часть

**Состояние, ключ шифрования и число циклов.** Rijndael представляет собой итеративный блочный шифр, имеющий переменную длину блоков и различные длины ключей. Длина ключа и длина блока могут быть независимо друг от друга 128, 192 или 256 бит.

Разнообразные преобразования работают с промежуточным результатом, называемым Состоянием (State).

Состояние можно представить в виде прямоугольного массива байтов. Этот массив имеет 4 строки, а число столбцов обозначено как Nb и равно длине блока, деленной на 32.

Ключ шифрования также представлен в виде прямоугольного массива с четырьмя строками. Число столбцов обозначено как Nk и равно длине ключа, деленной на 32. Это показано на рис. 1.

В некоторых случаях ключ шифрования показан как линейный массив четырехбайтовых слов. Слова состоят из 4-х байтов, которые находятся в одном столбце (при представлении в виде прямоугольного массива).

| | | | | | |
|---|---|---|---|---|---|
| $a_{0,0}$ | $a_{0,1}$ | $a_{0,2}$ | $a_{0,3}$ | $a_{0,4}$ | $a_{0,5}$ |
| $a_{1,0}$ | $a_{1,1}$ | $a_{1,2}$ | $a_{1,3}$ | $a_{1,4}$ | $a_{1,5}$ |
| $a_{2,0}$ | $a_{2,1}$ | $a_{2,2}$ | $a_{2,3}$ | $a_{2,4}$ | $a_{2,5}$ |
| $a_{3,0}$ | $a_{3,1}$ | $a_{3,2}$ | $a_{3,3}$ | $a_{3,4}$ | $a_{3,5}$ |

| | | | |
|---|---|---|---|
| $k_{0,0}$ | $k_{0,1}$ | $k_{0,2}$ | $k_{0,3}$ |
| $k_{1,0}$ | $k_{1,1}$ | $k_{1,2}$ | $k_{1,3}$ |
| $k_{2,0}$ | $k_{2,1}$ | $k_{2,2}$ | $k_{2,3}$ |
| $k_{3,0}$ | $k_{3,1}$ | $k_{3,2}$ | $k_{3,3}$ |

Рисунок 1. Пример представления Состояния (Nb=6) и Ключа шифрования (Nk=4)

Входные данные для шифра обозначаются как байты состояния в порядке a0,0, a1,0, a2,0, a3,0, a0,1, a1,1, a2,1, a3,1 ... После завершения действия шифра выходные данные получаются из байтов состояния в том же порядке. Число циклов обозначено как Nr и зависит от значений Nb и Nk.

Таблица 4.1. Число циклов (Nr) как функция от длины ключа и длины блока

| **Nr** | Nb = 4 | Nb = 6 | Nb = 8 |
|---|---|---|---|
| Nk = 4 | 10 | 12 | 14 |
| Nk = 6 | 12 | 12 | 14 |
| Nk = 8 | 14 | 14 | 14 |

**Цикловое преобразование.** Цикловое преобразование состоит из четырех различных преобразований: замена байт, сдвиг строк, замешивание столбцов, добавление циклового ключа. Последний цикл отличается от простого цикла только отсутствием замешивания столбцов.

```
Round (State, RoundKey)
{
ByteSub(State); // замена байт
ShiftRow(State); // сдвиг строк
MixColumn(State); // замешивание столбцов
AddRoundKey(State, RoundKey); // добавление циклового ключа
}
```

Последний цикл шифра немного отличается:

```
FinalRound(State, RoundKey)
{
ByteSub(State); // замена байт
ShiftRow(State); // сдвиг строк
AddRoundKey(State, RoundKey); // добавление циклового ключа
}
```

Как можно заметить, последний цикл отличается от простого цикла только отсутствием замешивания столбцов. Каждое из приведенных преобразований подробно рассмотрено далее.

**Преобразование замешивания столбцов (MixColumn)**

Преобразование представляет собой умножение состояния на матрицу ME при шифровании или матрицу MD при расшифровании:

ME State

MD State

$$\begin{vmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{vmatrix} \otimes \begin{vmatrix} b1 & b5 & b9 & b13 \\ b2 & b6 & b10 & b14 \\ b3 & b7 & b11 & b15 \\ b4 & b8 & b12 & b16 \end{vmatrix}$$

```
b1 = (b1 * 2) XOR (b2*3) XOR (b3*1) XOR (b4*1)
```

Умножение двух байт происходит по следующему алгоритму:

- если один из байт равен 0, результатом будет 0;
- если один из байт равен 1, результатом будет другой байт;
- в остальных случаях происходит замена каждого байта по таблице L, заменённые байты складываются, при необходимости вычитается 255 для попадания в интервал [0, 255] и происходит замена по таблице E, что и даёт результат.

```
L = {
,          0x00, 0x19, 0x01, 0x32, 0x02, 0x1A, 0xC6, 0x4B, 0xC7, 0x1B, 0x68, 0x33, 0xEE, 0xDF,
0x03,
0x64, 0x04, 0xE0, 0x0E, 0x34, 0x8D, 0x81, 0xEF, 0x4C, 0x71, 0x08, 0xC8, 0xF8, 0x69, 0x1C, 0xC1,
0x7D, 0xC2, 0x1D, 0xB5, 0xF9, 0xB9, 0x27, 0x6A, 0x4D, 0xE4, 0xA6, 0x72, 0x9A, 0xC9, 0x09, 0x78,
0x65, 0x2F, 0x8A, 0x05, 0x21, 0x0F, 0xE1, 0x24, 0x12, 0xF0, 0x82, 0x45, 0x35, 0x93, 0xDA, 0x8E,
0x96, 0x8F, 0xDB, 0xBD, 0x36, 0xD0, 0xCE, 0x94, 0x13, 0x5C, 0xD2, 0xF1, 0x40, 0x46, 0x83, 0x38,
0x66, 0xDD, 0xFD, 0x30, 0xBF, 0x06, 0x8B, 0x62, 0xB3, 0x25, 0xE2, 0x98, 0x22, 0x88, 0x91, 0x10,
0x7E, 0x6E, 0x48, 0xC3, 0xA3, 0xB6, 0x1E, 0x42, 0x3A, 0x6B, 0x28, 0x54, 0xFA, 0x85, 0x3D, 0xBA,
0x2B, 0x79, 0x0A, 0x15, 0x9B, 0x9F, 0x5E, 0xCA, 0x4E, 0xD4, 0xAC, 0xE5, 0xF3, 0x73, 0xA7, 0x57,
0xAF, 0x58, 0xA8, 0x50, 0xF4, 0xEA, 0xD6, 0x74, 0x4F, 0xAE, 0xE9, 0xD5, 0xE7, 0xE6, 0xAD, 0xE8,
0x2C, 0xD7, 0x75, 0x7A, 0xEB, 0x16, 0x0B, 0xF5, 0x59, 0xCB, 0x5F, 0xB0, 0x9C, 0xA9, 0x51, 0xA0,
0x7F, 0x0C, 0xF6, 0x6F, 0x17, 0xC4, 0x49, 0xEC, 0xD8, 0x43, 0x1F, 0x2D, 0xA4, 0x76, 0x7B, 0xB7,
0xCC, 0xBB, 0x3E, 0x5A, 0xFB, 0x60, 0xB1, 0x86, 0x3B, 0x52, 0xA1, 0x6C, 0xAA, 0x55, 0x29, 0x9D,
0x97, 0xB2, 0x87, 0x90, 0x61, 0xBE, 0xDC, 0xFC, 0xBC, 0x95, 0xCF, 0xCD, 0x37, 0x3F, 0x5B, 0xD1,
0x53, 0x39, 0x84, 0x3C, 0x41, 0xA2, 0x6D, 0x47, 0x14, 0x2A, 0x9E, 0x5D, 0x56, 0xF2, 0xD3, 0xAB,
0x44, 0x11, 0x92, 0xD9, 0x23, 0x20, 0x2E, 0x89, 0xB4, 0x7C, 0xB8, 0x26, 0x77, 0x99, 0xE3, 0xA5,
0x67, 0x4A, 0xED, 0xDE, 0xC5, 0x31, 0xFE, 0x18, 0x0D, 0x63, 0x8C, 0x80, 0xC0, 0xF7, 0x70, 0x07
};

E = {
0x01, 0x03, 0x05, 0x0F, 0x11, 0x33, 0x55, 0xFF, 0x1A, 0x2E, 0x72, 0x96, 0xA1, 0xF8, 0x13,
0x35,
0x5F, 0xE1, 0x38, 0x48, 0xD8, 0x73, 0x95, 0xA4, 0xF7, 0x02, 0x06, 0x0A, 0x1E, 0x22, 0x66, 0xAA,
0xE5, 0x34, 0x5C, 0xE4, 0x37, 0x59, 0xEB, 0x26, 0x6A, 0xBE, 0xD9, 0x70, 0x90, 0xAB, 0xE6, 0x31,
0x53, 0xF5, 0x04, 0x0C, 0x14, 0x3C, 0x44, 0xCC, 0x4F, 0xD1, 0x68, 0xB8, 0xD3, 0x6E, 0xB2, 0xCD,
0x4C, 0xD4, 0x67, 0xA9, 0xE0, 0x3B, 0x4D, 0xD7, 0x62, 0xA6, 0xF1, 0x08, 0x18, 0x28, 0x78, 0x88,
0x83, 0x9E, 0xB9, 0xD0, 0x6B, 0xBD, 0xDC, 0x7F, 0x81, 0x98, 0xB3, 0xCE, 0x49, 0xDB, 0x76, 0x9A,
0xB5, 0xC4, 0x57, 0xF9, 0x10, 0x30, 0x50, 0xF0, 0x0B, 0x1D, 0x27, 0x69, 0xBB, 0xD6, 0x61, 0xA3,
0xFE, 0x19, 0x2B, 0x7D, 0x87, 0x92, 0xAD, 0xEC, 0x2F, 0x71, 0x93, 0xAE, 0xE9, 0x20, 0x60, 0xA0,
0xFB, 0x16, 0x3A, 0x4E, 0xD2, 0x6D, 0xB7, 0xC2, 0x5D, 0xE7, 0x32, 0x56, 0xFA, 0x15, 0x3F, 0x41,
0xC3, 0x5E, 0xE2, 0x3D, 0x47, 0xC9, 0x40, 0xC0, 0x5B, 0xED, 0x2C, 0x74, 0x9C, 0xBF, 0xDA, 0x75,
0x9F, 0xBA, 0xD5, 0x64, 0xAC, 0xEF, 0x2A, 0x7E, 0x82, 0x9D, 0xBC, 0xDF, 0x7A, 0x8E, 0x89, 0x80,
0x9B, 0xB6, 0xC1, 0x58, 0xE8, 0x23, 0x65, 0xAF, 0xEA, 0x25, 0x6F, 0xB1, 0xC8, 0x43, 0xC5, 0x54,
0xFC, 0x1F, 0x21, 0x63, 0xA5, 0xF4, 0x07, 0x09, 0x1B, 0x2D, 0x77, 0x99, 0xB0, 0xCB, 0x46, 0xCA,
0x45, 0xCF, 0x4A, 0xDE, 0x79, 0x8B, 0x86, 0x91, 0xA8, 0xE3, 0x3E, 0x42, 0xC6, 0x51, 0xF3, 0x0E,
0x12, 0x36, 0x5A, 0xEE, 0x29, 0x7B, 0x8D, 0x8C, 0x8F, 0x8A, 0x85, 0x94, 0xA7, 0xF2, 0x0D,
0x17,
0x39, 0x4B, 0xDD, 0x7C, 0x84, 0x97, 0xA2, 0xFD, 0x1C, 0x24, 0x6C, 0xB4, 0xC7, 0x52, 0xF6, 0x01
};
```

**Добавление циклового ключа**

В данной операции цикловой ключ добавляется к состоянию посредством простого EXOR. Цикловой ключ вырабатывается из ключа шифрования посредством алгоритма выработки ключей (key schedule).

Длина циклового ключа равна длине блока Nb.

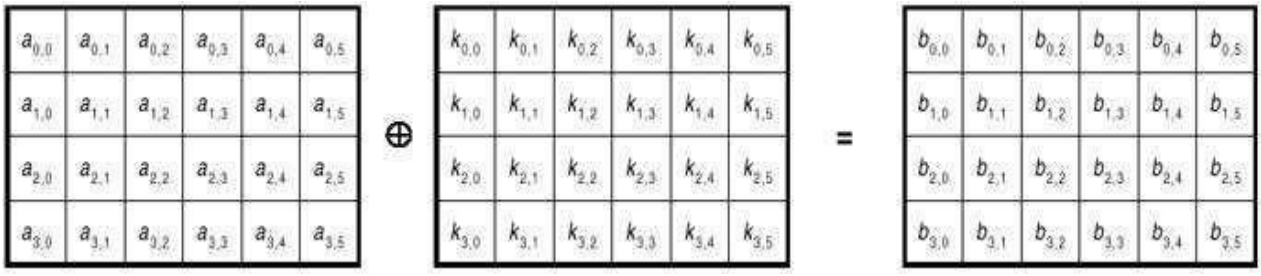



Рис. 4. 5. При добавлении ключа цикловой ключ складывается посредством EXOR с состоянием

При шифровании  части расширенного ключа выбираются от начала к концу, при расшифровании – от конца к началу.

**Расширение ключа (Key Expansion)**

Расширенный ключ представляет собой линейный массив 4-х байтовых слов и обозначен как W[Nb*(Nr+1)]. Первые Nk слов содержат ключ шифрования. Все остальные слова определяются рекурсивно из слов с меньшими индексами. Алгоритм выработки ключей зависит от величины Nk: ниже приведена версия для Nk равного или меньшего  6 и версия для Nk большего 6.

Для Nk<6 или Nk=6 мы имеем:

```
KeyExpansion(CipherKey,W)
{
        for (i = 0; i < Nk; i++) W[i] = CipherKey[i];
       for (j = Nk;j < Nb*(Nk+1);j+=Nk)
{
W[j] = W[j-Nk] ^ SubByte( Rotl( W[j-1] ) ) ^ Rcon[j/Nk];

for (i = 1;i < Nk&& i+j < Nb*(Nr+1);i++)
W[i+j] = W[i+j-Nk] ^ W[i+j-1];
}
}
```

Как можно заметить, первые Nk слов заполняются ключом шифрования. Каждое последующее слово W[i] получается посредством EXOR предыдущего слова W[i-1] и слова на Nk позиций ранее W[i-Nk]. Для слов, позиция которых кратна Nk, перед EXOR применяется преобразование к W[i-1], а затем еще прибавляется цикловая константа. Преобразование содержит циклический сдвиг байтов в слове, обозначенный как Rotl, затем следует SubByte - применение замены байт.

Для Nk>6 мы имеем:

```
KeyExpansion(CipherKey,W)
{
for (i=0; i<Nk; i++) W[i]=CipherKey[i];
for (j=Nk; j<Nb*(Nk+1); j+=Nk)
{
W[j] = W[j-Nk] ^ SubByte(Rotl(W[j-1])) ^ Rcon[j/Nk]; for (i=1; i<4; i++) W[i+j] = W[i+j-
Nk] ^ W[i+j-1]; W[j+4] = W[j+4-Nk] ^ SubByte(W[j+3]);
```

```
for (i=5; i<Nk; i++) W[i+j] = W[i+j-Nk] ^ W[i+j-1];
}
}
```

Отличие для схемы при Nk>6 состоит в применении SubByte для каждого 4-го байта из Nk.

Цикловая константа независит от Nk и определяется следующим образом:

```
Rcon[i] = ( RC[i], '00' , '00' , '00' ), где
RC[0]='01'
RC[i]=xtime(Rcon[i-1])
```

**Шифрование**

Шифр Rijndael включает следующие преобразования:

- начальное добавление циклового ключа;
- Nr-1 циклов;
- заключительный цикл.

На псевдо-Си это выглядит следующим образом:

```
Rijndael (State, CipherKey)
{
KeyExpansion(CipherKey, ExpandedKey); // Расширение ключа
AddRoundKey(State, ExpandedKey); // Добавление циклового ключа
For ( i=1 ; i<Nr ; i++) Round(State,ExpandedKey+Nb*i); // циклы
FinalRound(State, ExpandedKey+Nb*Nr); // заключительный цикл
}
```

Если предварительно выполнена процедура расширения ключа, то процедура будет выглядеть следующим образом:

```
Rijndael (State, CipherKey)
{
AddRoundKey(State, ExpandedKey);
For ( i=1 ; i<Nr ; i++) Round(State,ExpandedKey+Nb*i); FinalRound(State, ExpandedKey+Nb*Nr);
}
```

**Описание демонстрационной программы** Программа выполнена на языке C# и состоит из двух элементов – файла Rijndael.dll, содержащего реализацию алгоритма шифрования, и демонстрационного приложения RijndaelDemo.exe. Для работы приложения необходима ОС Windows с установленным .NET Framework v1.1. В основном окне демонстрационной программы задаются длина ключа, длина блока, можно посмотреть расширенный ключ шифрования, вычисляемый в соответствии с заданным ключом шифрования.

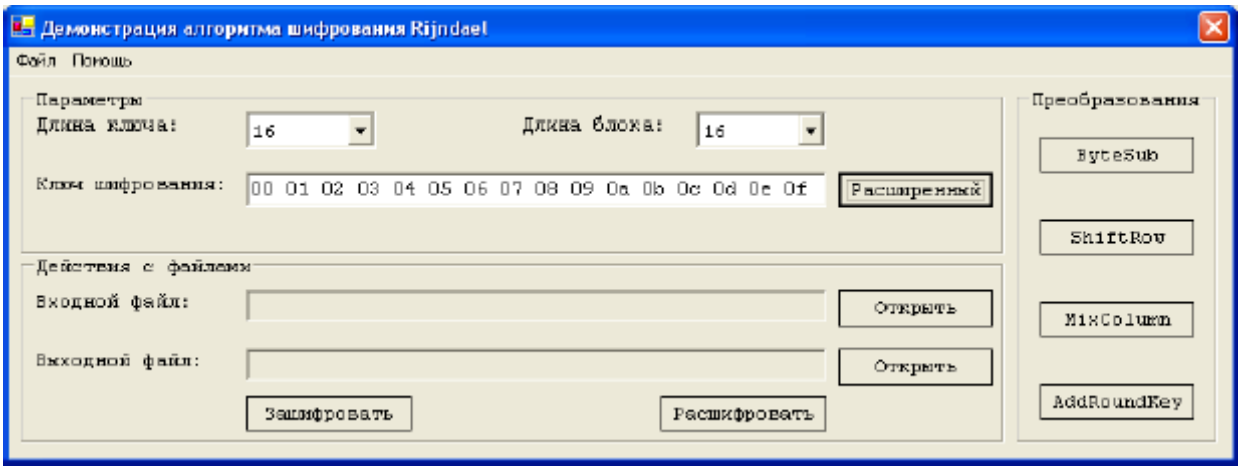



Рисунок 2. Главное окно демонстрационной программы

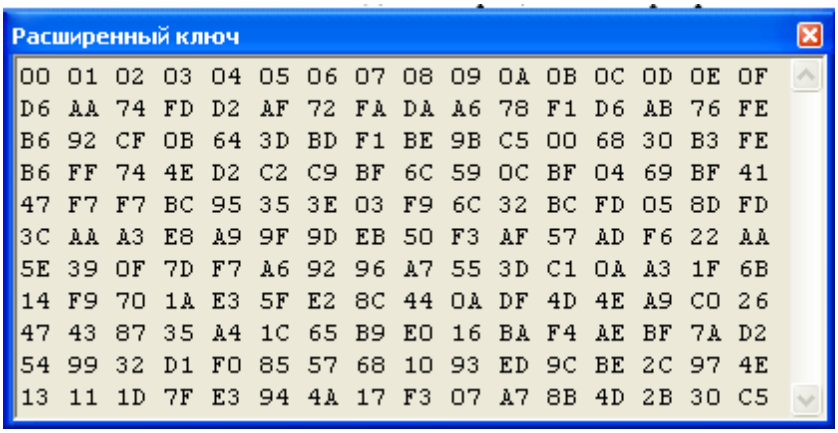



Рисунок 3. Окно расширенного ключа

Также есть возможность подробно рассмотреть действие всех цикловых преобразований (ByteSub, ShiftRow, MixColumn, AddRoundKey) как при шифровании, так и при расшифровании, рис.4, 5.

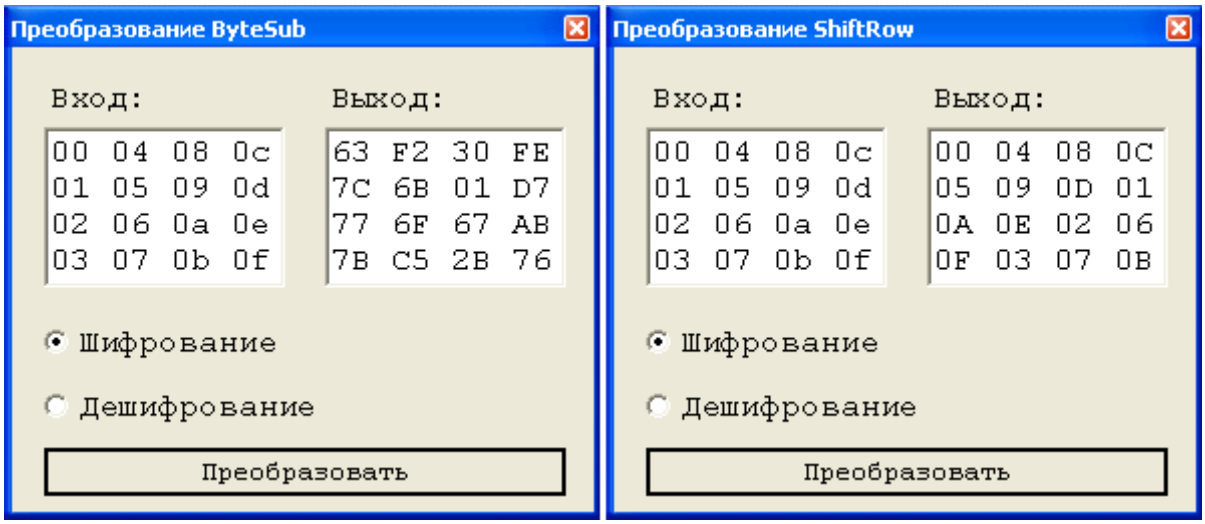
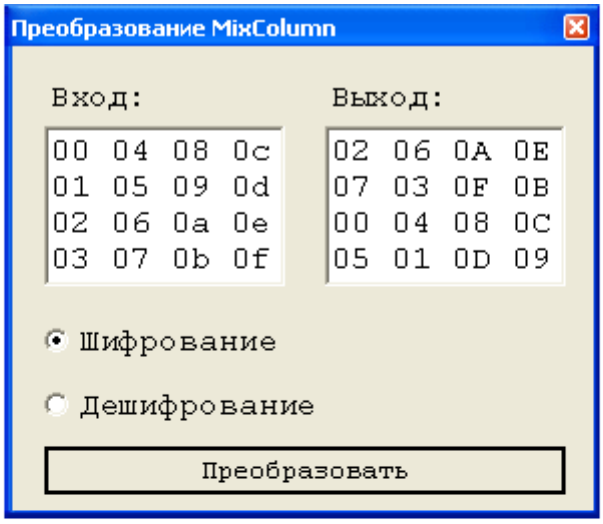



Рисунок 4. Окна преобразований ByteSub и ShiftRow



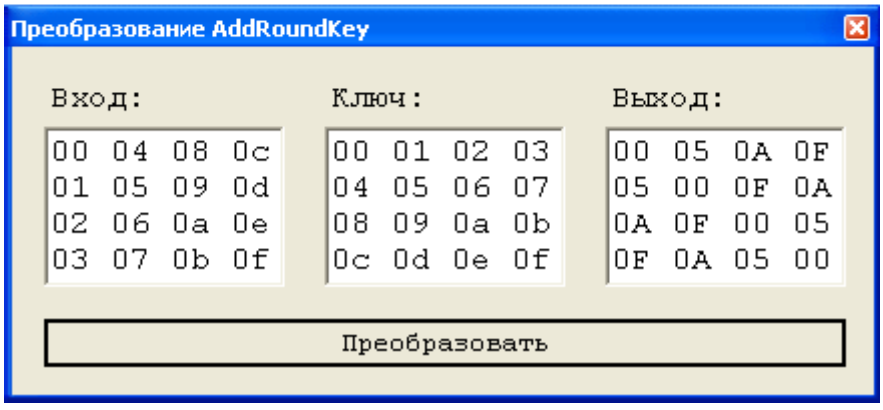




Рисунок 5. Окна преобразований MixColumn и AddRoundKey

При шифровании предлагается выбрать исходный файл и файл, куда будет помещен результат шифрования, при расшифровании соответственно зашифрованный файл и файл, предназначенный для помещения резуль-

тата расшифрования. В процессе используются, указанные в главном окне программы, ключ шифрования и длины ключа и блока.

**4. Оборудование и материалы**

Для выполнения работы потребуется:

- класс персональных ЭВМ не ниже Pentium-4, классная доска, мел;

- операционная система Microsoft Windows, программное средство ***RijndaelDemo*** ,

**5. Указания по технике безопасности**

Требования безопасности перед началом работ:

– ЗАПРЕЩАЕТСЯ: переодеваться, пользоваться огнем, курить, принимать пищу в лаборатории;

– убедиться в целостности электрических розеток и разъемов. Запрещается работать с неисправным электроинструментом, разбивать и самостоятельно ремонтировать его;

– в лаборатории необходимо быть в сменной обуви;

– включение компьютера производить только после получения допуска по выполняемой работе и разрешения преподавателя или лаборанта.

Требования безопасности во время работы:

– выполняя лабораторную работу, студенты обязаны использовать только вычислительную технику, периферийное оборудование, соединительные кабели, измерительное оборудование и дискеты, непосредственно относящиеся к данному лабораторному занятию;

– подключение и отсоединение составляющих вычислительного комплекса производить только при полном снятии напряжения со всех составляющих вычислительного комплекса.

Требования безопасности по окончании работы:

– по команде преподавателя студенты обязаны выключить компьютер и составляющие вычислительного комплекса;

– рабочее место привести в порядок.

**6. Задания**

**Задание №1.** Ознакомиться со сведениями о программе ***RijndaelDemo*** , изложенными в разделе 3. Запустить модуль ***RijndaelDemo.exe***.

**Задание №2.** Изучить на примере обычных текстовых файлов способы шифрования и расшифрования с помощью алгоритма *Rijndael*. Подробно рассмотреть действие всех цикловых преобразований (ByteSub, ShiftRow, MixColumn, AddRoundKey), как при шифровании, так и расшифровании. Исходный текст для шифрования может быть подготовлен заранее и сохранен в файле **.txt*.

**Задание №3.** Сохранить в отчете экранные формы, демонстрирующие процесс шифрования и расшифрования информации, проанализировать полученные результаты.

**Задание №4.** Включить в отчет о лабораторной работе ответы на контрольные вопросы, выбранные в соответствии с номером варианта, указанным преподавателем.

**7. Содержание отчета**

Отчет должен содержать:

1. Наименование работы, цель.

2. Последовательность действий при выполнении каждого задания.

3. Выводы по выполнению лабораторной работы.

## 8. Контрольные вопросы

| Номер варианта | Контрольные вопросы |
|---|---|
| 1,5,7,26 | Сравните основные характеристики алгоритмов *Rijndael* и ГОСТ 28147-89. |
| 2,4,6 | Сравните основные характеристики алгоритмов *Rijndael* и *DES*. |
| 11,13 | Опишите структуру сети Фейстеля. |
| 12,14,16 | Приведите обобщенные схемы шифрования данных с помощью алгоритма *Rijndael* и ГОСТ 28147-89. Дайте их сравнительный анализ. |
| 3,9,18,29 | Сравните один раунд шифрования данных с помощью алгоритма *Rijndael* и ГОСТ 28147-89. |
| 20,22,24 | Сравните эквивалентность прямого и обратного преобразований в алгоритмах *Rijndael* и ГОСТ 28147-89. |
| 10,17,19 | Сравните выработку ключевой информации в алгоритмах *Rijndael* и ГОСТ 28147-89. |
| 21,23,25 | Сравните алгоримы *Rijndael* и ГОСТ 28147-89 по показателям диффузии. |
| 8, 28,27 | Сравните алгоримы *Rijndael* и ГОСТ 28147-89 по показателям стойкости. |
| 12,15,30 | Сравните алгоримы *Rijndael* и ГОСТ 28147-89 по показателям производительности и удобству реализации. |

## 9. Список литературы, рекомендованный к использованию по данной теме

1. Жук, А.П., Жук, Е.П., Лепешкин, О.М., Тимошкин, А.И. Защита информации: Учебное пособие для бакалавров [Текст] / А.П. Жук и др. – М.: ИНФРА-М, 2015. -392с.

2. Шаньгин, В.Ф. Защита компьютерной информации. Эффективные методы и средства [Текст] / В.Ф. Шаньгин – М.: ДМК Пресс, 2012. – 544 с.

# ПРАКТИЧЕСКАЯ РАБОТА №4
# ИЗУЧЕНИЕ УСТРОЙСТВА И ПРИНЦИПА РАБОТЫ ШИФРОВАЛЬНОЙ МАШИНЫ ЭНИГМА *(Enigma)*

**Цель работы:** Изучение принципов шифрования/расшифрования информации, используемых в шифровальной машине Энигма. Ознакомление с общими принципами действия шифровальной машины Энигма на примере эмулятора. Предварительно необходимо установить программу эмулятор *Enigma3S*.

## 1. Описание лабораторной работы

### Энигма, устройство и принцип действия

Рис. 3.1. Внешний вид шифровальной машины Энигма

Энигма (*Enigma*)[4] — портативная шифровальная машина, использовавшаяся для шифрования и расшифрования секретных сообщений. Более точно, Энигма — целое семейство электромеханических роторных машин, применявшихся с 20-х годов XX века.

[4] Эмуляторы Энигмы *http://www.attlabs.att.co.uk/andyc/enigma*

Энигма использовалась в коммерческих целях, а также в военных и государственных службах во многих странах мира, но наибольшее распространение получила в Германии во время второй мировой войны. Именно Энигма Вермахта *(Wehrmacht Enigma)* — немецкая военная модель — чаще всего является предметом изучения.

Хотя шифр Энигмы, с точки зрения криптографии, был достаточно слаб, но на практике лишь сочетание этого фактора с другими, такими, как ошибки оп

ераторов, процедурные изъяны и захваты экземпляров Энигмы

и шифровальных книг, позволило английским криптоаналитикам вскрывать сообщения, зашифрованные шифром Энигмы.

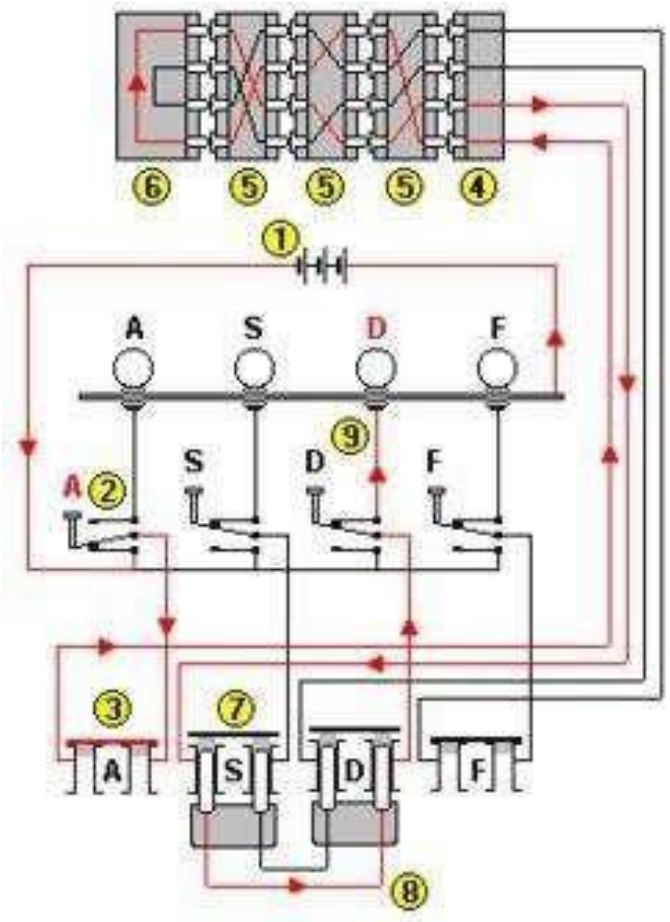



Рис. 3.2. Электрическая схема Энигмы (замена в тексте буквы 'A' буквой 'D')

Шифрующее действие Энигмы показано для двух последовательно нажатых клавиш — ток течет через роторы, "отражается" от рефлектора, затем снова возвращается через роторы. Серыми линиями на рисунке показаны другие возможные электрические цепи внутри каждого ротора. Буква “A” заменяется в шифротексте по-разному при последовательных нажатиях клавиши, сначала на “G”, затем на “C”. Сигнал идет по другому маршруту за счет поворота ротора.

Как и другие роторные машины, Энигма состояла из комбинации механических и электрических систем. Механическая часть включала клавиатуру, набор вращающихся дисков (роторов), расположенных вдоль вала, и ступенчатого механизма, приводящего в движение один или более роторов при каждом нажатии клавиши. Движение роторов приводит к различным вариантам подстановки символов при каждом следующем нажатии клавиши на клавиатуре.

Механические части двигались, образуя меняющийся электрический контур — то есть, фактически, шифрование осуществлялось электрически. При нажатии клавиш контур замыкался, ток проходил через различные компоненты и в итоге включал одну из множества лампочек, отображавшую выводимую букву. Например, при шифровании сообщения, начинающегося “ANX...”, оператор вначале нажимал кнопку “A”, и загоралась лампочка “Z”, то есть “Z” становилась первой буквой криптограммы. Оператор продолжал шифрование, нажимая на клавиатуре “N” и, так далее, до конца исходного сообщения.

Постоянное изменение электрической цепи, через которую шел ток, вследствие вращения роторов, позволяло реализовать многоалфавитный шифр подстановки, что давало высокую стойкость для того времени.

**Роторы**

Рис. 3.3. Левая сторона ротора Энигмы, видны плоские электрические контакты

Рис. 3.4. Правая сторона ротора, видны штыревые контакты

Роторы — это сердце Энигмы. Каждый ротор представляет собой диск примерно 10 см в диаметре, сделанный из твердой резины или бакелита, с пружинными штыревыми контактами на одной стороне ротора, расположенными по окружности; на другой стороне соответствующее количество плоских электрических контактов. Штыревые и плоские контакты соответсвуют буквам в алфавите; обычно это 26 букв “A”…“Z”. При соприкосновении контакты соседних роторов замыкают электрическую цепь. Внутри ротора каждый штыревой контакт соединен с некоторым плоским. Порядок соединения может быть различным.

Сам по себе ротор воспроизводит шифрование простой заменой символов. Например, контакт, отвечающий за букву “E”, может быть соединен с контактом буквы “T” на другой стороне ротора. Но при использовании не-

скольких роторов в связке (обычно трех или четырех), за счет их постоянного движения, получается более стойкий тип многоалфавитного шифрования.

Ротор может занимать одну из 26 позиций в машине. Он может быть повернут вручную при помощи рифленого пальцевого колесика, которое выдается наружу, как показано на Рис. 3.5. Чтобы оператор всегда мог определить положение ротора, на каждом ободе находится алфавитное кольцо; одна из букв видна через окошко. В ранних моделях Энигмы алфавитное кольцо было фиксировано; в более поздних версиях ввели усложненную конструкцию с возможностью его регулировки. Каждый ротор содержит выемку (или несколько выемок), используемых для управления движением роторов.

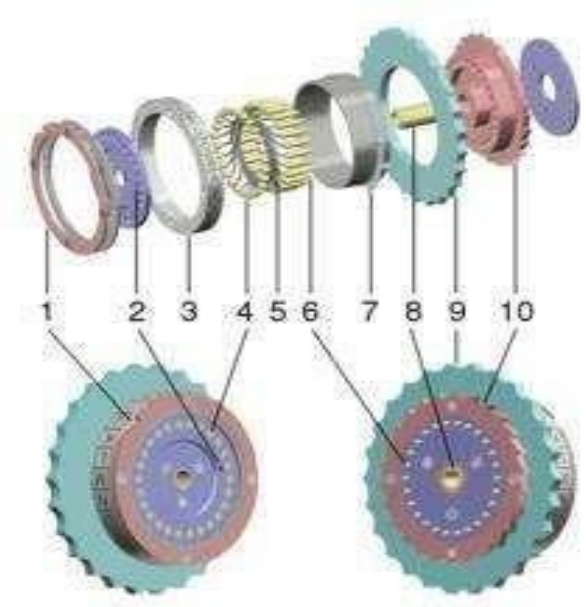



Рис. 3.5. Ротор в разобранном виде
1. кольцо с выемками
2. маркирующая точка для контакта "A"
3. алфавитное кольцо
4. залуженные контакты
5. электропроводка
6. штыревые контакты
7. пружинный рычаг для настройки кольца
8. втулка
9. пальцевое кольцо
10. храповое колесо

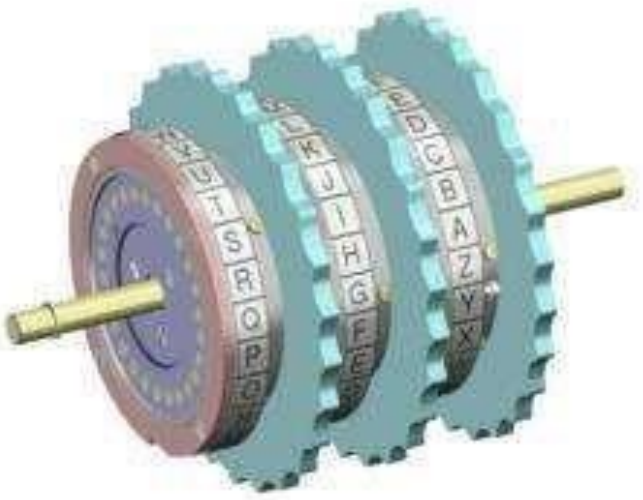


Рис. 3.6. Три последовательно соединенных ротора

Военные версии Энигмы выпускались с несколькими роторами; первая модель содержала только три. В 1938 г. их стало пять, но только три из них одновременно использовались в машине. Эти типы роторов были маркированы римскими цифрами I, II, III, IV, V, и все с одной выемкой, расположенной в разных местах алфавитного кольца. В военно-морских версиях *Wehrmacht Enigma* содержалось большее количество роторов, чем в других: шесть, семь или восемь.

В *Wehrmacht Enigma* каждый ротор прикреплен к регулируемому кольцу с выемками. Пять базовых роторов (I-V) имели по одной выемке,

тогда как военно-морские с дополнительными роторами (VI-VIII) — по две. В определенный момент, выемка попадает напротив собачки, позволяя ей зацепить храповик следующего ротора при последующем нажатии клавиши. Когда же собачка не попадает в выемку, она просто проскальзывает по поверхности кольца, не цепляя шестеренки. В системе с одной выемкой второй ротор продвигается вперед на одну позицию в то время, как первый продвигается на 26. Аналогично, третий ротор продвигается на один шаг в то время, как второй делает 26 шагов. Особенностью было то, что второй ротор также поворачивался, если поворачивался третий; это означает, что второй ротор мог повернуться дважды при двух последовательных нажатиях клавиш, так называемое "двухшаговое движение", что приводит к уменьшению периода при шифровании.

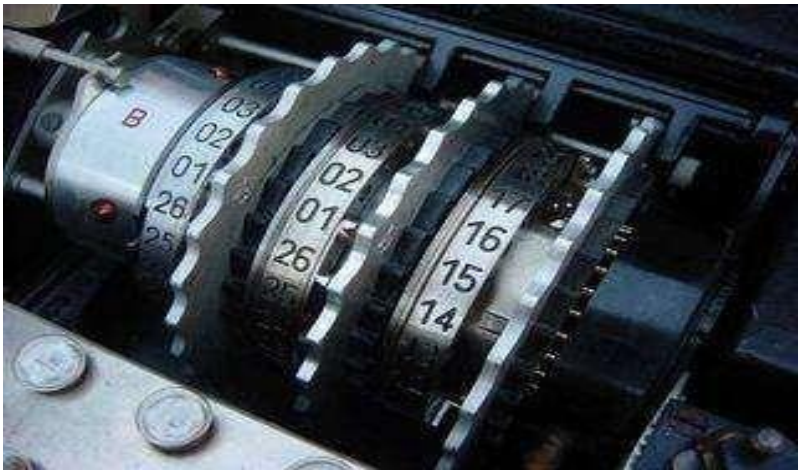



Рис. 3.7. Роторы Энигмы в собранном состоянии. Три подвижных ротора помещены между двумя неподвижными деталями: входное кольцо и рефлектор (помечен "B" слева)

**Входное колесо**

Входное колесо (*Eintrittswalze* по-немецки), или входной статор, соединяет коммутационную панель, или (в случае ее отсутствия) клавиатуру и ламповую панель, с роторами. Несмотря на то, что фиксированное соединение проводов имеет сравнительно небольшее значение с точки зрения безопасности, это оказалось некоторым препятствием в работе польского криптоаналитика Марьяна Реджевски, когда он пытался определить способ коммутации проводов внутри роторов. Коммерческая версия Энигмы соединяла буквы в порядке их следования на клавиатуре: QA, WB, EC и так далее. Однако, военная версия соединяла их в прямом алфавитном порядке: AA, BB, CC и т.д.

**Рефлектор**

За исключением ранних моделей A и B, за последним ротором следовал рефлектор (Umkehrwalze по-немецки), запатентованная деталь, отличавшая семейство Энигмы от других роторных машин, разработанных в то время. Рефлектор соединяет контакты последнего ротора попарно, коммутируя ток через роторы в обратном направлении, но по другому маршруту. Рефлектор гарантирует, что преобразование, реализуемое Энигмой, есть инволю-

ция, т.е. процесс расшифрования симметричен процессу шифрования. Кроме того, рефлектор придает Энигме то свойство, что никакая буква не может быть зашифрована собой же. Это

было серьезным концептуальным недостатком, впоследствии использованным дешифровальщиками.

**Коммутационная панель**


Рис. 3.8. Коммутационная панель в передней части машины. Могло использоваться до 13 соединений. На фотографии две пары букв переключены (“S”-“O” и “J”-“A”)

Коммутационная панель, позволяющая оператору варьировать соединения проводов, впервые появилась в немецких военных версиях в 1930 г. и вскоре успешно использовалась и в военно-морских версиях. Коммутационная панель внесла огромный вклад в усложнение шифра Энигмы, даже больший, чем введение дополнительного ротора. С Энигмой без коммутационной панели — *"unsteckered"* Энигмой — дешифровальщик может справиться практически вручную, однако эти методы оказывались беспомощными при добавлении коммутационной панели, и взломщики были вынуждены конструировать специальные дешифровальные машины. Кабель, помещенный на коммутационную панель, соединяет буквы попарно, например, “E” и “Q” могут быть соединены в пару. Эффект состоит в перестановке этих букв до и после прохождения сигнала через роторы. Например, когда оператор нажимал “E”, сигнал направлялся в “Q”, и только после этого уже во входной ротор. Одновременно могло использоваться несколько таких пар (до 13).

**Аксессуары**

Удобной деталью, использовавшейся на M4 Энигма, был *"Schreibmax",* маленькое печатающие устройство, которое могло печатать все 26 букв на небольшом листе бумаги.

**Процедуры использования Энигмы**

Во время второй мировой войны немецкие операторы использовали шифровальную книгу только для установки роторов и настроек колец. Для каждого сообщения, они выбирали случайную стартовую позицию, допустим “WZA”, и случайный ключ сообщения, допустим “SXT”. Далее оператор устанавливал роторы в стартовую позицию “WZA”, и шифровал ключ сообщения “SXT”. Предположим, что в результате шифрования ключа получится “UHL”. Далее оператор ставил ключ сообщения “SXT”, как начальную позицию роторов, и шифровал сообщение. Затем оператор отправлял стартовую позицию “WZA” и зашифрованный ключ “UHL” вместе с сообщением. Получатель устанавливал стартовую позицию в соответствии с первой трехграммой “WZA” и, расшифровывая вторую триграмму “UHL” Привести в отчете ответы на контрольные вопросы, в соответствии с номером варианта, указанным преподавателем.

Распознавал исходный ключ “SXT”. Далее, получатель использовал этот ключ как стартовую позицию для расшифровки сообщения. Обычно срок действия ключей составлял один день.

Военная версия Энигмы использовала только 26 букв. Прочие символы заменялись редкими комбинациями букв. Пробел пропускался либо заменялся “X”. Символ “X” также использовался для обозначения точки либо конца сообщения. Некоторые особые символы использовались

в отдельных вооруженных частях, например, *Wehrmacht* заменяла запятую двумя символами “ZZ” и вопросительный знак - “FRAGE” либо “FRAQ”.

А *Kriegsmarine* заменяла запятую - “Y” и вопросительный знак - “UD”. Два, три или четыре нуля заменялись CENTA, MILLE и MYRIA соответственно.

При выполнении лабораторной работы для исследования шифра Энигмы используется программа-эмулятор ***Enigma3S.***

**2. Задание**

1. Запустите эмулятор Энигмы ***Enigma3S*** из папки, указанной преподавателем. Ознакомьтесь с файлом справки: опция меню

Help=>Help. В меню программы выберите пункт Settings => Reset, рис.3.9.

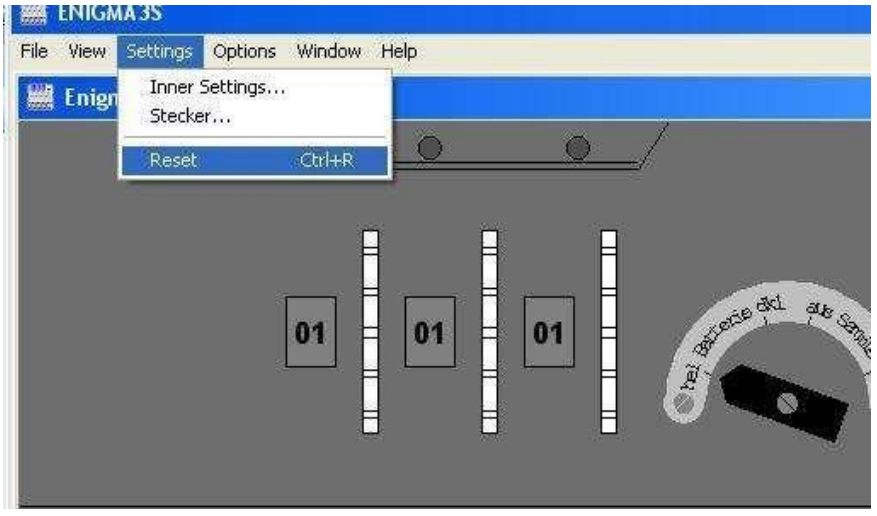



Рис. 3.9. Начальные установки эмулятора Энигмы

3. Установите значения для колец 01 01 01 путем выбора пункта меню

View=>Open Cover. В меню Settings=>Inner Settings установите следующие значения Reflector – B, Left – I, Middle – II, Right – III, Ringstellung – A-A-A (Будем считать данное положение начальным),

рис.3.10, 3.11.

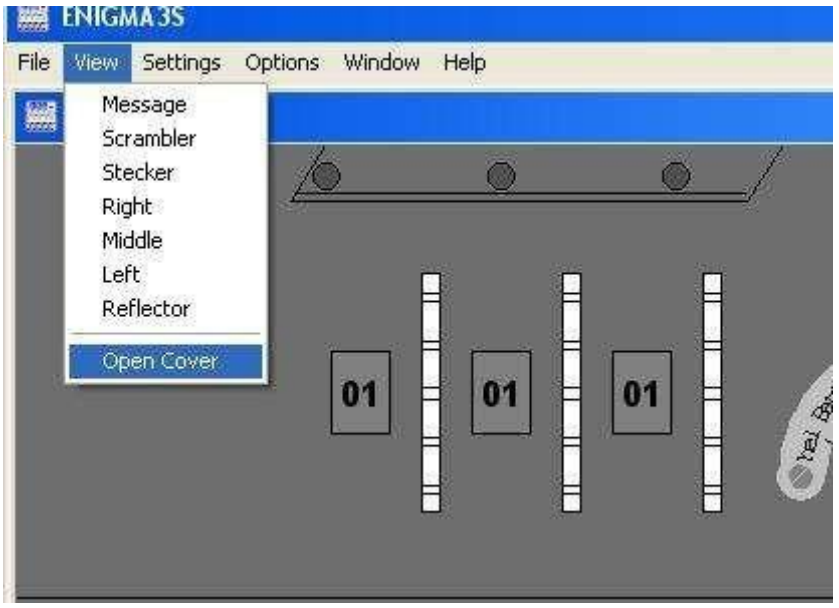



Рис. 3.10. Просмотр начальных установок

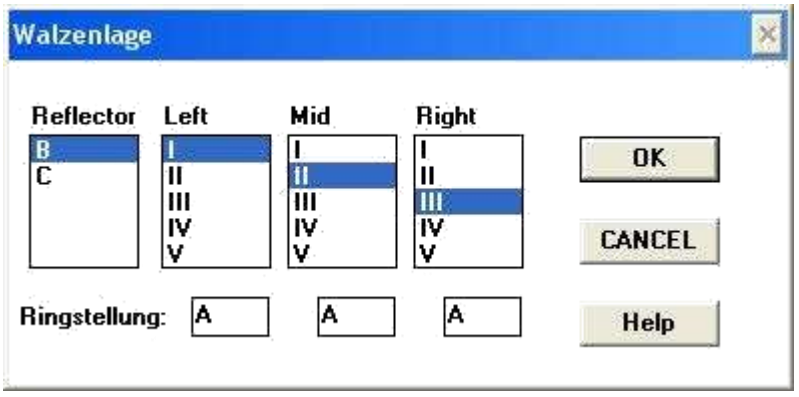



Рис. 3.11. Установки Reflector, Left, Mid, Right

4. Введите на клавиатуре Энигмы сообщение “SECRET MESSAGE”. Какое сообщение получено на выходе?

5. Повтроите пункт 3, изменив настройки Ringstellung – A-A-A на Ringstellung – A-B-C

6. Введите на клавиатуре Энигмы сообщение “SECRET MESSAGE”. Какое сообщение получено на выходе? Насколько оно отличается от сообщения полученного в пункте 4?

7. Сохраните полученный в пункте 6 шифротекст при помощи опции меню File=>Save CT as, рис. 3.12.

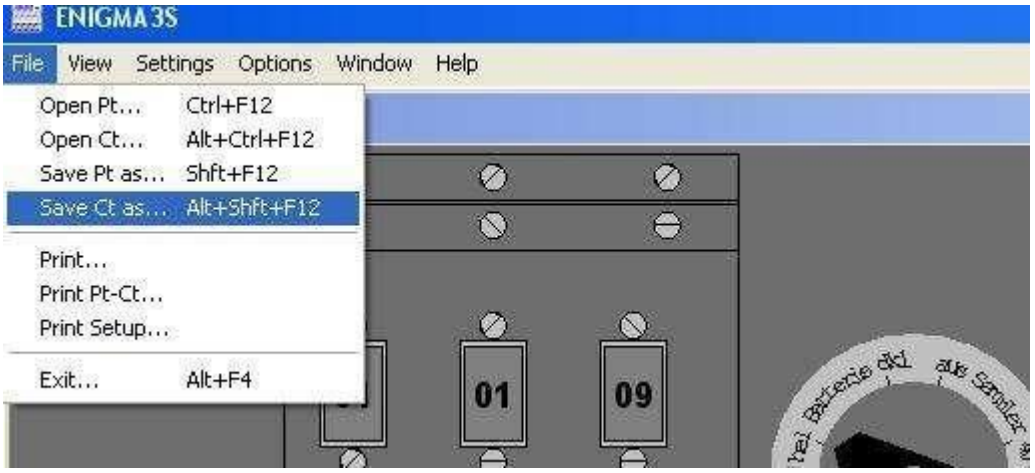



Рис. 3.12. Сохранение шифротекста

8. Создайте в корне папки с программой эмулятором файл с расширением .pln, откройте его для редактирования в блокноте и запишите в него открытый текст для шифрования.

9. В меню программы выберите опцию File=>Open Pt и выберите файл, созданный в пункте 9. Получите шифротекст из открытого текста, выбрав опцию меню Options=>Encipher Text.

10. Установите эмулятор в начальное положение. В меню программы выберите пункт Window=> Scrambler. Введите при помощи клавиатуры произвольное сообщение из 22 символов (варианты указаны в *таблице 3.1*), обращая внимание на положение колец. Введите последовательность из 22 символов еще раз. Как изменилось положение колец? Отличается ли новая зашифрованная последовательность от начальной? Почему изменилась выходная последовательность?

11. Используя окно Scrambler проследите за тем как изменяется шифротекст в зависимости от настройки положения контактных колес. Что дает возможность настройки порядка следования контактных колес?

В окне программы выберите опцию View=>Right, рис. 3.13.

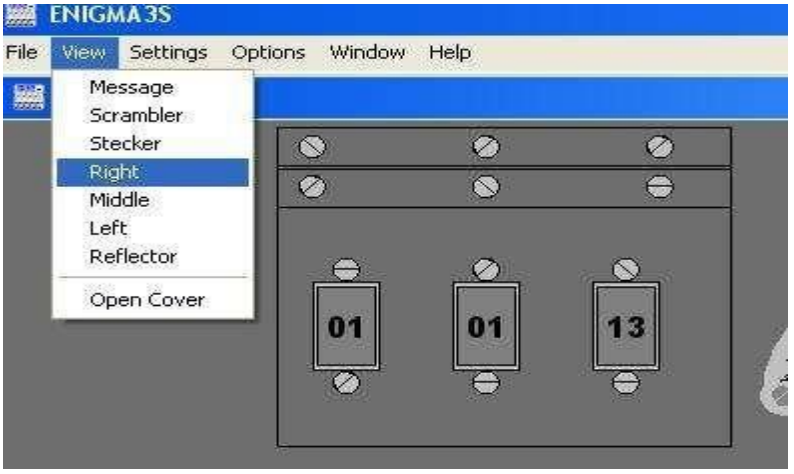


Рис. 3.13. Просмотр установок для выполнения задания

Заполните таблицу соответствия для 5-ти букв варианты указаны в таблице 3.1) для 12 первых угловых положений правого колеса.

**Таблица 3.1**

| **Буквына входе** | **Угловые положения колеса** | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 11 | 22 | 33 | 44 | 55 | 66 | 77 | 88 | 99 | 110 | 111 | 112 |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |
| | | | | | | | | | | | | |

13. Зная, что эмулятор установлен в начальное положение расшифруйте текст в соответствии с номером варианта, указанным преподавателем (варианты контрольных заданий указаны в *таблицах 3.2,3.3,3.4*).

Таблица 3.2

| **Номера вариантов** | **Исходный алфавит** |
|---|---|
| 1,5,9,13,17 | QRTYU FBNAK GHERL ADLKE DS |
| 2,6,10,14,18,22 | UIERT PAEVC DSNCY OPLKD BV |
| 3,7,11,15,19,23,27 | UIFGH KLBVQ FDIIT QKJLS DB |
| 4,8,12,16,20,24,28 | LLWER TYYWV BAFDP WRTOPF JK |
| 21,25,29,26,30 | OPJHG JFDPJ GFDSK LDFHU BX |

Таблица 3.3

| **Номера вариантов** | **Шифротекст** |
|---|---|
| 1,5,9,13,17 | ABCDE |
| 2,6,10,14,18,22 | FGHIJK |
| 3,7,11,15,19,23,27 | LMNOP |
| 4,8,12,16,20,24,28 | QRSTU |
| 21,25,29,26,30 | VWXYZ |

Таблица 3.4

| **Номера вариантов** | **Шифротекст** |
|---|---|
| 1,5,9,13,17 | FQGAH WABUN NL |
| 2,6,10,14,18,22 | QIKOL RCRJS EGBSS X |
| 3,7,11,15,19,23,27 | OOKWE PRFMI M |
| 4,8,12,16,20,24,28 | KIXDI ACTHJ L |
| 21,25,29,26,30 | XLXOO EABUN NL |

## Практическая работа №5

## Генерация простых чисел, используемых в асимметричных системах шифрования

**Цель работы**: исследования генерации простых чисел, используемых в системах шифрования с открытым ключом и умений делать самостоятельные выводы по результатам исследований.

### 2. Формируемые компетенции

Данная Практическая работа направлена на формирование следующих компетенций выпускника бакалавриата:

ОПК-1: способностью инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем;

ПК-5: способностью сопрягать аппаратные и программные средства в составе информационных и автоматизированных систем.

### 3. Теоретическая часть

### Асимметричные системы шифрования

*Асимметричные криптосистемы* (*системы открытого шифрования, с открытым ключом - public key systems*) – смысл данных криптосистем состоит в том, что для зашифрования и расшифрования используются разные преобразования. Одно из них – зашифрование – является абсолютно открытым для всех. Другое же – расшифрование – остается секретным за счет секретности ключа расшифрования. Таким образом, любой, кто хочет что-либо зашифровать, пользуется открытым преобразованием. Но расшифровать и прочитать это сможет лишь тот, кто владеет секретным ключом. Схема асимметричной криптосистемы представлена на рис.1.1.

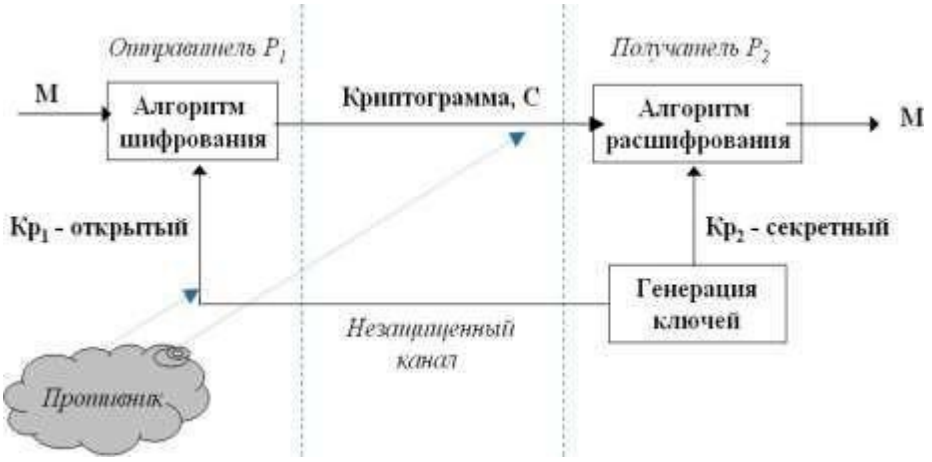



Рис. 1.1. Обобщенная схема асимметричной криптосистемы

В настоящий момент во многих асимметричных криптосистемах вид преобразования определяется ключом. У пользователя есть два ключа – секретный и открытый. Открытый ключ публикуется в общедоступном месте, и каждый, кто захочет послать сообщение этому пользователю – зашифровывает текст открытым ключом. Расшифровать сможет только упомянутый пользователь с секретным ключом. Таким образом, отпадает проблема передачи секретного ключа, как в симметричных системах. Однако, несмотря на все свои преимущества, эти криптосистемы достаточно трудоемки и медлительны. Стойкость асимметричных криптосистем базируется, в основном, на алгоритмической трудности решить за приемлемое время какую-либо задачу. Если злоумышленнику удастся построить такой алгоритм, то дискредитирована будет вся система и все сообщения, зашифрованные с помощью этой системы. В этом состоит главная опасность асимметричных криптосистем в отличие от симметричных.

**Алгоритм Диффи-Хеллмана**

Алгоритм Диффи-Хеллмана *(Diffie-Hellman)* использует функцию дискретного возведения в степень. Сначала генерируются два больших простых числа $n$ и $q$. Эти два числа не обязательно хранить в секрете. Далее один из

партнеров $P_1$ генерирует случайное число $x$ и посылает другому участнику будущих обменов $P_2$ значение

$$A = q^x \bmod n$$

По получении $A$ партнер $P_2$ генерирует случайное число $y$ и посылает участнику обмена $P_1$ вычисленное значение

$$B = q^y \bmod n$$

Партнер $P_1$, получив $B$, вычисляет $K_x = B^x \bmod n$, а партнер $P_2$ вычисляет $K_y = A^y \bmod n$. Алгоритм гарантирует, что числа $K_y$ и $K_x$ равны и могут быть использованы в качестве секретного ключа для шифрования. Ведь даже перехватив числа $A$ и $B$, трудно вычислить $K_x$ или $K_y$. Схематично, работа алгоритма Диффи-Хеллмана представлена на рис. 1.2.

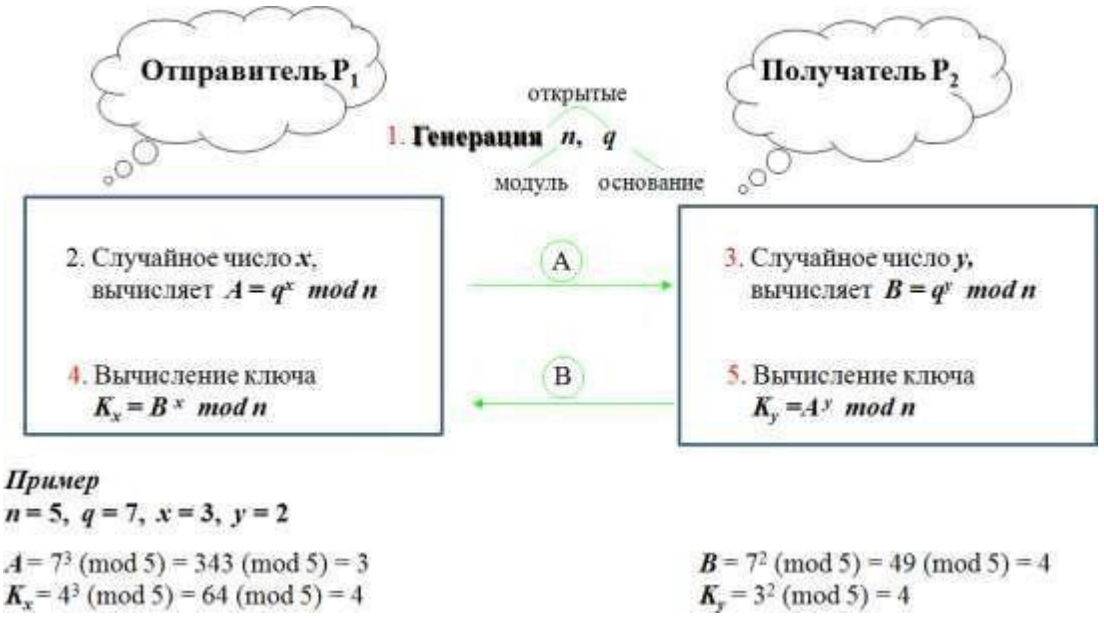



Рис. 1.2. Алгоритм Диффи-Хеллмана

**Алгоритм RSA**

Первое практическое воплощение принцип открытого шифрования получил в системе *RSA,* разработанной в 1977 году в Массачусетском Технологическом Институте (США) и получившей свое название от первых букв фамилий авторов: Рональд Ривест *(R.Rivest),* Эди Шамир *(A.Shamir),* Леонард Адлеман *(L.Adleman)*.

Идея авторов состояла в том, что взяв целое число *N* в виде произведения двух больших простых чисел *N=P*Q*, легко подобрать пару чисел *Y* и *X*, таких, чтобы для любого целого числа *M*, меньшего *N*, справедливо соотношение:

$$(M^X)^Y = M \bmod N$$

В качестве открытого ключа шифрования в системе *RSA* выступают ключ *Y* и модуль *N*, а секретным ключом для расшифрования сообщений является число *X*.

Процедура шифрования сообщения *M*, рассматриваемого как целое число (такое допущение возможно вследствие того, что любой контент может быть представлен в числовой форме при обработке в средствах вычислительной техники), меньшее *N* (при необходимости длинное сообщение разбивается на отрезки, шифруемые независимо), состоит в вычислении значения:

$$C = M^Y \bmod N$$

Расшифрование осуществляется аналогично с использованием секретного ключа *X:*

$$M = C^X \bmod N$$

Математически строго можно доказать, что определение по паре чисел (*N*, *Y)* секретного ключа *X*, не проще разложения на простые множители числа *N*, то есть нахождения *P* и *Q*. Задача же разложения на множители целого числа изучается в математике с древнейших времен и известна как сложная вычислительная задача. На настоящий момент разложение числа из нескольких сотен десятичных знаков потребует от современных вычислительных машин сотен лет непрерывной работы.

Схематично, работа алгоритма *RSA* представлена на рис. 1.3.

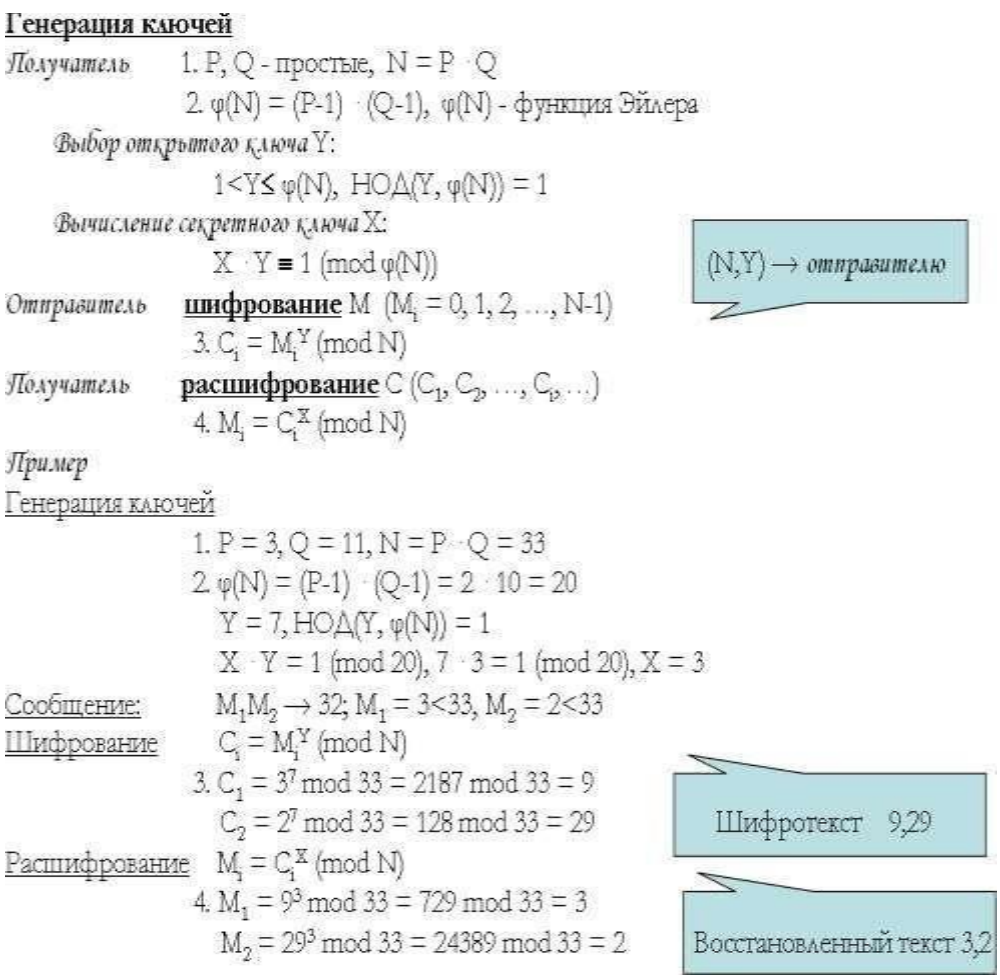



Рис. 1.3. Алгоритм RSA

### 1.1. Методы проверки чисел на простоту

Одна из главных проблем асимметричного шифрования – генерация больших простых чисел. Простейшим методом проверки простоты натурального числа $N$ является метод пробных делений: для $d$=2, 3, 4 … мы проверяем выполнение условия $(d, N)>1$ (здесь $(d, N)$ – наибольший общий делитель чисел $d$ и $N$). Число операций, требуемых для этого метода, имеет порядок корня из $N$.

Поэтому уже для чисел порядка $10^{30}$–$10^{40}$ он не применим.

В отличие от таких “детерминированных“ тестов существуют еще “вероятностные“ тесты проверки простоты. Для исследуемого числа прове-

ряется выполнение некоторых, связанных со случайными числами, условий. Если какое- либо из этих условий не выполнено, то $N$ – составное

число. Если же все условия выполнены, то с некоторой вероятностью можно утверждать, что $N$ – простое число. Эта вероятность тем ближе к 1, чем большее количество случайных чисел мы проверим.

Обычно эти условия основаны на малой теореме Ферма, утверждающей, что для любого положительного числа $b$, не превосходящего некоторого простого числа $p$ :

$b^{(p-1)} = 1(mod\ p)$.

Например, $2^6 = 64 = 63+1 = 1(mod\ 7)$. Если требуется определить, является ли целое число $r$ простым, то можно выбрать любое положительное целое число $b$, меньшее $r$, и проверить, выполнено ли равенство

$b^{(r-1)} = 1(mod\ r)$.

Если равенство не выполнено, то на основании теоремы Ферма можно быть совершенно уверенным, что $r$ – не простое число. Если же равенство выполнено, то можно лишь предполагать, что $r$ – простое число и поэтому назвать его “псевдопростым по основанию $b$“. Вероятность $P\ (x)$ того, что составное число $x$ окажется псевдопростым по случайному основанию, убывает с ростом $x$.

К сожалению, существуют так называемые числа Кармайкла – такие составные числа, которые обладают свойством:

$b^{(r-1)} = 1(mod\ r)$ для всех $b$ из интервала [1, $r$], которые взаимно просты с $r$.

Примером числа Кармайкла является число 561 = 3*11*17.

Классический результат теории чисел – теорема Чебышева – показывает, что доля положительных целых чисел, меньших некоторого целого *m* и являющихся простыми, близка к *1/(ln m)*. Например, доля целых чисел, меньших $10^{100}$ и являющихся простыми, близка к $1/(\ln 10^{100}) = 1/230$.

Таким образом, если мы выберем случайно большое целое положительное нечетное число *x* и будем последовательно проверять на простоту числа *x, x+1, x+2, …* , то, в среднем, мы впервые встретим простое число на шаге с номером *ln x.*

**4. Оборудование и материалы**

Для выполнения работы потребуется:

- класс персональных ЭВМ не ниже Pentium-4, классная доска, мел;

- операционная система Microsoft Windows, программное средство *L_PROST.EXE.*

**5. Указания по технике безопасности**

Требования безопасности перед началом работ:

– ЗАПРЕЩАЕТСЯ: переодеваться, пользоваться огнем, курить, принимать пищу в лаборатории;

– убедиться в целостности электрических розеток и разъемов. Запрещается работать с неисправным электроинструментом, разбивать и самостоятельно ремонтировать его;

– в лаборатории необходимо быть в сменной обуви;

– включение компьютера производить только после получения допуска по выполняемой работе и разрешения преподавателя или лаборанта.

Требования безопасности во время работы:

– выполняя лабораторную работу, студенты обязаны использовать только вычислительную технику, периферийное оборудование, соединительные кабели, измерительное оборудование и дискеты, непосредственно относящиеся к данному лабораторному занятию;

– подключение и отсоединение составляющих вычислительного комплекса производить только при полном снятии напряжения со всех составляющих вычислительного комплекса.

Требования безопасности по окончании работы:

– по команде преподавателя студенты обязаны выключить компьютер и составляющие вычислительного комплекса;

– рабочее место привести в порядок.

**Описание лабораторной работы**

Для выполнения лабораторной работы необходимо запустить программу *L_PROST.EXE.* На экране дисплея появляется окно, с размещенным в его центре текстовым редактором (для отображения информации об осуществленных программой действиях), в верхней строке окна расположено главное меню, чуть ниже основного меню расположена панель инструментов (для управления быстрыми командными кнопками и другими “горячими” элементами управления), а в самом низу окна, под тестовым редактором, находится строка состояния, в которой указывается подсказка и выводится дополнительная информация. Клавиши панели инструментов, для удобства, снабжены всплывающими подсказками.

Для того чтобы попасть в основное меню, необходимо нажать клавишу F10. Передвижение по главному меню осуществляется клавишами перемещения курсора. Чтобы вызвать пункт меню, нужно нажать клавишу “ENTER”, “ESC” – выход из основного меню.

Для удобства в программе предусмотрена работа с мышью. В этом случае указатель подводится к нужному пункту главного меню или к нужной кнопке на панели инструментов и нажимается левая клавиша мыши, для отказа достаточно нажать клавишу “ESC”.

Кроме основных функций главного меню (“Генерация простого P”, “Поиск в интервале”, “Проверка на простоту” и так далее) панель инструментов содержит клавишу, при нажатии которой выводится информация о программе.

**Генерация простого P**

Возможность генерации простого числа; количество разрядов генерируемого числа задается пользователем (от 1 до 5).

**Поиск в интервале**

Возможность поиска простых чисел в заданном интервале. Пользователем задается: начало интервала – значение x, затем длина интервала – значение L. Поиск будет осуществляться в интервале (x, x+L).

При проверке на простоту каждого числа интервала сначала выполняется тест пробных делений на первые по порядку простые числа, а затем проверка по тесту Ферма. Для задания способов проверки на простоту необходимо левой клавишей мыши отметить флажок рядом с названием нужного метода, а затем указать все необходимые данные для начала поиска.

В методе пробных делений исходные данные – количество первых простых чисел для деления, а в тесте Ферма надо указать количество оснований и их значения.

По окончании расчета на экран выведутся найденные простые числа и их количество. Полную картину результатов работы можно просмотреть в пункте меню “Вывод результатов“.

**Проверка на простоту**

Возможность проверки на простоту любого числа. Необходимо ввести число и параметры расчета аналогично поиску в интервале.

**Вывод результатов**

Возможность просмотра всех результатов последней обработки. При входе в программу, когда никаких расчетов еще не производилось,предоставляется возможность просмотра исходного файла первых простых чисел, используемых для теста пробных делений.

**Дополнительные сведения о программе**

1. При запуске утилит генерации простого числа, поиска в интервале и проверки на простоту у пользователя запрашивается подтверждение на правильность выбранного метода для работы.

2. Во время работы длительных по исполнению процедур запускается прогресс процесса и гасится окно текстового редактора. По полоске прогресса можно наблюдать и оценивать примерную скорость работы алгоритма и время окончания текущего процесса.

3. Будьте внимательны при установке параметров работы, так как в процессе вычисления по ходу работы эти параметры изменить уже не удастся.

4. Описание "горячих клавиш":

Ctrl+F1 - ‘Генерация простого P’

Ctrl+F2 - ‘Поиск в интервале’

Ctrl+F3 - ‘Проверка на простоту’

Ctrl+F4 - ‘Вывод результатов’

Ctrl+X - ‘Выход из программы’

5. В лабораторной работе из-за большого времени счета рекомендуется использовать числа не более пяти разрядов и длину интервала выбирать не более 500, количество оснований для теста Ферма – не более 5.

6. Для правильного функционирования программы, в рабочей директории (вместе с файлом *l_prost.exe*) обязательно должны находиться файлы *prost.txt* и *work.txt*. Не рекомендуется вносить какие-либо изменения в эти текстовые файлы иначе последствия могут быть непредсказуемые.

## 6. Задания

**Задание №1.**

1. Проверить на простоту два произвольных целых числа, разрядностью не менее 5.

2. Распределение простых чисел.

2.1. Задан интервал вида *[x, x+L]*. Вычислить количество *П(x, L)* простых чисел в интервале и сравнить с величиной *L/ln(x)*. При каких условиях *П(x,L)/L* близко к *1/ln(x)*?

*x*=2000, *L*=500,

- количество простых чисел для деления: 5-15;

- количество оснований: 1-2.

2.2. Найти в заданном интервале все простые числа. Пусть *L(i)* – разность между двумя соседними простыми числами. Построить гистограмму для *L(i)*. Вычислить выборочное среднее $L_{\text{сред.}}$ . Сравнить с величиной *ln(x)*, где *x* – середина интервала.

Интервал: (1000, 1000+300):

- количество простых чисел для деления: 5-20;

- количество оснований: 1-3.

2.3. Для заданного набора чисел $\{k\}$ оценить относительную погрешность формулы для $k$ -го простого числа:

$p(k)=k/ln(k)$, $k=\{10, 15, 20, 30, 35\}$.

3. Методы генерации простых чисел.

3.1. В заданном интервале построить график относительного количества натуральных чисел, проходящих “решето Эратосфена” (т. е. не делящихся на первые $k$ простых).

Интервал: (500, 500+200).

Расчет производится для всех $k<=10$.

3.2. Для заданного интервала:

а) рассчитать точное количество $P_0$ простых чисел в интервале, т.е. при проверке задать только тест на делимость.

Количество первых простых чисел для деления определяется из расчета: максимальное число для деления равно квадратному корню из максимального значения интервала.

б) составить тест с небольшим количеством пробных делений и одним основанием в тесте Ферма. Вычислить количество $P_1$ вероятно простых чисел, удовлетворяющих этому тесту.

в) составить тест с большим, чем в предыдущем случае, количеством пробных делений и двумя или тремя основаниями в тесте Ферма. Вычислить количество $P_2$ вероятно простых чисел, удовлетворяющих этому тесту.

Интервал: (1500,1500 + 300).

Проанализировать полученные данные.

3.3. Известно, что в заданном интервале имеются числа Кармайкла. Найти их. Варианты интервалов: (1050, 1050 + 100) (1700, 1700 + 100) (2400, 2400 + 100)

4. Привести в отчете ответы на контрольные вопросы, в соответствии с номером варианта, указанным преподавателем.

## 7. Содержание отчета

Отчет должен содержать:

1. Наименование работы, цель.

2. Последовательность действий при выполнении каждого задания.

5. Выводы по выполнению лабораторной работы.

## 8. Контрольные вопросы

| **Номер варианта** | **Контрольные вопросы** |
|---|---|
| 1,5,7, 3,9,18,28 | Почему в качестве первого основания в тестах типа теста Ферма для проверки на простоту очень больших чисел целесообразно использовать число 2? |
| 2,4,6,8, 20,22,24, 26,30 | Какова вероятность *P(x)* того, что наугад взятое нечетное очень большое число, не превосходящее *x*, окажется простым? |
| 11,13,15, 10,17,19, 27 | Вычислить:<br>1812 (mod 13), 127 (mod7). |
| 12,14,16 21,23.25, 29 | Сформулируйте суть теста на простоту с использованием пробных делений. |

## 9. Список литературы, рекомендованный к использованию по данной теме

1. Жук, А.П., Жук, Е.П., Лепешкин, О.М., Тимошкин, А.И. Защита информации: Учебное пособие для бакалавров [Текст] / А.П. Жук и др. – М.: ИНФРА-М, 2015. -392с.

2. Шаньгин, В.Ф. Защита компьютерной информации. Эффективные методы и средства [Текст] / В.Ф. Шаньгин – М.: ДМК Пресс, 2012. – 544 с.

## Практическая работа № 6
## Электронная цифровая подпись

**Цель работы:** исследования принципов защищенного электронного документооборота в телекоммуникационных сетях, алгоритмов постановки электронной цифровой подписи (ЭЦП) и умений делать самостоятельные выводы по результатам исследований.

### 2. Формируемые компетенции

Данная Практическая работа направлена на формирование следующих компетенций выпускника бакалавриата:

ОПК-1: способностью инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем;

ПК-5: способностью сопрягать аппаратные и программные средства в составе информационных и автоматизированных систем.

### 3. Теоретическая часть

При обмене электронными документами по сети связи существенно снижаются затраты на обработку и хранение документов, ускоряется их поиск. Но при этом возникает проблема аутентификации автора документа и самого документа, т.е. установления подлинности автора и отсутствия изменений в полученном документе. При обработке документов в электронной форме совершенно непригодны традиционные способы установления подлинности по рукописной подписи и оттиску печати на бумажном документе. Принципиально новым решением является *электронная цифровая подпись (ЭЦП).* Первая схема ЭЦП - *RSA* - была разработана еще в конце 1970-х годов. Однако проблема подтверждения авторства стала актуаль-

ной настолько, что потребовалось установление стандарта, только в 1990-х годах, во время взрывного роста глобальной сети Интернет и массового распространения электронной торговли и оказания услуг. Именно по указанной причине стандарты ЭЦП в России и США были приняты практически одновременно, в 1994 году. Из предложенных криптологами схем ЭЦП наиболее удачными оказались *RSA* и схема Эль-Гамаля. Но первая из них была запатентована в США и ряде других стран (патент на *RSA* прекратил свое действие совсем недавно). Во второй же схеме существует большое количество ее возможных модификаций, и все их запатентовать весьма затруднительно. Именно по этой причине схема ЭЦП Эль-Гамаля осталась по большей части свободной от патентов. Кроме того, эта схема имеет и определенные практические преимущества: размер блоков, которыми оперируют алгоритмы, и соответственно размер ЭЦП в ней оказались значительно меньше, чем в *RSA*, при той же самой стойкости. Именно поэтому стандарты ЭЦП России и США базируются на схеме Эль-Гамаля. Законы об ЭЦП сегодня имеют уже более 60-ти государств. В этом списке значится и Россия. Закон “Об электронной цифровой подписи”, принятый в 2002 году должен оказать стимулирующее воздействие на развитие отечественной электронной коммерции, особенно если в соответствие с ним будут своевременно приведены иные нормативно-правовые акты.

● Правительство РФ финансово поддерживает осуществление федеральной целевой программы «Электронная Россия».

● Принят закон «О внесении изменения в статью 80 части первой Налогового кодекса Российской Федерации».

● Еще в январе 2001 г. правление Пенсионного фонда постановлением «О введении в системе Пенсионного фонда РФ криптографической защиты информации и электронной цифровой подписи» регламентировало

регистрацию и подключение юридических и физических лиц к системе своего электронного документооборота.

● В 2002 г. вышел приказ МНС России «Об утверждении порядка представления налоговой декларации в электронном виде по телекоммуникационным каналам связи», благодаря которому сегодня любое физическое или юридическое лицо может связаться с налоговой инспекцией, используя защищенную электронную почту.

● В 2004 г. были утверждены поправки к статьям 13 и 15 закона «О бухгалтерском учете», согласно которым бухгалтерская отчетность предприятия может вестись, храниться и предоставляться в контролирующие органы в электронном виде.

**Принцип построения ЭЦП.** Асимметрия ролей отправителя и получателя в схемах ЭЦП требует наличия двух тесно связанных ключей: *секретного*, или ключа подписи, и *открытого*, или ключа проверки подписи. Любая схема ЭЦП обязана определить три следующих алгоритма:

● алгоритм генерации ключевой пары для подписи и ее проверки;

● алгоритм постановки подписи;

● алгоритм проверки подписи.

Стандарты России и США очень похожи, они различаются лишь некоторыми числовыми параметрами и отдельными деталями выработки ключевой пары, вычисления и проверки подписи. Действительно, оба стандарта являются вариантами одной и той же схемы ЭЦП Эль-Гамаля. ЭЦП используется для аутентификации текстов, передаваемых по телекоммуникационным каналам. Функционально она аналогична обычной рукописной подписи и обладает основными ее достоинствами:

● удостоверяет, что подписанный текст исходит от лица, поставившего подпись;

● не дает самому этому лицу возможность отказаться от обязательств, связанных с подписанным текстом;

● гарантирует целостность подписанного текста.

ЭЦП представляет собой относительно небольшое количество дополнительной цифровой информации, передаваемой вместе с подписываемым текстом, и включает две процедуры:

● процедуру постановки подписи, в которой используется секретный ключ отправителя сообщения;

● процедуру проверки подписи, в которой используется открытый ключ отправителя.

*Процедура постановки подписи.* При формировании ЭЦП, отправитель, прежде всего, вычисляет хэш-функцию *h(M)* подписываемого текста *M*. Вычисленное значения хэш-функции *h(M)* представляет собой один короткий блок информации *m* , характеризующий весь текст *M* в целом. Затем значение *m* шифруется секретным ключом отправителя. Получаемая при этом пара чисел представляет собой ЭЦП для данного текста *M*.

*Процедура проверки подписи*. При проверке ЭЦП, получатель сообщения снова вычисляет хэш-функцию $m = h(M)$ принятого по каналу текста *M*, после чего при помощи открытого ключа отправителя проверяет, соответствует ли полученная подпись вычисленному значению *m* хэш-функции.

Принципиальным моментом в системе ЭЦП является невозможность подделки ЭЦП пользователя без знания его секретного ключа.

Каждая подпись, как правило, содержит следующую информацию:

● дату подписи;

● срок окончания действия ключа данной подписи;

● информацию о лице, подписавшем текст;

● идентификатор подписавшего (имя открытого ключа);

● собственно цифровую подпись.

*Однонаправленные хэш-функции.* Хэш-функция предназначена для сжатия подписываемого документа *M* до нескольких десятков или сотен бит. Хэш-функция *h(.)* использует в качестве аргумента сообщение *M* произвольной длины и возвращает хэш-значение *h(M)=H* фиксированной длины. Обычно хэшированная информация является сжатым двоичным представлением основного сообщения произвольной длины. Следует отметить, что значение хэш-функции *h(M)* сложным образом зависит от документа *M* и не позволяет восстановить сам документ *M*. Хэш-функция должна удовлетворять целому ряду условий:

● должна быть чувствительна к всевозможным изменениям в тексте *M*;

● должна обладать свойством необратимости, т.е. задача подбора документа *M1* , который обладал бы требуемым значением хэш-функции, должна быть вычислительно неразрешима;

● вероятность того, что значения хэш-функции двух различных документов совпадут, должна быть ничтожно мала. Большинство хэш-функций строится на основе однонаправленной функции *f(.)* , которая образует выходное значение длиной *n* при задании двух входных значений длиной *n*. Этими входами являются блок исходного текста *Mi* и хэш-значение *Hi-1* предыдущего блока текста, рис.1.

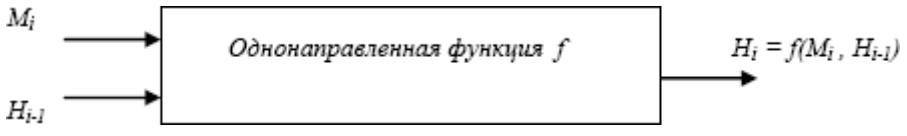



Рисунок 1. Формирование хэш-функции

**Алгоритм цифровой подписи *DSA.*** Алгоритм цифровой подписи *DSA (Digital Signature Authorization)* предложен в 1991 году в США и явля-

ется развитием алгоритма цифровой подписи Эль-Гамаля. Отправитель и получатель электронного документа используют при вычислении большие целые числа:

*G, P – простые числа по L-бит каждое (L = 512 .. 1024 бит);*

*q – простое число длиной 160 бит делитель числа (P – 1).*

*G, P, q* являются открытыми и могут быть общими для всех пользователей сети.

*Описание алгоритма*

1. Отправитель выбирает случайное целое число $X$, $1< X< q$. Число $X$ является *секретным ключом* отправителя для формирования электронной подписи.

2. Отправитель вычисляет значение

$Y = G^X \bmod P$.

Число $Y$ является *открытым ключом* для проверки подписи отправителя и предается всем получателям документа.

*3.* Для того чтобы подписать документ $M$, отправитель хэширует его в целое хэш-значение $m$:

$m = h(M)$, $1<m<q$.

Затем генерирует случайное целое число $K$, $1<K<q$ и вычисляет число $r$:

$r = (G^K \bmod P) \bmod q$.

4. При помощи секретного ключа X отправитель вычисляет число s:

$s = ((m + r \cdot X)/ K) \bmod q$.

Пара чисел $r$ , $s$ образуют цифровую подпись $S = (r , s )$ под документом $M$.

5. Доставленное получателю сообщение вместе с подписью представляет собой тройку чисел *[M, r, s]*. Прежде всего получатель проверяет выполнение соотношений:

$0<r<q;\ 0<s<q.$

6. Далее получатель вычисляет значения:

$w = 1/s \bmod q;$

$m = h(M)$ – *хэш-значение;*

$u_1 = (m \cdot w) \bmod q;$

$u_2 = (r \cdot w) \bmod q.$

Затем при помощи открытого ключа *Y* вычисляется значение

$v = ((G^{u1} \cdot Y^{u2}) \bmod P) \bmod q,$

и проверяется выполнение равенства $v = r$. Если оно выполняется, то подпись признается подлинной, так как можно строго математически доказать, что последнее равенство будет выполняться тогда и только тогда, когда подпись $S = (r, s)$ под документом *M* получена при помощи именно того секретного ключа *X*, из которого был получен открытый ключ *Y*.

**Новые стандарты ЭЦП.** Последние достижения криптографии показали, что общая проблема логарифмирования в дискретных полях, являющаяся базой указанной схемы ЭЦП, не может считаться достаточно прочным фундаментом. Например, размеры блоков, считающиеся "безопасными", растут сравнительно быстрыми темпами. В результате это привело к тому, что стандарты ЭЦП России и США в 2001 году были обновлены: переведены на эллиптические кривые. Схемы ЭЦП при этом остались прежними, но в качестве чисел, которыми они оперируют, теперь используются не элементы конечного поля *GF(2n)* или *GF(p),* а эллиптические числа - решения уравнения эллиптических кривых над указанными конечными по-

лями. Роль операции возведения числа в степень в конечном поле в обновленных стандартах выполняет операция взятия кратной точки эллиптической кривой – “умножение” точки на целое число.

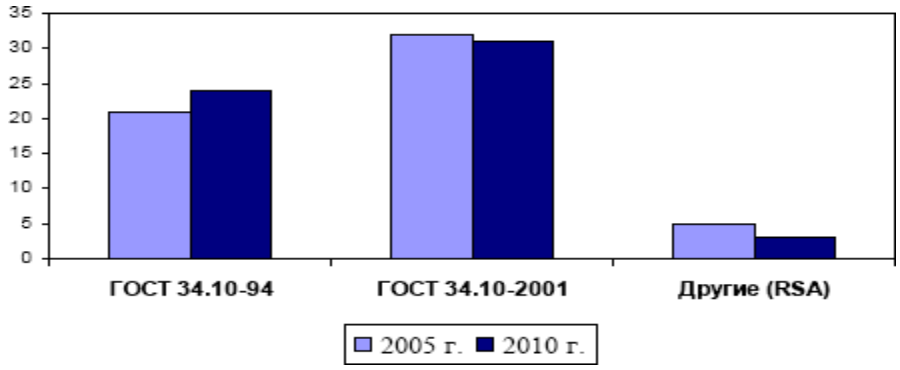



Рисунок 2. Алгоритмы ЭЦП, используемые в РФ

Надлежащий выбор типа эллиптической кривой позволяет многократно усложнить задачу взлома схемы ЭЦП и уменьшить рабочий размер блоков данных. Старый российский стандарт ЭЦП оперирует 1024-битовыми блоками, а новый, основанный на эллиптических кривых, - 256-битовыми, и при этом обладает большей стойкостью. Стойкость схемы подписи ГОСТ Р34.10-94 базируется на сложности решения задачи дискретного логарифмирования в простом поле. В настоящее время наиболее быстрым алгоритмом ее решения для общего случая является алгоритм обобщенного решета числового поля. В ГОСТ Р34.10-2001 стойкость схемы ЭЦП основана на сложности решения задачи дискретного логарифмирования в группе точек эллиптической кривой. При правильном выборе параметров кривой самыми эффективными методами ее решения являются более трудоемкие *r*- и *l*-методы Полларда. Так, по разным оценкам специалистов, трудоемкость взлома старого и нового стандартов ЭЦП России составляет величину порядка 1026 и 1038 операций умножения в базовом поле *GF(p)* соответственно. Очевидно, что новый стандарт более стойкий.

**4. Оборудование и материалы**

Для выполнения работы потребуется:

- класс персональных ЭВМ не ниже Pentium-4, классная доска, мел;

- операционная система Microsoft Windows, программное средство *labWork6.exe*

**5. Указания по технике безопасности**

Требования безопасности перед началом работ:

– ЗАПРЕЩАЕТСЯ: переодеваться, пользоваться огнем, курить, принимать пищу в лаборатории;

– убедиться в целостности электрических розеток и разъемов. Запрещается работать с неисправным электроинструментом, разбивать и самостоятельно ремонтировать его;

– в лаборатории необходимо быть в сменной обуви;

– включение компьютера производить только после получения допуска по выполняемой работе и разрешения преподавателя или лаборанта.

Требования безопасности во время работы:

– выполняя лабораторную работу, студенты обязаны использовать только вычислительную технику, периферийное оборудование, соединительные кабели, измерительное оборудование и дискеты, непосредственно относящиеся к данному лабораторному занятию;

– подключение и отсоединение составляющих вычислительного комплекса производить только при полном снятии напряжения со всех составляющих вычислительного комплекса.

Требования безопасности по окончании работы:

– по команде преподавателя студенты обязаны выключить компьютер и составляющие вычислительного комплекса;

– рабочее место привести в порядок.

**6. Задания**

**Задание №1.** Ознакомиться с основными направлениями работ в рамках федеральной целевой программы “Электронная Россия”, а также со сведениями о порядке использования и действующих алгоритмах постановки электронной цифровой, изложенными в параграфе 3. Запустить программу *labWork6.exe***,** предназначенную для демонстрации порядка постановки и проверки электронной цифровой подписи.

**Задание №2.** Сгенерировать и переслать участникам обмена ключи для шифрования исходного документа и ключи для подписания документа. Исходный текст для шифрования набирается непосредственно в окне программы.

**Задание №3.** Зашифровать исходное сообщение и подписать его на секретном ключе отправителя.

**Задание №4.** Переслать зашифрованное и подписанное сообщение получателю. Выполнить проверку правильности ЭЦП и восстановить исходный текст сообщения.

**Задание №5.** Сохранить в отчете экранные формы, демонстрирующие процесс генерации и распространения ключей; процесс шифрования исходного документа и постановки ЭЦП.

**Задание №5.** Привести в отчете ответы на контрольные вопросы, в соответствии с номером варианта, указанным преподавателем.

**7. Содержание отчета**

Отчет должен содержать:

1. Наименование работы, цель.

2. Последовательность действий при выполнении каждого задания.

3. Выводы по выполнению лабораторной работы.

**8. Контрольные вопросы**

| Номер варианта | Контрольные вопросы |
|---|---|
| 1,5,7, 3,9,18,28 | В чем состоит назначение хэш-функций и какие требования предъявляются к хэш-функциям, используемым для постановки ЭЦП? Перечислите стандарты хэш-функций, действующие в Российской федерации. |
| 2,4,6,8, 20,22,24, 26,30 | Опишите процедуры постановки и проверки ЭЦП. Какая информация содержится в ЭЦП? |
| 11,13,15, 10,17,19, 27 | Перечислите стандарты ЭЦП, действующие в Российской федерации. На каких принципах основана криптостойкость современных алгоритмов ЭЦП? |
| 12,14,16 21,23.25, 29 | Приведите пример реализации алгоритма ЭЦП (*RSA, El-Gamal, DSA*) |

**9. Список литературы, рекомендованный к использованию по данной теме**

1. Жук, А.П., Жук, Е.П., Лепешкин, О.М., Тимошкин, А.И. Защита информации: Учебное пособие для бакалавров [Текст] / А.П. Жук и др. – М.: ИНФРА-М, 2015. -392с.

2. Шаньгин, В.Ф. Защита компьютерной информации. Эффективные методы и средства [Текст] / В.Ф. Шаньгин – М.: ДМК Пресс, 2012. – 544 с.

## Практическая работа № 7

## Шифрование методом скользящей перестановки

**Цель работы:** исследования шифра скользящей перестановки с использованием программной реализации *XY-Move.* и умений делать самостоятельные выводы по результатам исследований.

### 2. Формируемые компетенции

Данная Практическая работа направлена на формирование следующих компетенций выпускника бакалавриата:

ОПК-1: способностью инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем;

ПК-5: способностью сопрягать аппаратные и программные средства в составе информационных и автоматизированных систем.

### 3. Теоретическая часть

### Устойчивые закономерности открытого текста и их использование при дешифровании шифров простой замены и перестановки

Возможность дешифрования какого либо шифра в значительной мере зависит от того, в какой степени криптографические преобразования разрушают вероятностно-статистические закономерности, присутствующие в открытом содержательном тексте. Так в осмысленных текстах любого естественного языка различные буквы встречаются с разной частотой, при этом относительные частоты букв в различных текстах одного языка близки между собой. То же самое можно сказать и о частотах пар, троек букв открытого текста. Кроме того, любой естественный язык обладает так

называемой избыточностью, что позволяет с большой вероятностью “угадывать” смысл сообщения, даже, если часть букв в сообщении не известна.

Таблица 1. Относительные частоты появления букв алфавита русского языка в тексте

| | | | | | |
|---|---|---|---|---|---|
| 1 | а - 0,062 | 12 | л - 0,035 | 23 | ц - 0,004 |
| 2 | б - 0,014 | 13 | м - 0,026 | 24 | ч - 0,012 |
| 3 | в - 0,038 | 14 | н - 0,053 | 25 | ш - 0,006 |
| 4 | г - 0,013 | 15 | о - 0,090 | 26 | щ - 0,003 |
| 5 | д - 0,025 | 16 | п - 0,023 | 27 | ы - 0,016 |
| 6 | е,е - 0,072 | 17 | р - 0,040 | 28 | ъ,ь - 0,014 |
| 7 | ж - 0,077 | 18 | с - 0,045 | 29 | э - 0,003 |
| 8 | з - 0,016 | 19 | т - 0,053 | 30 | ю - 0,006 |
| 9 | и - 0,062 | 20 | у - 0,021 | 31 | я - 0,018 |
| 10 | й - 0,010 | 21 | ф - 0,002 | 32 | - 0,175 |
| 11 | к - 0,028 | 22 | х - 0,009 | | |

Подобные таблицы приводятся в разных книгах. Они получены на основе подсчетов частот на больших объемах открытого текста. Учитывая, что для экспериментов берется различный исходный материал, значения вероятностей несколько отличаются между собой. Если упорядочить буквы по убыванию вероятностей, то мы получим следующий вариационный ряд:

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

Например, в слове **СЕНОВАЛИТР** содержатся 10 наиболее часто встречающихся букв.

Частоты знаков алфавита зависят не только от языка, но и от характера текста. Так в тексте по криптографии будет повышена вероятность букв Ф, Ш (из-за часто встречающихся слов “шифр”, “криптография”). В некоторых математических текстах может быть завышена частота буквы Ф (из-за слов “функция”, “функционал” и т.п.). В стандартных текстовых файлах наиболее частым является символ “пробел”. Частотная диаграмма содержа-

тельных текстов является устойчивой характеристикой текста. Из теории вероятностей следует, что при достаточно слабых ограничениях на вероятностные свойства случайного процесса справедлив закон больших чисел, т.е. относительные частоты $\vartheta/N$ знаков сходятся по вероятности к значениям их вероятностей $p_k$

$$P\{|\frac{\vartheta_k}{N} - p_k| > \varepsilon\} \xrightarrow[N\to\infty]{} 0 .$$

Шифры перестановки и простой замены не полностью разрушают вероятностно-статистические свойства, имеющиеся в открытом сообщении.

При дешифровании текста, зашифрованного шифром простой замены, используют частотные характеристики открытого текста. Именно, если подсчитать частоты встречаемости знаков в шифрованном тексте, упорядочить их по убыванию, и сравнить с вариационным рядом вероятностей открытого текста, то эти две последовательности будут близки.

Скорее всего на первом месте окажется пробел, далее будут следовать буквы О, Е, А, И.

Конечно, если текст не очень длинный, то не обязательно полное совпадение. Может оказаться на втором месте О, а на третьем Е, но в любом случае в первых и вторых рядах одинаковые буквы будут располагаться недалеко друг от друга, и чем ближе к началу (чем больше вероятность знаков), тем меньше будет расстояние между знаками.

Аналогичная картина наблюдается и для пар соседних букв (биграмм) открытого текста (наиболее частая биграмма русского открытого текста - СТ). Однако для получения устойчивой картины длина анализируемой последовательности должна быть достаточно большой. На сравнительно небольших отрезках открытого текста эта картина как-то смазана. Более устойчивой характеристикой биграмм является отсутствие в осмысленном тексте некоторых биграмм, как говорят, наличие запретных биграмм, име-

ющих вероятность равную практически 0. Видели ли вы когда-нибудь в открытом тексте биграмму ЪЬ, или биграммы вида: “гласная” Ь; “пробел” Ь? Знание и использование указанных особенностей открытого текста значительно облегчает дешифрование шифра перестановки и замены.

**Шифр перестановки**. Положим X – множество открытых (содержательных) текстов в алфавите I. Длины всех возможных открытых текстов кратны величине T. Множеством ключей является симметрическая группа подстановок $S_T$ степени T, для $g \in S_T$ определим функцию шифрования fg положив для $(i_1,i_2,\ldots,i_T) \in X$

$fg(i_1,i_2,\ldots,i_T) = (i_{g(1)},i_{g(2)},\ldots,i_{g(T)})$;

доопределим fg на остальных элементах из X по правилу: текст $x \in X$ делится на отрезки длины T и каждый отрезок длины T шифруется на ключе g по указанному выше закону шифрования. Последовательность, составленная из букв образов зашифрованных слов, является шифрованным текстом, соответствующим открытому тексту x и ключу g. Для шифрования текста длины не кратной T его дополняют буквами до длины кратной T.

**Дешифрование шифра перестановки.** Шифрованный текст записывается в таблицу с T столбцами. Для восстановления открытого текста шифра перестановки нам необходимо переставить колонки таким образом, чтобы в строках появился осмысленный текст.

Рассмотрим пример дешифрования шифра перестановки восьми столбцов (Учебное пособие «Принципы и методы защиты информации» Проскурин Г.В.). Пусть шифротекст имеет следующий вид:

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| п | а | я | | в | | и | м |
| о | ч | ш | г | | у | | е |
| е | б | ж | л | | е | | о |
| м | | ч | | о | т | о | я |
| | е | г | е | | у | с | щ |
| | а | к | ь | з | а | т | т |
| я | р | е | | е | п | | ь |
| | ю | з | в | а | н | в | |
| о | й | а | в | е | ш | л | |
| | е | е | я | м | | п | н |
| ь | р | р | н | з | е | е | е |
| з | а | м | а | н | | а | к |
| ч | с | т | а | | ь | а | н |
| о | я | л | м | | а | л | |
| о | ь | ч | х | т | а | т | |
| в | | е | о | а | л | е | п |
| о | е | р | м | т | ь | е | |
| д | с | г | ы | | о | а | т |
| е | б | в | н | | ы | | |
| | а | у | и | н | з | н | л |
| | г | и | а | о | к | к | д |
| | а | о | б | д | г | н | |
| | ж | а | у | е | д | я | д |
| х | л | и | | е | м | о | а |
| к | р | т | д | | ь | о | е |
| | ь | х | в | т | о | н | |
| р | л | е | | е | д | а | ю |
| р | | з | е | в | | е | д |
| ш | | в | а | е | н | е | н |
| т | и | й | е | в | | | д |
| | | в | | с | д | | |

Сопоставим перестановке столбцов таблицу 8*8, при этом поставим на пересечении i-той строки и j-ого столбца единицу, если j-ая колонка после обратной перестановки должна следовать за i-ой. Наша задача восстановить таблицу, отвечающую правильной перестановке столбцов.

Давайте теперь попарно пристраивать один столбец к другому. Если при этом в некоторых строках появляются запретные биграммы, то столбцы не могут в открытом тексте следовать друг за другом и соответствующая клеточка зачеркиваются. В нашем примере 6-ой столбец не может следовать за 4-ым, т.к. иначе в тексте в первой строке будет подряд два пробе-

ла. Посмотрим, например, 6-ую строку. Если бы 4-ый столбец следовал за 1-ым, то в тексте были бы слова, начинающиеся с Ь.

После просмотра всех строк мы получим следующую таблицу:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | x | x | | x | x | x | | x |
| 2 | | x | | x | | x | | |
| 3 | | | x | | | | | x |
| 4 | x | x | | x | | x | x | x |
| 5 | x | | | | x | x | x | x |
| 6 | x | x | | x | | x | x | |
| 7 | x | | | x | x | x | x | x |
| 8 | x | x | | | x | x | x | x |

Если бы текст был подлиннее и строк было бы побольше, то в каждой строке и в каждом столбце осталось бы ровно по одной не зачеркнутой клеточке и перестановка была бы восстановлена.

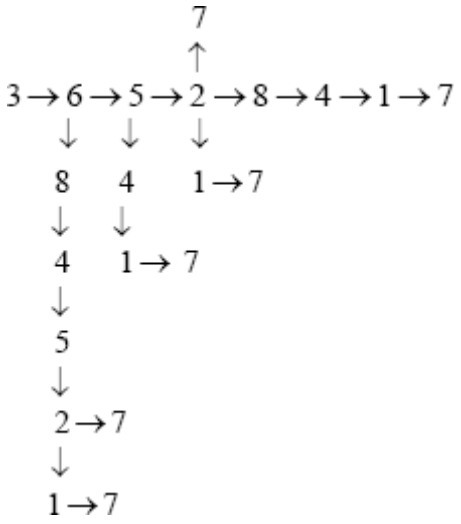
В нашей таблице мы только можем утверждать, что 6-ой столбец следует за 3-им (обозначим это событие следующим образом 3→6), если 6-ой столбец не является последним. Для 6-го столбца может быть два варианта продолжения

$$\begin{array}{ccccc} & & 8 & & \\ & & \uparrow & & \\ 3 & \rightarrow & 6 & \rightarrow & 5 \end{array}$$

Нам надо рассмотреть оба и постараться отсеять ложный вариант. Если отсеять ложный вариант не удастся, то надо продолжать оба варианта

$$\begin{array}{ccccccccc} & & 8 & \rightarrow & 4 & & 1 & & \\ & & \uparrow & & & & \uparrow & & \\ 3 & \rightarrow & 6 & \rightarrow & 5 & \rightarrow & 2 & \rightarrow & 7 \end{array}$$

В итоге получаем некоторое дерево возможного следования столбцов в открытом тексте.




Каждой ветви дерева соответствует некоторая перестановка столбцов

Далее проверяем каждый вариант на осмысленность и получаем правильный вариант.

$$3 \to 6 \to 5 \to 2 \to 8 \to 4 \to 1 \to 7$$

Заметим, что не обязательно было строить дерево до конца. Например, ветвь 3→ 6 →8 →4 →5 можно было отсеять сразу. Разве можно признать осмысленным следующий фрагмент текста.

| 3 | 6 | 8 | 4 | 5 |
|---|---|---|---|---|
| я |  | м |  | в |
| ш | у | е | г |  |
| ж | е | о | л |  |
| ч | т | я |  | о |
| г | у | щ | е | з |
| к | а | т | ь | е |
| е | а | т |  | а |

Такая процедура отсечения ветвей была бы просто необходима, если бы строк было поменьше, и дерево было бы соответственно гораздо ветвистей.

Предложенную процедуру легко автоматизировать и сделать пригодной для реализации на ЭВМ. Алгоритм дешифрования должен состоять из следующих этапов.

1. Предварительная работа. Анализируя достаточно представительный объем открытых текстов построить множество запретных биграмм.

2. Предварительная работа. Составить словарь всех возможных v-грамм для v=2,3,...,d, которые могут встретиться в открытом тексте. Число d выбирается, исходя из возможностей вычислительной техники.

Построить таблицу 8*8. При этом перебираются последовательно все запретные биграммы и для каждой опробуемой биграммы - последовательно все строки. Если хотя бы в одной строке первый символ биграммы встречается в i-том столбце, а второй в - j-ом, то клеточка i x j таблицы зачеркивается.

3. Выбрать некоторый столбец в качестве начального столбца.

4. Начать процедуру построения дерева путем пристраивания к исходному столбцу всех вариантов столбцов.

5. Для каждого полученного варианта добавить еще один из оставшихся столбцов. Если хотя бы в одной из строк таблицы встретится 3-грамма, которая отсутствует в словаре размещенных 3-грамм, то вариант отсеивается.

6. Для каждого из не отсеянных вариантов добавляем еще один столбец и проводим отсев ложных вариантов по словарю разрешенных 4-грамм.

Если словарь был построен только для $d \leq 3$, то отсев проводится путем проверки на допустимость 3-грамм, встретившихся в последних 3-х столбцах каждой строки. Продолжаем этот процесс до получения полной перестановки.
Ниже, в таблице, приведен восстановленный для нашего примера текст.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 1 | я | | в | а | м | | п | и |
| 2 | ш | у | | ч | е | г | о | |
| 3 | ж | е | | б | о | л | е | |
| 4 | ч | т | о | | я | | м | о |
| 5 | г | у | | е | щ | е | | с |
| 6 | к | а | з | а | т | ь | | т |
| 7 | е | п | е | р | ь | | я | |
| 8 | з | н | а | ю | | в | | в |
| 9 | а | ш | е | й | | в | о | л |
| 10 | е | | м | е | н | я | | п |
| 11 | р | е | з | р | е | н | ь | е |
| 12 | м | | н | а | к | а | з | а |
| 13 | т | ь | | с | н | а | ч | а |
| 14 | л | а | | я | | м | о | л |
| 15 | ч | а | т | ь | | х | о | т |
| 16 | е | л | а | | п | о | в | е |
| 17 | р | ь | т | е | | м | о | е |
| 18 | г | о | | с | т | ы | д | а |
| 19 | в | ы | | б | | н | е | |
| 20 | у | з | н | а | л | и | | н |
| 21 | е | к | о | г | д | а | | к |
| 22 | о | г | д | а | | б | | н |
| 23 | а | д | е | ж | д | у | | я |
| 24 | и | м | е | л | а | | х | о |
| 25 | т | ь | | р | е | д | к | о |
| 26 | х | о | т | ь | | в | | н |
| 27 | е | д | е | л | ю | | р | а |
| 28 | з | | в | | д | е | р | е |
| 29 | в | н | е | | н | а | ш | е |
| 30 | й | | в | и | д | е | т | ь |
| 31 | в | а | с | | | | | |

Дальнейшее развитие шифры перестановки получили при осуществлением идеи непрерывной локальной перестановки символов открытого текста под действием управляющей последовательности. Для осуществления перемешивания знаков открытого текста в памяти шифратора запоминаются отдельные знаки текста и проводится задержка их передачи в дискретном времени. Введем параметры $n_1$ и $n_2$, так, что $n=n_1+n_2$. В этих шифрах i-й знак передаваемого сообщения переставляется в шифрованном сообщении на j-е место, где $i-n_1 \le j \le i+n_2$ . Управляющую последовательность временем задержки стараются выбрать так, чтобы время задержки каждого символа было случайной величиной с равномерным распределением, то есть вероятность каждого фиксированного значения времени задержки должна быть близка к *1/n.*

**Шифрующий автомат скользящей перестановки**

Рассмотрим схему шифрующего автомата, позволяющего при подходящей управляющей последовательности реализовать произвольный ши скользящей перестановки, рис.7.1.

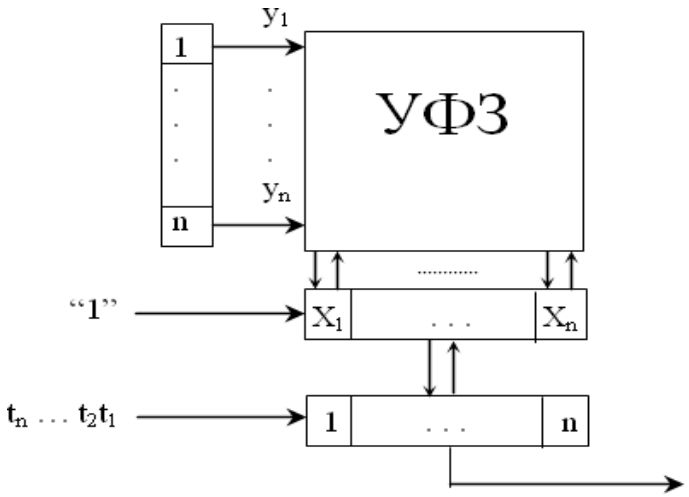



Рис. 7.1. Схема реализации шифрующего автомата скользящей перестановки

На вход узла формирования задержки (УФЗ) в каждом такте $t$ подается вектор $\vec{\ }$

Узел формирования задержки является конечным автоматом ( $F^n$ , $F^n$ , $\{1,..., n\}$,$\delta$ , $\lambda$), множеством состояний которого является множество всевозможныхдвоичных векторов - заполнений $x(t)=(x^t,...,x^{t)}$ нижнего накоителя.

В такте $t$ вырабатывается натуральное число – значение задержки.

$t$

$$\gamma_t = \max j : \{x_j^t \quad y_i^t \in \{0,1\}, \ i=\overline{2,n-1}, \ y_1^t = y_n^t = 1.$$

При этом автомат переходит в следующее состояние:

$$\bar{x}=(t+1)=(x_1^{t+1},\ldots,x_n^{t+1}),$$

где

$$x_1^{t+1}=1,$$

$$x_i^{t+1}=\begin{cases}0, & i=\gamma_t+1\\ x'_{i-1}, & i\in\{2,\ldots,\gamma_t,\gamma_t+2,\ldots,n\}\end{cases}$$

Знаки открытого текста записываются на нижний накопитель. В линию в *t*-ом такте посылается знак открытого текста, записанный в ячейке с номером.

Состояния автомата $x$ (t) являются индикаторами, показывающими какие из знаков открытого текста еще не считаны с нижнего накопителя.

Для зашифрования последовательности $t_N,\ldots,t_2,t_1$ поступают следующим образом. Сначала записываем в нижний накопитель первые $n_1$ знаков открытого текста

$(t_{n1},\ldots,t_1,0,\ldots,0)$

Одновременно автомат устанавливается в начальное состояние

$x(1)=(0,\ldots,0)$.

После этого автомат УФЗ начинает работать по описанному выше закону до тех пор, пока в накопитель не поступит последний знак $t_N$ открытого текста. С прекращением подачи на накопитель знаков открытого текста происходит прекращение подачи единиц на накопитель-индикатор. В оставшееся $n_1$ тактов производится считывание записанной в накопителе информации.

При расшифровывании УФЗ работает по той же схеме, только вместо считывания необходимо организовывать запись информации во второй накопитель.

**Особенности работы УФЗ**

В каждом такте $t$ (за исключением последних $n_1$ тактов) в накопителе-индикаторе $\vec{x}$ (t) записано ровно $n_1$ единиц. Поэтому в такте $t$ величина задержки может принимать только одно из $n_1$ значений в интервале $\{1,\ldots,n\}$. В частном случае, когда $n$=1, либо $n_1$=$n$ УФЗ вырабатывает постоянно значения задержки $\gamma_t = 1$ и $\gamma_t = n$ соответственно. Легко видеть, что результирующее преобразование открытого текста действительно будет шифром скользящей перестановки. Условие $y_1^t$ =1, t=1,2,… обеспечивает постоянное считывание во всех тактах, а условие $y_2^t$ =1 ограничивает величину задержки $\gamma_t \le$n.

**Пример, поясняющий работу шифрующего автомата скользящей перестановки**
Возьмем для примера n=7, $n_1$=3, $n_2$=4

На вход УФЗ на каждом шаге $t$ работы шифрующего автомата подается вектор

$\vec{y} = \left(y_1^t,\ldots,y_n^t\right)$, в линейном регистре сдвига (ЛРС), рис. 7.2.

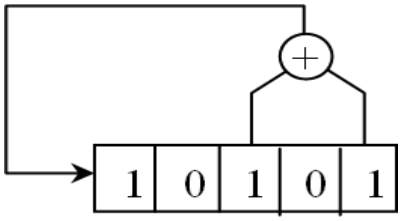
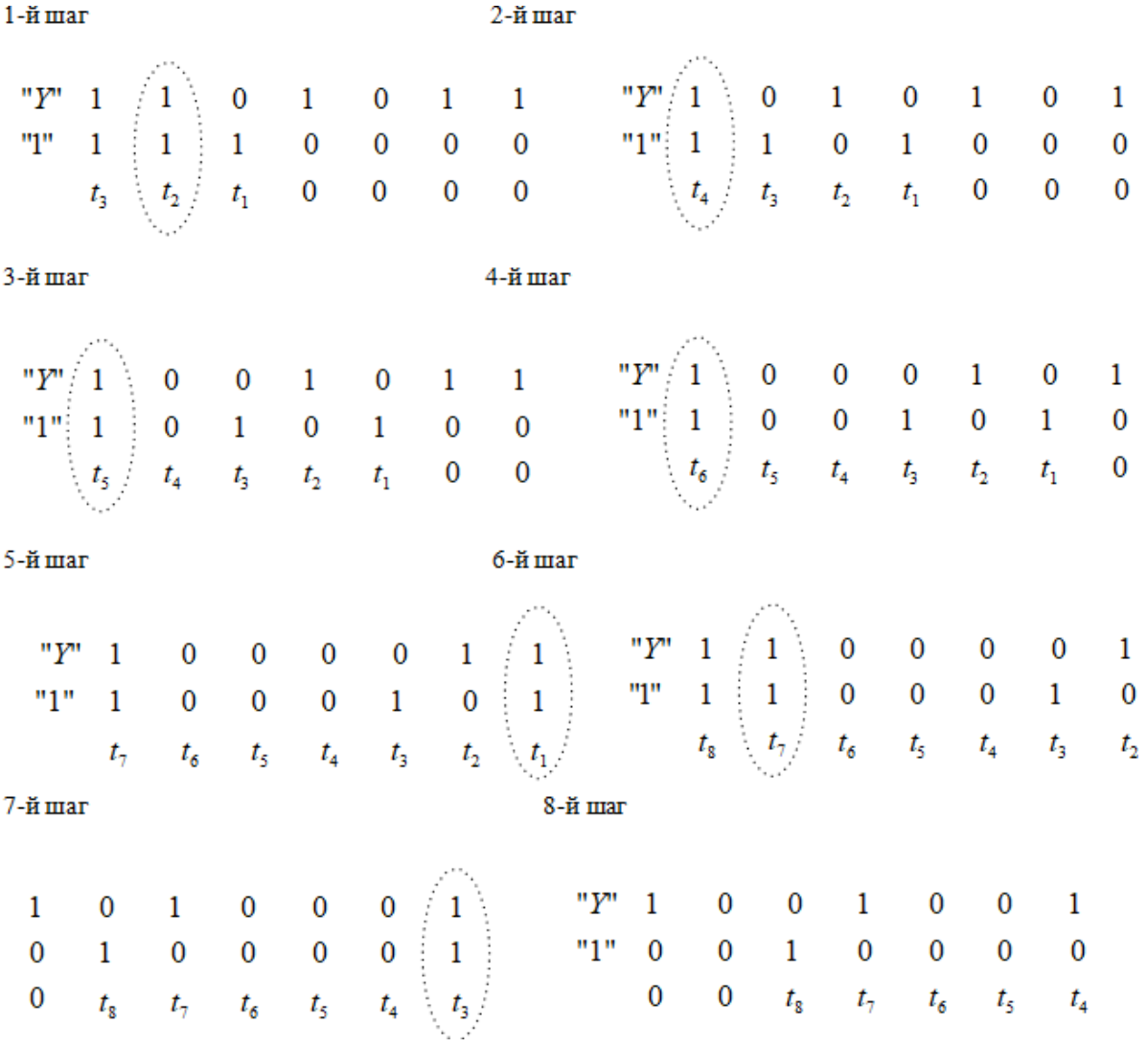



Рис.7.2. Линейный регистр сдвига

Мы надеемся, что читатель сможет написать последовательность состояний данного линейного регистра сдвига, с помощью которой образуется управляющая последовательность шифрующего автомата.

Будем обозначать на каждом шаге работы шифрующего автомата последовательность $Y_1,Y_2,\ldots,Y_n$ (в нашем случае $Y_1,Y_2,\ldots,Y_7$), поступающую с ЛРС, как

«Y», а последовательность единиц $X_1,X_2, \ldots, X_n$ (в нашем случае $X_1,X_2, \ldots, X_7$) как «1». В нижней строке будем записывать знаки открытого текста, находящиеся на данном шаге в нижнем накопителе шифрующего автомата $t_1, t_2, \ldots, t_n$ (в нашем случае $t_1, t_2, \ldots, t_7$). Рассмотрим пошагово работу шифратора при конкретных условиях.




На каждом шаге мы искали начиная с левого края и идя направо первое совпадение в строках “Y” и “1” (1 в обеих строках) и для удобства обводили этот столбец. Элемент открытого текста, который оказался в выбранном столбце уходит в линию. Таким образом, в нашем примере, последовательность ушедшая в линию имеет следующий вид: $t_2,t_4,t_5,t_6,t_1,t_7,t_3,t_8$

**Описание программной реализации**

Для выполнения лабораторной работы на компьютере необходимо установить программный модуль XY-Mover.

**1) Строка меню** В данной программе меню состоит из трех пунктов: Шифрование, Вид, Помощь. Необходимый пункт меню можно выбрать, щелкнув по нему левой кнопкой мыши, или с помощью кнопок клавиатуры «вправо», «влево», нажав перед этим функциональную клавишу «F10». После того как пользователь выбрал необходимый ему пункт меню, откроется ниспадающее подменю. Рассмотрим пункты меню подробнее.

● *Шифрование.* Пункты ниспадающего меню можно выбрать либо левой кнопкой мыши, либо кнопками клавиатуры «вверх», «вниз». Рассмотрим подробнее пункты подменю.

- Редактирование параметров – эта функция меню позволяет задать необходимые параметры в поле окна программы «Параметры шифратора»;

- Шифровать – выбор этого пункта меню запускает процесс шифрования;

- Дешифрование – выбор этого пункта меню запускает процесс дешифрования;

- Выход – завершение работы программы.

● *Вид.* Этот пункт меню позволяет выбрать внешний вид программы. Ниже приводятся пункты подменю:

- Параметры – выбор этого пункта меню позволяет использовать поле «Параметры шифратора», описанную ниже.

- Схема шифратора – выбор этого пункта меню позволяет наблюдать структурную схему шифрующего автомата.

- Строка состояния – выбор этого пункта меню позволяет наблюдать строку состояния, описанную ниже.

**2) Панель инструментов**

Возможно, пользователю будет удобнее воспользоваться панелью инструментов (так называемыми «кнопками»), вместо работы с меню. «Кнопки» дублируют некоторые пункты меню, но выбрать «кнопку» гораздо удобнее, чем пункт подменю. Для этого необходимо просто щелкнуть по выбранной кнопке левой кнопкой мыши. Рассмотрим «кнопки» слева направо:

- «А», выбор этой кнопки позволяет получить результат аналогичный пункту меню: Вид-Параметры;
- «Схема шифратора » – выбор этой кнопки позволяет получить результат аналогичный пункту меню Вид- Схема шифратора;
- «Строка состояния» – выбор этой кнопки позволяет получить результат аналогичный пункту меню: Вид- Строка состояния;
- «Редактирование параметров» – выбор этой кнопки позволяет получить результат аналогичный пункту меню: Шифрование-Редактирование параметров;
- «Шифровать» – выбор этой кнопки позволяет получить результат аналогичный пункту меню Шифрование – Шифровать;
- «Дешифровать» – выбор этой кнопки позволяет получить результат аналогичный пункту меню Шифрование – Дешифровать;
- «Прервать» – выбор этой кнопки позволяет прервать процесс обработки данных (шифрование или дешифрование).

**3) Строка состояния**

Строка состояния находится в нижней части окна программы. На ней выводится информация о состоянии программы:

- «Шифрование методом скользящей перестановки» – это сообщение указывает на то, что программа готова к работе.
- «Завершено … %» – индикация объема выполненной работы при зашифровании или расшифровании.

**4) Описание полей окна программы**

● Параметры шифратора

- *n1* – число знаков, которые записываются в нижний накопитель первыми;

- *n2* – остальные знаки (всего их 127);

- отводы регистра – точки съема ЛРС;

- начальное заполнение – начальное заполнение ЛРС, которое пользователь может изменять.

● Входной поток

- поле, в котором отражается имя текстового файла, содержание которого нужно шифровать/расшифровать;

- кнопка «открыть» – при нажатии на эту кнопку открывается стандартное для *Windows* окно «Открытие файла»;

- текст – содержимое файла, открытого для шифрования/расшифрования;

- битовый поток – побитное представление символов выбранного файла.

● Выходной поток

- поле, в котором отражается имя текстового файла, содержание которого нужно расшифровать/шифровать;

- кнопка «открыть» – при нажатии на эту кнопку открывается стандартное для *Windows* окно «Открытие файла»;

- текст – содержимое файла, открытого для расшифрования/шифрования;

- битовый поток – побитное представление символов выбранного файла.

## 4. Оборудование и материалы

Для выполнения работы потребуется:

- класс персональных ЭВМ не ниже Pentium-4, классная доска, мел;

- операционная система Microsoft Windows, программное средство *labWork6.exe*

## 5. Указания по технике безопасности

Требования безопасности перед началом работ:

– ЗАПРЕЩАЕТСЯ: переодеваться, пользоваться огнем, курить, принимать пищу в лаборатории;

– убедиться в целостности электрических розеток и разъемов. Запрещается работать с неисправным электроинструментом, разбивать и самостоятельно ремонтировать его;

– в лаборатории необходимо быть в сменной обуви;

– включение компьютера производить только после получения допуска по выполняемой работе и разрешения преподавателя или лаборанта.

Требования безопасности во время работы:

– выполняя лабораторную работу, студенты обязаны использовать только вычислительную технику, периферийное оборудование, соединительные кабели, измерительное оборудование и дискеты, непосредственно относящиеся к данному лабораторному занятию;

– подключение и отсоединение составляющих вычислительного комплекса производить только при полном снятии напряжения со всех составляющих вычислительного комплекса.

Требования безопасности по окончании работы:

– по команде преподавателя студенты обязаны выключить компьютер и составляющие вычислительного комплекса;

– рабочее место привести в порядок.

**6. Задания**

**1.** Для выполнения лабораторной работы на компьютере необходимо установить программный модуль XY-Mover.

**2.** Выполнить начальные установки шифратора, согласно примеру.

**3.** Загрузить файл для шифрования.

**4.** Произвести шифрование информации с использованием шифра скользящей перестановки, сохранить шифртекст в файле.

**5.** Описать в отчете процесс шифрования и расшифрования данных с использованием программы-эмулятора XY-Mover. Проанализировать полученные данные.

**6.** Включить в отчет о лабораторной работе ответы на контрольные вопросы, выбранные в соответствии с номером варианта, указанным преподавателем.

## 7. Содержание отчета

Отчет должен содержать:

1. Наименование работы, цель.
2. Последовательность действий при выполнении каждого задания.
3. Выводы по выполнению лабораторной работы.

## 8. Контрольные вопросы

| Номер варианта | Контрольные вопросы |
|---|---|
| 1,5,7, 3,9,18,28 | Почему шифрование методом гаммирования является наиболее подходящим для высокоскоростных линий телекоммуникационной связи? |
| 2,4,6,8, 20,22,24, 26,30 | Какие общие требования, предъявляются к гамме шифра? |
| 11,13,15, 10,17,19, 27 | Приведите пример, поясняющий работу шифрующего автомата скользящей перестановки при $n$=5, $n1$=2, $n2$=3. |
| 12,14,16 21,23.25, 29 | Кратко опишите работу схемы реализации шифра скользящей перестановки. |

## 9. Список литературы, рекомендованный к использованию по данной теме

1. Жук, А.П., Жук, Е.П., Лепешкин, О.М., Тимошкин, А.И. Защита информации: Учебное пособие для бакалавров [Текст] / А.П. Жук и др. – М.: ИНФРА-М, 2015. -392с.

2. Шаньгин, В.Ф. Защита компьютерной информации. Эффективные методы и средства [Текст] / В.Ф. Шаньгин – М.: ДМК Пресс, 2012. – 544 с.

## ПРАКТИЧЕСКАЯ РАБОТА №8

## ИЗУЧЕНИЕ ПРОГРАММНЫХ ПРОДУКТОВ ЗИ.
## ПРОГРАММА *PGP* (Pretty Good Privacy)

**Цель работы:** Ознакомление с общими принципами построения и использования программных средств защиты информации, в частности с программой **PGP.**

Освоение средств программной системы **PGP**, предназначенных для:

- шифрования конфиденциальных ресурсов и разграничения доступа к ним;
- обеспечения целостности информационных ресурсов с помощью механизма электронной цифровой подписи;
- надежного уничтожения остаточной конфиденциальной информации;
- скрытия присутствия в компьютерной системе конфиденциальной информации с помощью виртуального диска.
- Для выполнения лабораторной работы при отсутствии на компьютере программы PGP ее необходимо инсталлировать.
- Инсталляционный файл прилагается к описанию лабораторной работы PGPfreeware602i. Выбрать для установки только следующие компоненты:
  - PGP 6.0.2 Program Files;
  - PGP 6.0.2 User’s Manual;
  - Unconfigured PGP 6.0.2 Client Install;
  - PGPdisk for Windows.

На вопрос программы установки о существовании ключей ответить «Нет», а на вопрос о необходимости перезагрузки системы – «Да».

**Описание лабораторной работы**

**Возможности *PGP 6.0***

Основные задачи программы *PGP* – шифровать файлы и почтовые сообщения, заверять их электронными подписями, полностью уничтожать файлы на диске. Кроме этого, *PGP* предоставляет следующие возможности:

– хранение открытых ключей на удаленном сервере;

– использование трех симметричных алгоритмов шифрования и двух асимметричных;

– четыре способа запуска: *E-mail plugins, PGP tray, PGP tools*, контекстное меню;

– разделение ключей;

– установка уровня валидности ключа и доверия владельцу ключа.

**PGPkeys**

Это программа, входящая в состав *PGP 6.0,* и обеспечивающая работу с ключами, рис.8.1.

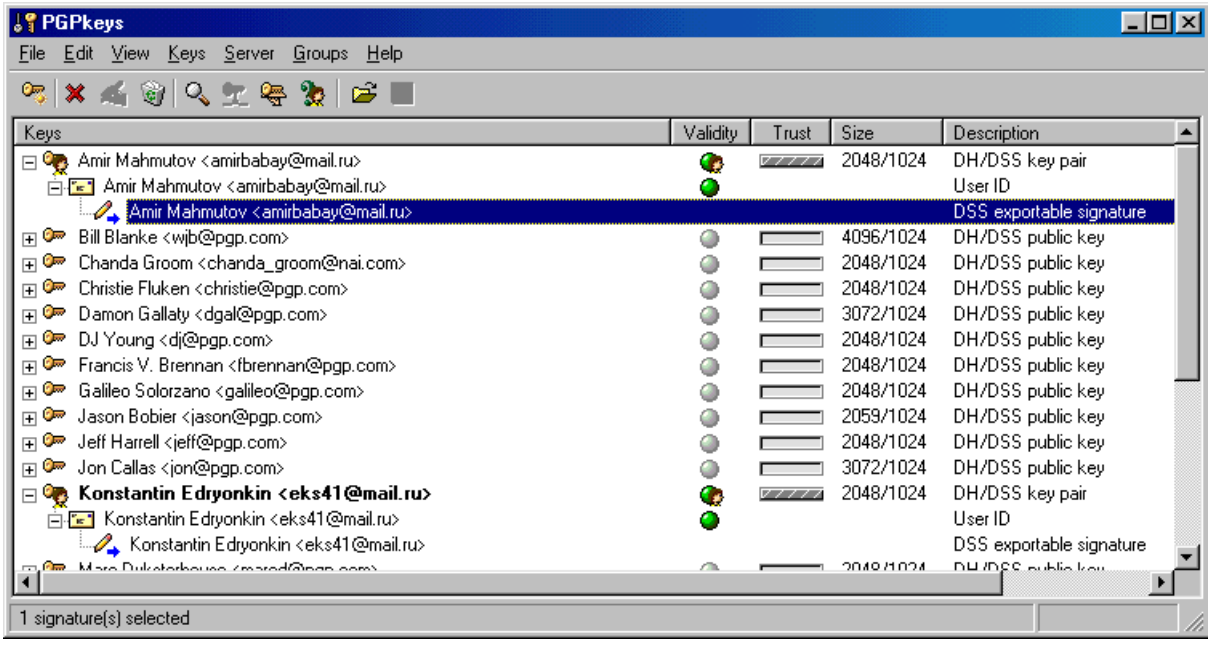



Рис. 8.1 Окно PGPkeys

В меню **File** содержатся две команды, одна из которых **Exit**, другая – **Send Key Shares** позволяет послать локальную копию разделенного ключа по сети.

Меню **Edit** кроме обычных команд содержит команду **Preferences**, которая служит для задания настроек программы, рис.8.2.

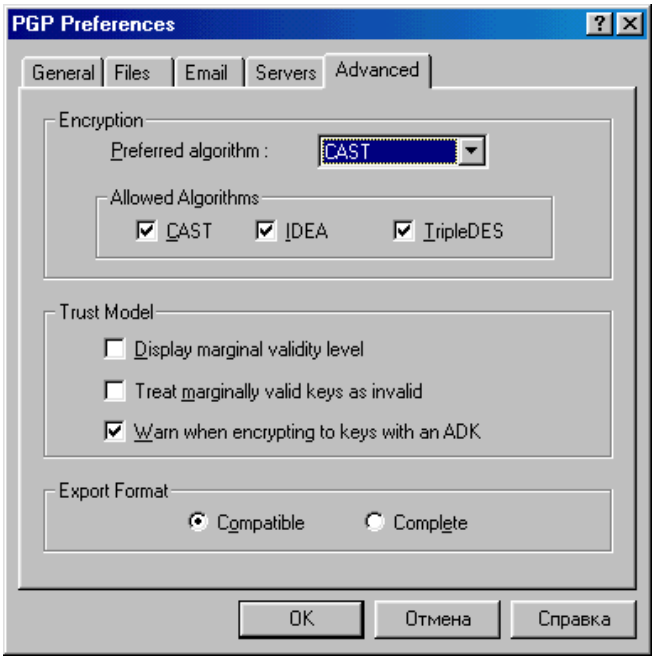



Рис. 8.2. Окно PGPPreferences

С помощью меню **View** можно задать отображаемые на экране свойства ключей.

**Команды меню *Keys***

**Sign** – позволяет подписать своим закрытым ключом открытые ключи других пользователей. Тем самым вы можете показать, что доверяете владельцу используемого ключа, то есть ключ действительно принадлежит владельцу. При этом можно задать модификаторы: 1) *Non Exportable* – для локального набора ключей; 2) *Exportable* – заверенный ключ отсылается на сервер; 3) *Meta-introducer Non-Exportable* – доверяете не только владельцу, но и любому доверенному им ключу; 4) *Trusted Introducer Exportable* – вы доверяете владельцу подписывать ключи, рис.8.3.

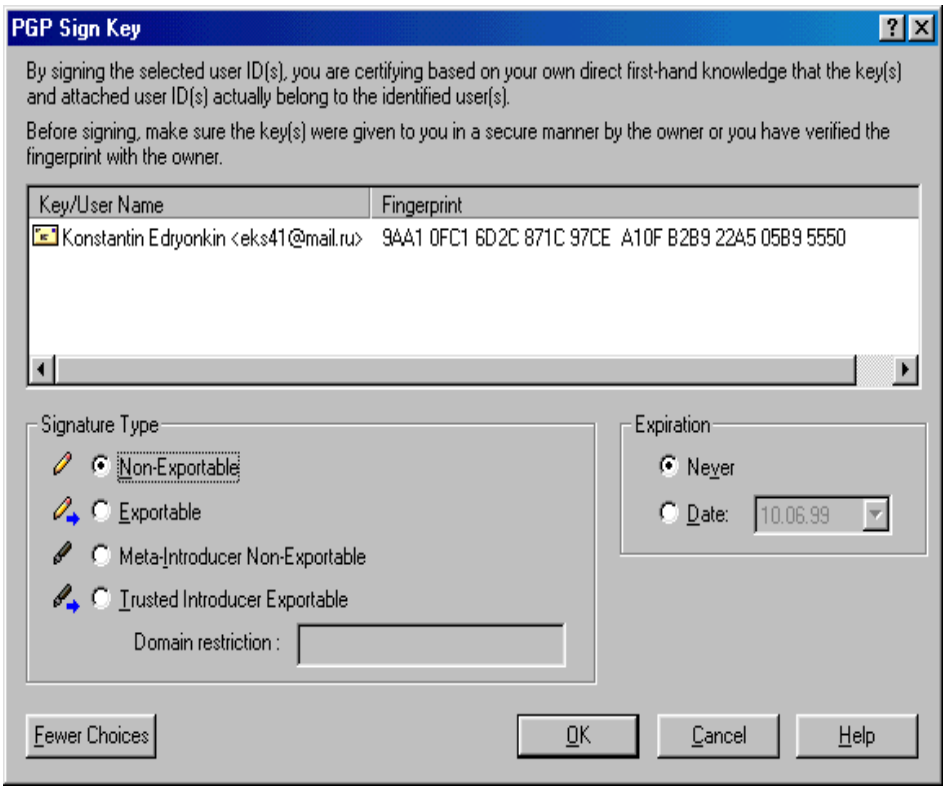



Рис.8.3. Окно PGP SignKey

**Set as Default Key** – назначить данный ключ ключом по умолчанию. **Add Name/Photo/Revoker** – позволяет добавить к свойствам ключа имя владельца, имеющего право объявить ваш ключ недействительным.

**Revoke** – объявить ключ недействительным.

**Reverify Signatures** – проверить сигнатуру (правильность) ключа.

**New Key** – добавить новый ключ.


Рис. 8.4. Окно Key Generation Wizard

**Share Split** – разделить ключ между несколькими владельцами. **Import/Export** – импортировать/экспортировать ключ из/в текстовый файл. **Key Properties** – свойства (при этом можно задать степень доверия, валидность ключа), рис.8.4, 8.5.

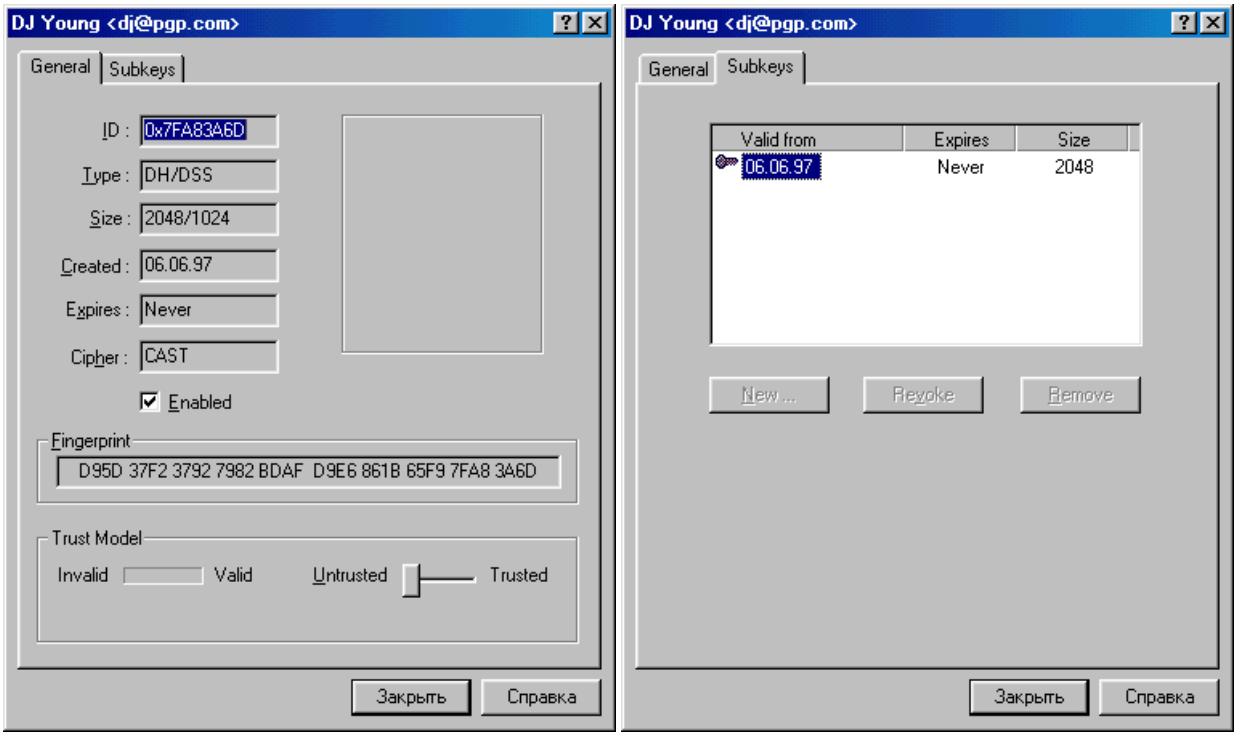



Рис. 8.5. Свойства ключа

В меню **Server** сосредоточены команды для работы с удаленным сервером ключей.

Команда Send to позволяет послать ключи на выбираемый сервер, рис.8.6.

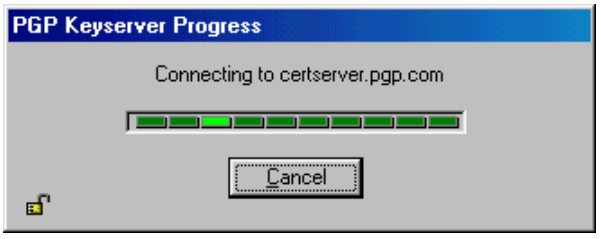



Рис.8.6. Соединение с удаленным сервером

При помощи команды Search можно попытаться найти на удаленном сервере ключ, задав соответствующие параметры, рис.8.7.

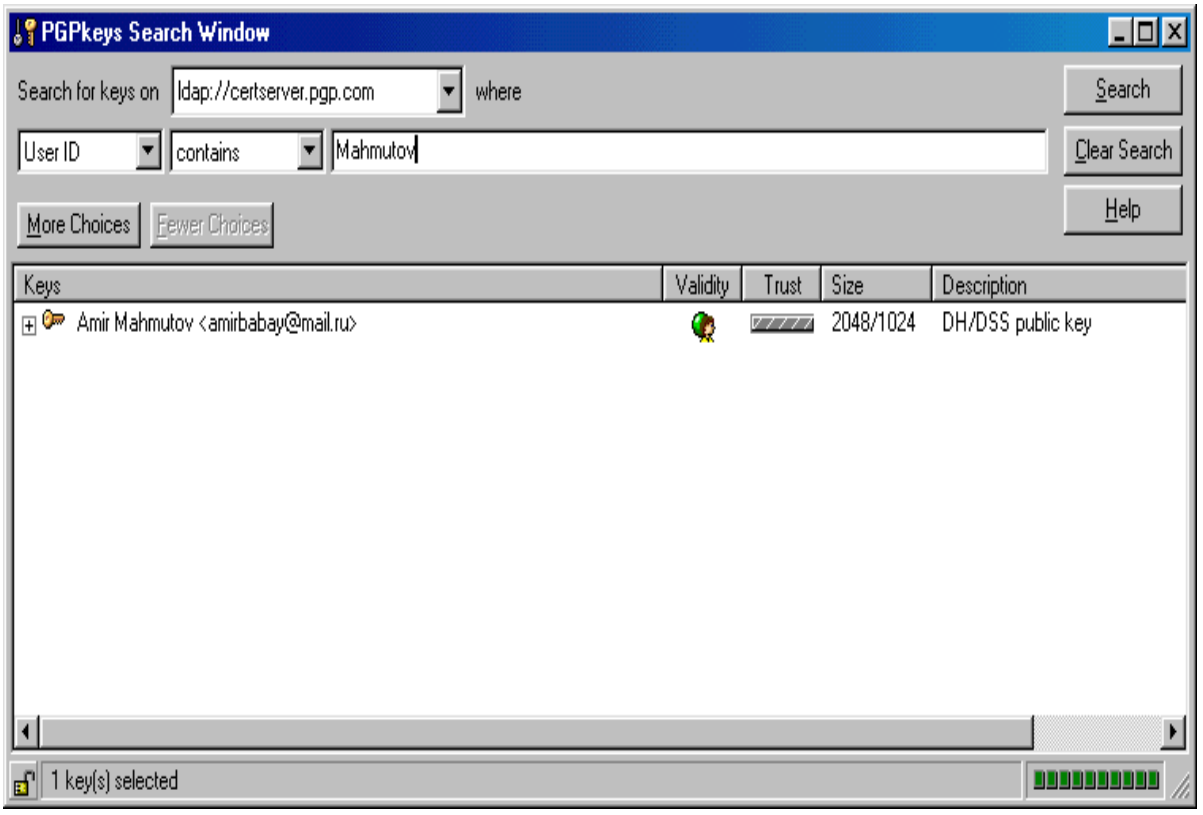



Рис. 8.7. Поиск ключа на удаленном сервере

Команда **Update** позволяет обновить выбранный ключ, получив информацию с сервера.

В меню **Groups** содержатся команды для работы с группами. Команды **New Group, Show Groups, Import Groups** позволяют,

соответственно, добавить группу, показать группы, импортировать группы из файла. Объединение владельцев ключей в группы позволяет легко и просто шифровать сообщения для отправки всем членам группы.

Наконец, меню **Help**, стандартное для *Windows*, позволяет получить справку.

**PGPtray**

Загружается при запуске *Windows*. Для активизации меню достаточно нажать кнопкой мыши на значок рядом с часами, рис.8.8.

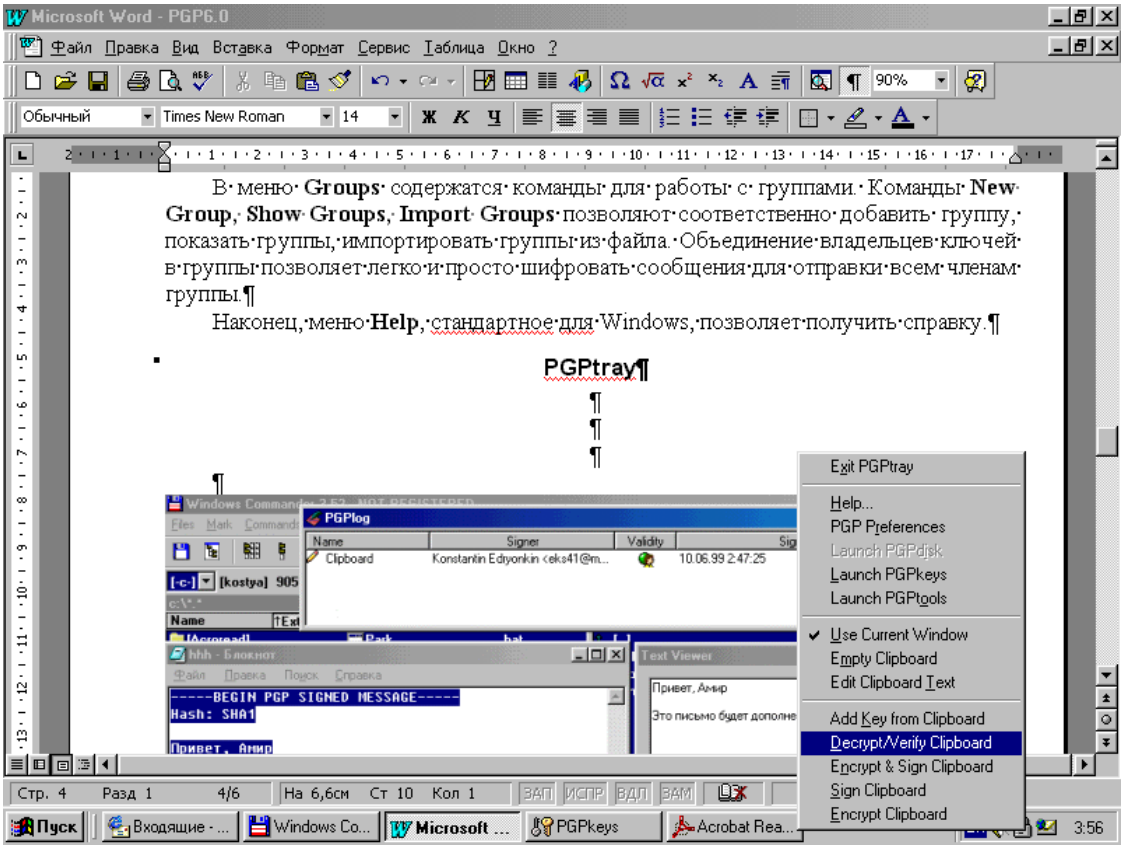



Рис. 8.8. Запуск PGP с использованием значка на панели задач

В меню содержатся команды для выхода, запуска других компонентов *PGP*, редактирования буфера обмена, работы с буфером обмена (добавление ключа, расшифрование/проверка подписи, зашифрование и подпись, просто подпись и просто зашифрование).

Если в используемом почтовом клиенте нет встроенных команд *PGP*, то можно работать с помощью *PGPtray*. Например, для подписи письма достаточно скопировать его в буфер и исполнить команду **Sign Clipboard**.

После этого достаточно вставить в тело письма уже подписанный текст, рис.8.10.

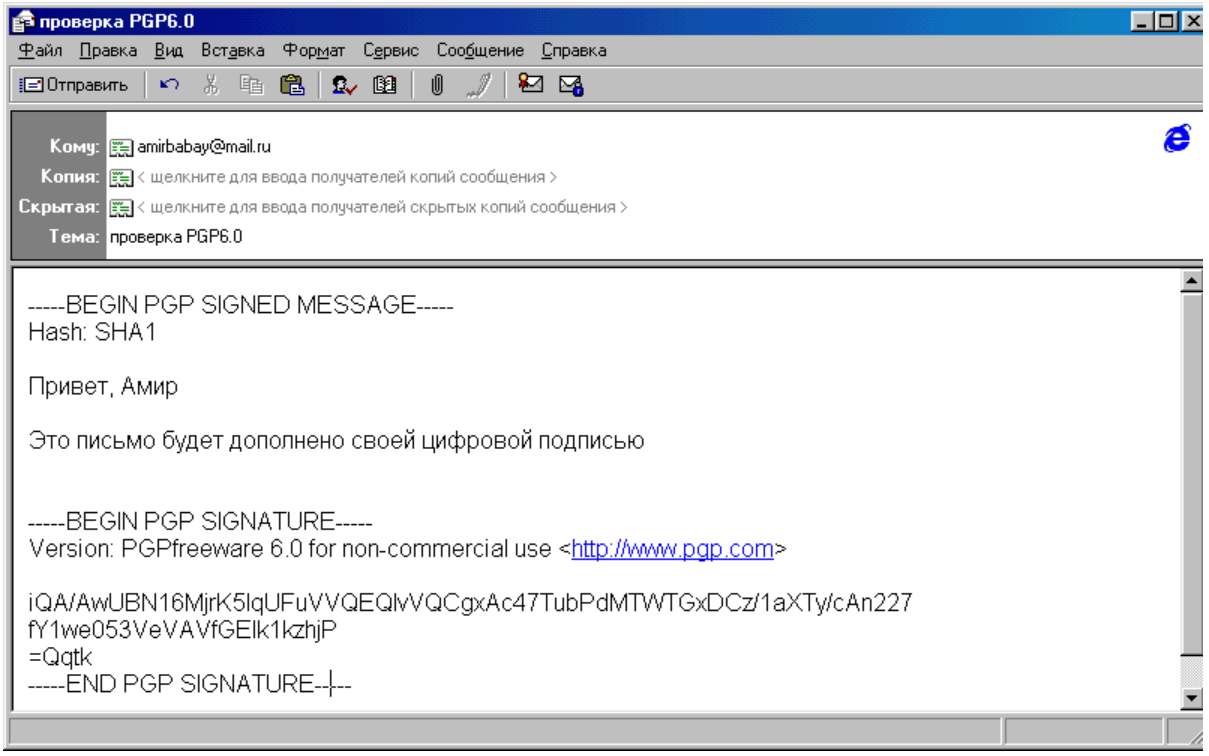



Рис. 8.10. Пример отправки письма с цифровой подписью

Аналогично получателю письма достаточно скопировать полученное письмо в буфер обмена и выбрать команду **Decript/Verify Clipboard**.

Естественно, в коллекции ключей должен быть открытый ключ владельца, подписавшего письмо, рис.8.11, 8.12.

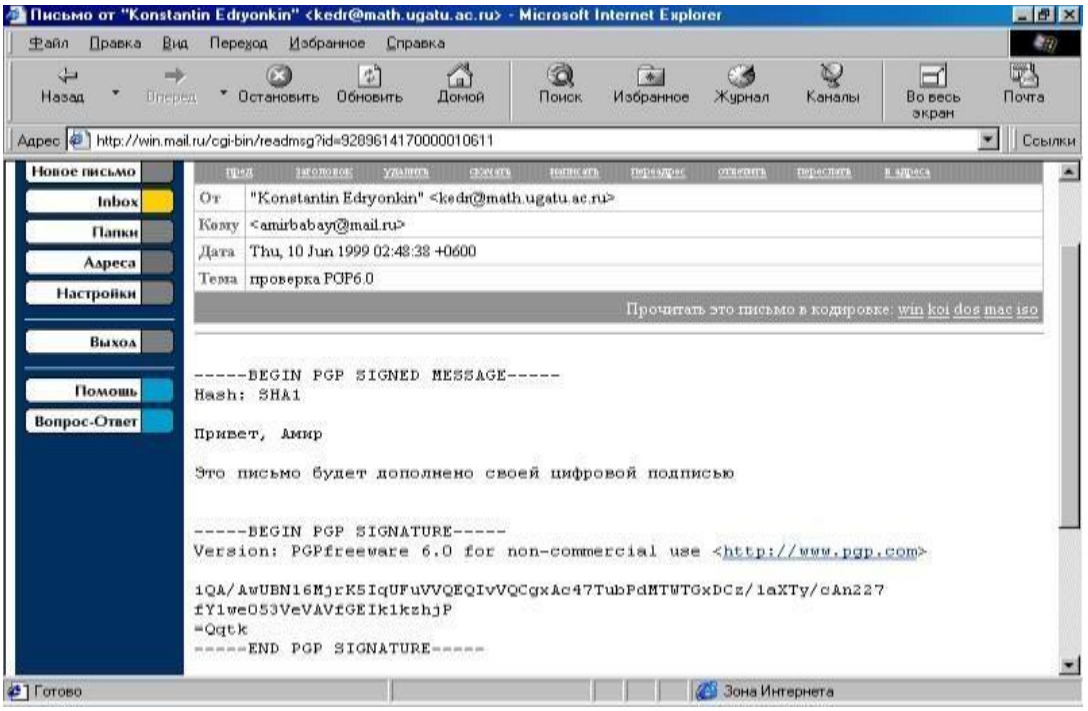



Рис. 8.11. Пример получения письма с цифровой подписью

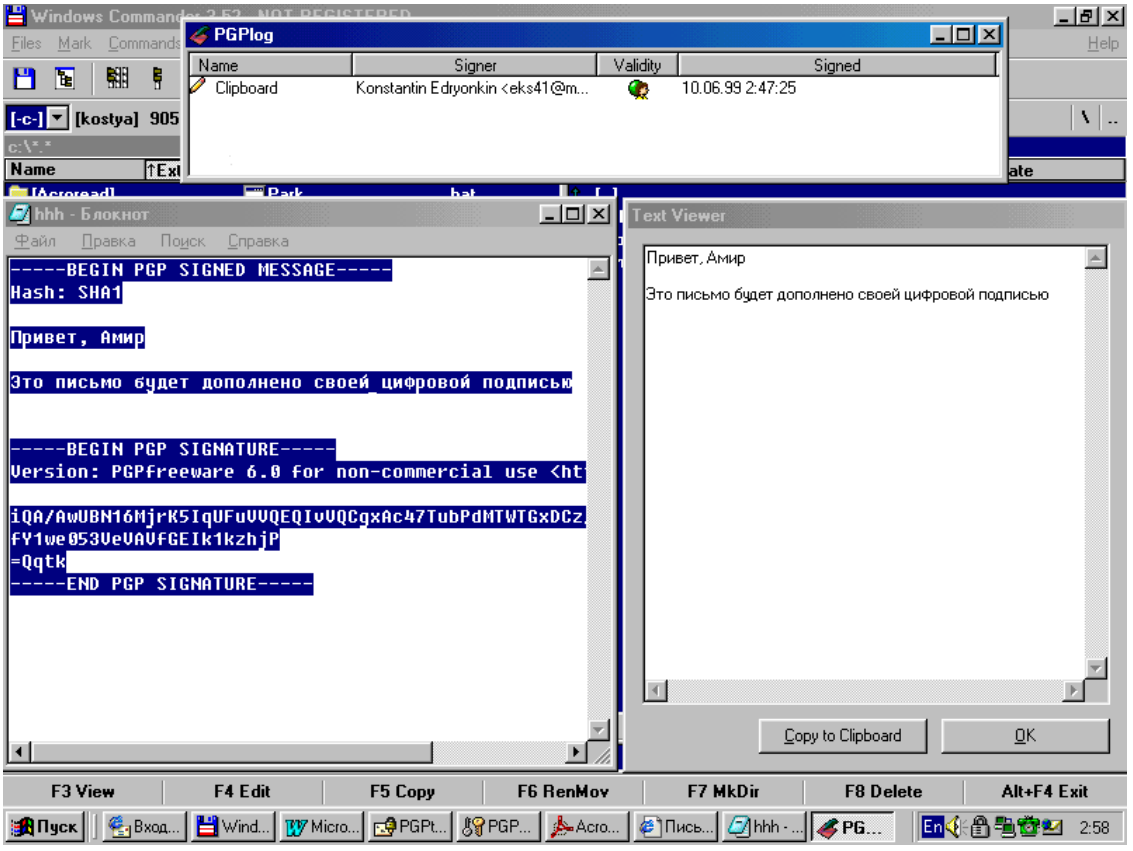



Рис. 8.12. Результаты проверки подписи

**PGPtools**

Этот компонент *PGP* представляет собой небольшую панель с кнопками, позволяющими выполнить нужное действие: открыть *PGPkeys,* зашифровать, подписать, зашифровать и подписать одновременно, расшифровать/проверить подпись, затереть файл, затереть неиспользуемое дисковое пространство, рис.8.13.


Рис. 8.13. Компоненты панели PGPtools

Альтернативой *PGPtools* могут служить команды, добавляющиеся при инсталляции *PGP* в контекстное меню работы с файлами и меню **Файл** Проводника *Windows*.

**Задание**

1. Ознакомиться со сведениями о программе PGP (см. параграф 1).

2. Запустить программу PGPtools (с помощью меню «Пуск» или значка PGPtray на панели задач), ознакомиться и отразить в отчете о лабораторной работе *состав программных средств, входящих в систему PGP* (при необходимости воспользоваться справкой о системе PGP).

3. Создать криптографические ключи с помощью программы PGPkeys. Включить в отчет о лабораторной работе *сведения о порядке создания ключей шифрования в системе PGP* и включить в электронную версию отчета копии используемых при этом экранных форм, а также ответы на вопросы:

- **как обеспечивается случайность выбираемых криптографических ключей в системе PGP**;
- **как и где хранится секретный ключ пользователя в системе PGP**;
- **как может быть обеспечена в системе PGP возможность восстановления секретного ключа пользователя при его случайной утрате**.

4. Изучить (на примере обычных текстовых файлов и файлов изображе-

ний) способы шифрования и расшифрования файлов с помощью функций Encrypt и Decrypt программы PGPtools. Включить в отчет о лабораторной работе *сведения о порядке шифрования и расшифрования файлов в системе PGP,* включить в электронную версию отчета копии используемых при этом экранных форм и ответы на вопросы:

- **какие дополнительные параметры шифрования могут быть использованы, в чем их смысл и возможное применение(обязательно**
- **проверить на примере и результаты проверки отразить в отчете);**
- **как генерируется, как и где хранится ключ симметричного шифрования файла в системе PGP;**
- **как может быть обеспечен доступ к зашифрованному файлу со стороны других пользователей**;
- **изменяется ли и как размер файла после его шифрования (привести конкретные примеры для разных типов файлов)**.

5. Изучить (на примере документов из своей папки) способы получения и проверки электронной цифровой подписи под файлами с помощью функций Sign и Verify программы PGPtools. Включить в отчет *сведения о порядке обеспечения аутентичности и целостности электронных документов в системе PGP,* включить в электронную версию отчета копии используемых при этом экранных форм и ответы на вопросы:

- **какие дополнительные параметры получения электронной цифровой подписи могут быть использованы, в чем их смысл и возможное применение (обязательно проверить на примере и результаты проверки отразить в отчете)**;
- **какова реакция программы на нарушение целостности подписанного документа (обязательно проверить на примере и результаты проверки отразить в отчете)**.

6. Изучить способы одновременного шифрования (расшифрования) и по-

лучения (проверки) электронной цифровой подписи в системе PGP с помощью функций Encrypt Sign и Decrypt/Verify программы PGPtools. Включить в отчет *сведения о порядке одновременного обеспечения конфиденциальности, аутентичности и целостности электронных документов в этой системе,* а также включить в электронную версию отчета копии используемых при этом экранных форм.

7. Изучить способы надежного удаления файлов с конфиденциальной информацией с помощью функции Wipe программы PGPtools. Включить в отчет *сведения о порядке уничтожения конфиденциальных электронных документов в системе PGP* и включить в электронную версию отчета копии используемых при этом экранных форм.

8. Изучить способы надежного уничтожения остаточной информации, которая может содержать конфиденциальные сведения, с помощью функции Freespace Wipe программы PGPtools. Включить в отчет *сведения о назначении и порядке использования этой функции программы,* копии используемых в ней экранных форм и ответы на вопросы:

- **как достигается надежное уничтожение остаточной конфиденциальной информации в системе PGP**;
- **является ли подобный метод уничтожения абсолютно надежным и, если нет, как может быть обеспечено абсолютно надежное уничтожение остаточной информации**

9. Изучить способы быстрого выполнения функций системы PGP с помощью программы PGPtray, ярлык которой размещен в правой части панели задач. Включить в отчет *сведения о назначении и порядке использования этой программы,* а также включить в электронную версию отчета копии используемых экранных форм.

10. Изучить способы управления настройками системы PGP при ее использовании в организациях с помощью программы PGPadmin (пройти все шаги диалога с мастером вплоть до последнего, на котором вместо кнопки «Save»

нажать кнопку «Отмена»). Включить в отчет *сведения о возможностях и порядке администрирования системы PGP*, включить в электронную версию отчета копии используемых при этом экранных форм и ответы на вопросы:

- **какие функции по управлению шифрованием и обеспечением целостности информационных ресурсов предоставляет администратору программа PGPAdmin**;
- **какие функции по управлению криптографическими ключами пользователей PGP предоставляет администратору программа PGPadmin**;
- **какие возможности предоставляет программа PGPadmin по управлению доступными для пользователей функциями программы PGP и где сохраняется подобная информация**.

11. Включить в отчет о лабораторной работе ответы на контрольные вопросы, выбранные в соответствии с номером варианта, указанным преподавателем.

| **Номер варианта** | **Контрольные вопросы** |
|---|---|
| 1,5,7 | Как выбрать длину криптографического ключа в системе PGP? |
| 2,4,6,8 | В чем достоинства и недостатки криптографических методов защиты информации? |
| 11,13,15, 17,19 | Какие компьютерные системы называются безопасными? |
| 12,14,16 | В чем заключаются основные требования к защищенности компьютерных систем? |
| 20,22,24, 30 | Для выполнения каких требований к защищенности компьютерных систем могут применяться криптографические методы защиты? |
| 21,23,25 | Насколько, на Ваш взгляд, надежны методы криптографической защиты информации, используемые в программе PGP? |

| | |
|---|---|
| 3,9,18, 28 | Какими основными функциями защиты информации обладает программа PGP? |
| 10,26,27, 29 | Какой принцип лежит в основе функции надежного уничтожения остаточной конфиденциальной информации программы PGP? |

**Список литературы**


1. Аграновский А. В., Хади Р. А. Практическая криптография. Алгоритмы и их программирование (+ CD-ROM). СПб, СОЛОН-Пресс, 2002.

2. Вельшенбах М. Криптография на Си и C++ в действии (+CD-ROM). – М.: Триумф, 2004.

3. Башлы П.Н., Бабаш А.В., Баранова Е.К. Информационная безопасность: учебно- практическое пособие. – М.: Изд. Центр ЕАОИ, 2010.

4. Коутинхо С. Введение в теорию чисел. Алгоритм RSA. – М.: Постмаркет, 2001.

5. Мао В. Современная криптография: теория и практика. – М.: Издательский дом «Вильямс», 2005.

6. Молдовян А. А., Молдовян Н. А., Советов Б. Я. Криптография. Учебник для вузов – М.: Лань, 2005.

7. Фомичев В.М. Дискретная математика и криптология. Курс лекций. – М.: Диалог- МИФИ, 2003.

# ПРАКТИЧЕСКАЯ РАБОТА № 9
# КОРРЕКТИРУЮЩИЕ КОДЫ. КОДЫ ХЕММИНГА

**Цель работы:** Ознакомление с общими принципами построения и использования корректирующих кодов для контроля целостности информации, распространяемой по телекоммуникационным каналам.

*Примечание*. Для выполнения лабораторной работы на компьютере необходимо установить файл

*Hemming.exe*, который находится в архиве *Код Хэмминга.rar*.

## 1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

### 1.1. Способы контроля правильности передачи данных

#### 1.1.1. Код с проверкой на четность

Контроль целостности информации при передачи от источника к приемнику может осуществляться с использованием корректирующих кодов.

Простейший корректирующий код – *код с проверкой на четность*, который образуется добавлением к группе информационных разрядов одного избыточного, значение которого выбирается таким образом, чтобы сумма единиц в кодовой комбинации (то есть вес кодовой комбинации) была всегда четна.

*Пример.* Рассмотрим код с проверкой на четность, образованный добавлением контрольного разряда к простому двоичному коду.

| | *Информационный* | *Контрольный* |
|---|---|---|
| 0 | 000 | 0 |
| 1 | 001 | 1 |
| 2 | 010 | 1 |
| 3 | 011 | 0 |
| 4 | 100 | 1 |
| 5 | 101 | 0 |
| 6 | 110 | 0 |
| 7 | 111 | 1 |

Такой код обнаруживает все одиночные ошибки и групповые ошибки нечетной кратности, так как четность количества единиц в этом случае будет также нарушаться.

Следует отметить, что при кодировании целесообразно число единиц в кодовой комбинации делать нечетным, и осуществлять контроль на нечетность, в этом случае любая комбинация, в том числе и изображающая нуль, будет иметь хотя бы одну единицу, что дает возможность отличить полное отсутствие информации от передачи нуля.

### 1.1.2. Коды Хэмминга

*N* - значный код Хэмминга имеет *m* - информационных разрядов и *k* - контрольных. Число контрольных разрядов должно удовлетворять соотношению:

$$k >= \log_2 (n+1),$$

откуда $$m <= n - \log_2 (n+1).$$

Код Хэмминга строится таким образом, что к имеющимся информационным разрядам кодовой комбинации добавляется вычисленное по вышеприведенной формуле количество контрольных разрядов, которые формируются путем подсчета четности суммы единиц для определенных групп информационных разрядов. При приеме такой кодовой комбинации из полученных информационных и контрольных разрядов путем аналогичных подсчетов четности составляют корректирующее число, которое равно нулю, при отсутствии ошибки, либо указывает номер ошибочного разряда.

Рассмотрим подробнее процесс кодирования.

Пусть первый контрольный разряд имеет нечетный порядковый номер, установим его при кодировании таким образом, чтобы сумма единиц всех разрядов с нечетными порядковыми номерами была равна 0. Такая операция может быть записана в виде соотношения:

$$E_1 = a_1 \square a_3 \square a_5 \square a_7 \ldots = 0,$$

где $a_1, a_3, a_5, a_7$ - двоичные символы, размещенные в разрядах с номерами 1, 3, 5, 7,…

Появление единицы во втором разряде (справа) корректирующего числа означает ошибку в тех разрядах кодовой комбинации, порядковые номера которых ( 2, 3, 6, 7, …) в двоичном изображении имеют единицу во втором справа разряде. Поэтому вторая операция кодирования, позволяющая найти второй контрольный разряд, имеет вид:

$$E_2 = a_2 \square a_3 \square a_6 \square a_7 \ldots = 0,$$

Рассуждая аналогичным образом:

$$E_3 = a_4 \square a_5 \square a_6 \square a_7 \ldots = 0, \; E_4 = a_8 \square a_9 \square a_{10} \square a_{11} \ldots = 0,$$

После приема кодовой комбинации, совместно со сформированными контрольными разрядами, выполняются те же операции подсчета, что были описаны выше, а полученное число:

$$E_k E_{k-1} \ldots E_2 E_1$$

считается корректирующим, причем при $E_k E_{k-1} \ldots E_2 E_1 = 0$ ошибки отсутствуют, а при наличии ошибок неравными нулю оказываются те суммы $E_i$ , в образовании которых участвовал ошибочный разряд.

Корректирующее число при этом будет равно порядковому номеру этого разряда.

Выбор места для контрольных разрядов в каждой из кодовых комбинаций определяется таким образом, чтобы контрольные разряды участвовали только в одной операции подсчета четности. Это упрощает процесс кодирования, такими позициями являются целые степени двойки: 1, 2, 4, 8, 16,… .

*Пример.* Составим шестизначный код Хэмминга

$n = 6, k >= log_2 7 , k = 3, \qquad m = n - k = 3$

| Цифра | Простой двоичный код | Код Хэмминга | | | | |
|---|---|---|---|---|---|---|
| | | *6* | *5* | *4* | | *3* |
| | | *2* | *1* | | | |
| 0 | 000 | 0 | 0 | 0 | 0 | 0 |
| | | 0 | | | | |
| 1 | 001 | 0 | 0 | 0 | 1 | 1 |
| | | 1 | | | | |
| 2 | 010 | 0 | 1 | 1 | 0 | 0 |
| | | 1 | | | | |
| 3 | 011 | 0 | 1 | 1 | 1 | 1 |
| | | 0 | | | | |
| 4 | 100 | 1 | 0 | 1 | 0 | 1 |
| | | 0 | | | | |
| 5 | 101 | 1 | 0 | 1 | 1 | 0 |
| | | 1 | | | | |
| 6 | 110 | 1 | 1 | 0 | 0 | 1 |
| | | 1 | | | | |
| **7** | 111 | 1 | 1 | 0 | 1 | 0 |
| | | 0 | | | | |

## Варианты исправления ошибок

*Принят код:* 111100 *исправлено* 110100 - *ошибка по корректирующему числу в разряде 4;*

111010 *исправлено* 101010 - *ошибка по корректирующему числу в разряде 5;*

100000 *исправлено* 000000 - *ошибка по корректирующему числу в разряде 6*

### 1.1.3. Циклические коды

Циклические коды являются разновидностью систематических кодов и поэтому обладают всеми их свойствами. Характерной особенностью циклического кода, определяющей его название, является то, что если $n$-значная кодовая комбинация $a_0\ a_1\ a_2\ \ldots\ a_{n-1}\ a_n$ принадлежит данному коду, то и комбинация $a_n\ a_0\ a_1\ a_2\ \ldots\ a_{n-1}$, полученная циклической перестановкой знаков, также принадлежит этому коду.

Идея построения циклических кодов базируется на использовании *неприводимых многочленов*. Неприводимым называется многочлен, который не может быть представлен в виде произведения многочленов низших степеней, то есть такой многочлен, который делится только на самого себя или на 1.

Неприводимые многочлены при построении циклических кодов играют роль так называемых образующих полиномов, от вида которых, собственно и зависят основные характеристики полученного кода: избыточность и корректирующая способность. В таблице 1.1 указаны неприводимые многочлены со степенями *k=1, 2, 3, 4.*

Таблица 1.1

| *P(x)* | | *P(1,0)* |
|---|---|---|
| *k=1* | $x+1$ | 11 |
| *k=2* | $x^2+x+1$ | 111 |
| *k=3* | $x^3+x+1$ | 1011 |
| $x^3+x^2+1$ | | 1101 |
| *k=4* | $x^4+x+1$ | 10011 |
| $x^4+x^3+1$ | | 11001 |
| $x^4+x^3+x^2+x+1$ | | 11111 |

Основные принципы кодирования в циклическом коде заключаются в следующем. Двоично-кодированное n-разрядное число представляется полиномом (n–1)-й степени некоторой переменной x, причем коэффициентами полинома являются двоичные знаки соответствующих разрядов. Запись, чтение и передача кодовых комбинаций в циклическом коде производятся, начиная со старшего разряда. В соответствии с этим правилом, в дальнейшем сами числа и соответствующие им полиномы будем записывать так, чтобы старший разряд оказывался справа.

0 1 2 3 4 5

Пример. Число 110101 (нумерация разрядов согласно выше приведенному правилу, ведется слева направо от 0 до 5) будет представлено полиномом пятой степени: 1 + x + x3 + x5.

Следует отметить, что циклическая перестановка разрядов в двоичном представлении числа соответствует умножению полинома на x, при котором xn заменяется 1 и переходит в начало полинома.

Пример. Произведем умножение полинома, полученного в предыдущем примере, на x. Новый полином x + x2 + x4 + x6 , преобразуем, заменив x6 на 1. Окончательно получим 1 + x + x2 + x4 , что соответствует числу 111010.

Циклический код n-значного числа, как всякий систематический код, состоит их m информационных и k контрольных знаков, причем последние занимают k младших разрядов. Так как последовательная передача кодовых комбинаций производится, как мы уже указывали, начиная со старших разрядов, контрольные знаки передаются в конце кода.

Как уже указывалось, образование кода производится при помощи так называемого порождающего полинома, P(x), степени k, видом которого определяются основные свойства кода – избыточность и корректирующая способность.

Кодовым полиномом, F(x), является полином степени, меньшей (m+k), если он делится без остатка на порождающий полином P(x). После передачи сообщения, декодирование состоит в выполнении деления полинома H(x), соответствующего принятому коду, на P(x). При отсутствии ошибок H(x)=F(x), и деление выполняется без остатка. Наличие ненулевого остатка указывает на то, что при передаче или хранении произошли искажения информации.

Для получения систематического циклического кода используется следующее соотношение:

$F(x) = x^k G(x) \oplus R(x),$

где $G(x)$ – полином, представляющий информационные символы *(информационный полином);*

$R(x)$ – остаток от деления $x^k G(x)$ на $P(x)$.

Пример. Рассмотрим кодирование 8-значного числа 10110111. Пусть для кодирования задан порождающий полином 3-й степени P(x) = 1 + x + x3.

Делим $x^3 G(x)$ на $P(x)$: $G(x) = 1 + x^2 + x^3 + x^5 + x^6 + x^7$;
$x^3 G(x) = x^3 + x^5 + x^6 + x^8 + x^9 + x^{10}$;

$x^{10} + x^9 + x^8 + x^6 + x^5 + x^3 x^{10} + x^8 + x^7$ | $1 + x + x^3$

$x^7 + x^6 + x^2$

$x^9 + x^7 + x^6 + x^5 + x^3$

$x^9 + x^7 + x^6$

$x^5 + x^3$

$x^5 + x^3 + x^2$

$R(x) = x^2$

Используя соотношение для получения систематического циклического кода, находим F(x):

$F(x) = (x^3 + x^5 + x^6 + x^8 + x^9 + x^{10}) \oplus x^2.$

Таким образом, окончательно кодовая комбинация, соответствующая F(x), имеет вид

00010110111
001

*контрольные разряды* → 001 10110111

Практически, применяемая процедура кодирования еще более проста, так как нас интересует только остаток, а частное в конечном результате не используется, то можно производить последовательное вычитание по модулю 2 делителя из делимого и полученных разностей до тех пор, пока разность не будет иметь более низкую степень, чем делитель. Эта разность и есть остаток. Такой алгоритм может быть реализован аппаратно при помощи k-разрядного сдвигающего регистра, имеющего обратные связи. Очевидно, что полученный таким способом циклический код будет являться систематическим.

Однако существует и второй вариант получения циклического кода, когда очередная кодовая комбинация получается путем умножения кодовой комбинации C(x) простого n-значного кода на образующий полином P(x).

При втором способе образования циклических кодов информационные и контрольные символы в комбинациях циклического кода не отделены друг от друга, что затрудняет процесс декодирования. Поэтому этот способ кодирования применяется реже, чем первый. Рассмотрим пример кодирования с использованием второго варианта, причем при выполнении операций будем использовать непосредственно записи исходных кодовых комбинаций в двоичном виде.

Пример. Дан порождающий полином вида P(1,0)=1101. Требуется построить циклический код из простого четырехзначного кода вторым способом. В качестве примера для построения используем исходную комбинацию C(1,0)=0011.Операция умножения этой комбинации на образующийполином запишется следующим образом:

```
  0011
  1101
 ------
  0011
⊕ 0011
  0011
 ------
```

0010111 – это и есть циклический код для 0011.

Аналогичным образом получены остальные кодовые комбинации несистематического циклического кода, приведенные в таблице 1.2.

Таблица 1.2

| *Простой четырехсимвольный код C(x)* | *Циклический (7,4) – код C(x) P(x), где P(1,0)=1101* |
|---|---|
| *0000* | *0000000* |
| *0001* | *0001101* |
| *0010* | *0011010* |
| *0011* * | *0010111* * |
| *0100* | *0110100* |
| *0101* | *0111001* |
| *0110* | *0101110* |
| *0111* | *0100011* |
| *1000* | *1101000* |
| *1001* | *1100101* |
| *1010* | *1110010* |
| *1011* | *1111111* |
| *1100* | *1011100* |
| *1101* | *1010001* |
| *1110* | *1000110* |
| *1111* | *1001011* |

Проблема обнаружения различного типа ошибок при помощи циклического кода, как уже указывалось, сводится к нахождению нужного порождающего полинома. В наши задачи не входит рассмотрение этого вопроса, достаточно полно основные принципы подбора порождающих полиномов изложены в учебнике.

## Эффективное кодирование информации

### 1.1.4. Общие положения

Напомним, что *кодирование* – это представление сообщений в форме, удобной для передачи по данному каналу, а *декодирование* – восстановление информации по принятому сигналу. Одним из важнейших вопросов кодирования является повышение его эффективности, то есть путем устранения избыточ-

ности снижение среднего числа символов, требующихся на букву сообщения. Такое кодирование получило название *эффективное кодирование.* Из выше сказанного следует, что эффективное кодирование решает задачу максимального *сжатия информации*.

Учитывая статистические свойства источника информации можно минимизировать среднее число двоичных символов, требующихся для выражения одной буквы сообщения, что при отсутствии помех позволяет уменьшить время передачи сообщения и его объем.

Эффективное кодирование базируется на *основной теореме Шеннона для канала без помех*[4], суть которой сводится к следующему.

---

[3] Вернер М. Основы кодирования. Учебник для ВУЗов. – М.: Техносфера, 2006. – 288 с.

[4] Шеннон К. Математическая теория связи // К.Шеннон. Работы по теории информации и кибернетике: пер.с англ.; под ред. Р.Л.Добрушина и О.Б.Лупанова. – М.: ИЛ, 1963. – 512 с.

***Суть теоремы Шеннона***

**Сообщения, составленные из букв некоторого алфавита, можно закодировать так, что среднее число двоичных символов на букву сколь угодно близко к энтропии источника этих сообщений, но не меньше этой величины**.

Наличие в сообщениях избыточности позволяет ставить вопрос о сжатии данных, то есть о передаче того же количества информации с помощью последовательностей символов меньшей длины – *методы сжатия без потерь.* Для этого используются специальные алгоритмы сжатия, уменьшающие избыточность. Эффект сжатия оценивают *коэффициентом сжатия*

$K = n/q,$

где $n$ – число минимально необходимых символов для передачи сообщения;

$q$ – число символов в сообщении, сжатом данным методом.

Так, при эффективном двоичном кодировании $n$ равно энтропии источника информации.

Наряду с методами сжатия, не уменьшающими количество информации в сообщении, применяются методы сжатия, основанные на потере малосущественной информации – *методы сжатия с потерями.* Заметим, что применение методов сжатия с потерями неприемлемо для некоторых видов информации, например, текстовой или числовой.


Рис. 1.1. Методы сжатия информации

Простейший способ сжатия числовой информации, представленной в коде *ASCII,* заключается в использовании сокращенного кода с четырьмя битами на символ вместо восьми, так как в *ASCII*-кодировке цифры, а также символы *"точка"*, *"запятая"* и *"пробел"* в качестве первого кодового символа имеют нуль, который может быть отброшен.

Среди широко используемых из за своей простоты алгоритмов сжатия наиболее известен *алгоритм RLE (Run Length Encoding),* где вместо передачи цепочки из одинаковых символов передаются символ и значение длины цепочки. Этот метод эффективен при передаче растровых изображений, особенно монохромных, где имеется большое число цепочек из 1 и 0, но малополезен при передаче текста. Алгоритм *RLE* реализован в формате *PCX*

**Пример использования алгоритма RLE**
**Исходный текст (9 байт): 77 77 77 77 11 11 11 01 FF**

В исходной последовательности:
1. Выделяется повторяющаяся последовательность байт
2. Заменяется счетчиком повторов (04) и кодирующим байтом (77)
3. 00 – нет повторений
4. 02 - сколько байт не повторяется
5. Какие байты не повторяются (01 FF)

**Результат сжатия (8 байт): 04 77 03 11 00 02 01 FF**

Недостатками алгоритма *RLE* являются:

- низкая степень сжатия;
- сильная зависимость от количества повторов байт в исходной информационной последовательности.

Достоинство – простота реализации.

Основная сфера использования методов сжатия с потерями –графические, аудио и видео файлы, где имеется возможность, частично пожертвовав качественными характеристиками информации, добиться значительного уменьшения ее объемов.

Наибольшее распространение среди методов сжатия информации без потерь нашли статистические алгоритмы сжатия, учитывающие вероятность появления отдельных символов в информационном потоке. В первую очередь, это алгоритмы Шеннона-Фано и Хаффмена.

### 1.1.5. Алгоритм Шеннона-Фано

Эффективное кодирование методом Шеннона-Фано базируется на основной теореме Шеннона для канала без помех.

*Алгоритм эффективного кодирования по Шеннону-Фано*

- Буквы алфавита сообщений выписываются в таблицу в порядке убывания вероятностей.

- Затем они разделяются на две группы так, чтобы суммы вероятностей в каждой группе были по возможности одинаковы.
- Всем буквам верхней половины в качестве первого символа приписывается 1, а всем нижним – 0.
- Каждая из полученных групп в свою очередь разбивается на две подгруппы с одинаковыми суммарными вероятностями и т. д., процесс повторяется до тех пор, пока в каждой подгруппе останется по одной букве.

*Пример. Рассмотрим алфавит из 7 букв, приписав каждой букве вероятность ее появления в сообщении $p_i$ (существуют специальные таблицы вероятности появления букв русского или латинского алфавитов в сообщениях, например, см. приложение 1.2.1*).

| Буква | Вероятность, $p_i$ | Процесс получения кода | Код Шеннона |
|---|---|---|---|
| A | 1/4 | 1 1 II | 11 |
| E | 1/4 | 1 I 0 | 10 |
| F | 1/8 | 0 1 1 III | 011 |
| C | 1/8 | 0 1 II 0 | 010 |
| B | 1/8 | 0 0 1 III | 001 |
| D | 1/16 | 0 0 0 1 IV | 0001 |
| G | 1/16 | 0 0 0 0 | 0000 |

$$H(\xi) = \sum_{k=1}^{N} p_k \log_2 1/p_k = 2/4 + 2/4 + 3/8 + 3/8 + 3/8 + 4/16 + 4/16 = 2{,}625.$$

Ясно, что при обычном двоичном кодировании, не учитывая статистических характеристик, для представления каждой буквы потребуется три символа. Наибольший эффект сжатия получается в случае, когда вероятности букв представляют собой целочисленные отрицательные степени двойки. Среднее число символов на букву в этом случае точно равно энтропии.

Среднее число символов на букву

$$L = \sum_{i=1}^{N} p_i n_i,$$

где $p_i$ – вероятность появления $i$-го символа алфавита;

$n_i$ – количество символов в кодовой комбинации $i$-го символа алфавита.

Для рассмотренного в таблице примера

$$L=1/4 * 2 +1/4 * 2 +1/8 * 3 +1/8 * 3 +1/8 * 3 +1/16 * 4 +1/16 * 4 = 2,625.$$

Разность величин *(L – H)* – есть *избыточность кода,* а величина

*(L – H)/L – относительная избыточность.*

Избыточность может трактоваться как мера бесполезно совершаемых альтернативных выборов. Так как в практических случаях отдельные знаки никогда не встречаются одинаково часто, то кодирование с постоянной длиной кодовых слов в большинстве случаев избыточно. Несмотря на это, такое кодирование применяется достаточно часто, руководствуясь техническими соображениями.

### 1.1.6. Алгоритм Хаффмена

*Алгоритм эффективного кодирования по Хаффмену*

Буквы алфавита сообщений выписываются в основной столбец таблицы в порядке убывания вероятностей.

- Две последние буквы объединяются в одну вспомогательную букву, которой приписывается суммарная вероятность.
- Вероятности букв, не участвовавших в объединении, и полученная суммарная вероятность снова располагаются в порядке убывания вероятностей, а две последние объединяются, и т. д. до тех пор, пока не получают единственную вспомогательную букву с вероятностью 1.
- Далее для построения кода используется бинарное дерево, в корне ко-

торого располагается буква с вероятностью 1, при ветвлении ветви с большей вероятностью присваивается код 1, а с меньшей – код 0 (возможно левой – 1, а правой – 0).

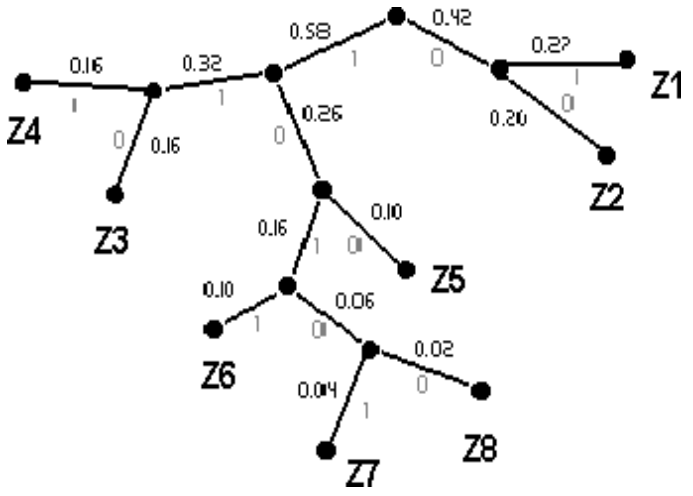
*Пример. Рассмотрим условный алфавит из восьми букв, каждой из которых приписана соответствующая вероятность ее появления в сообщении.*

| Буквы | Вероятности | Вспомогательные столбцы вероятностей | Код Хаффмена |
|---|---|---|---|
| **Z1** | 0,22 | 0,22 0,22 0,26 0,32 0,42 0,58 1 | 01 |
| **Z2** | 0,20 | 0,20 0,20 0,22 0,26 0,32 0,42 | 00 |
| **Z3** | 0,16 | 0,16 0,16 0,20 0,22 0,26 | 110 |
| **Z4** | 0,16 | 0,16 0,16 0,16 0,20 | 111 |
| **Z5** | 0,10 | 0,10 0,16 0,16 | 100 |
| **Z6** | 0,10 | 0,10 0,10 | 1011 |
| **Z7** | 0,04 | 0,06 | 10101 |
| **Z8** | 0,02 | | 10100 |

$L=0,22 \times 2 + 0,20 \times 2 + 0,16 \times 3 + 0,16 \times 3 + 0,10 \times 3 + 0,10 \times 4 + 0,04 \times 5 + 0,02 \times 5 = 2,8;$
$H=2,76;$
$L - H = 0,04.$

*Для сравнения: в коде Шеннона с таким же распределением вероятностей L – H = 0,08.*

*Построение бинарного дерева*




В *приложении 1.2.2* приводятся задания для самостоятельной работы по кодированию информации с использованием алгоритмов Шеннона-Фано и Хаффмена.

*Приложение 1.2.1*

*Вероятности отдельных букв в русском языке*

| Буква | Вероятность | Буква | Вероятность |
|---|---|---|---|
| _ | 0,175 | **Я** | 0,018 |
| **О** | 0,090 | **Ы** | 0,016 |
| **Е** | 0,072 | **З** | 0,016 |
| **А** | 0,062 | **Ъ,Ь** | 0,014 |
| **И** | 0,062 | **Б** | 0,014 |
| **Т** | 0,053 | **Г** | 0,013 |
| **Н** | 0,053 | **Ч** | 0,012 |
| **С** | 0,045 | **Й** | 0,010 |
| **Р** | 0,040 | **Х** | 0,009 |
| **В** | 0,038 | **Ж** | 0,007 |
| **Л** | 0,035 | **Ю** | 0,006 |
| **К** | 0,028 | **Ш** | 0,006 |
| **М** | 0,026 | **Ц** | 0,004 |
| **Д** | 0,025 | **Щ** | 0,003 |
| **П** | 0,023 | **Э** | 0,003 |
| **У** | 0,021 | **Ф** | 0,002 |

*Приложение 1.2.2*

### Задания для самостоятельной работы

*1. Написать процедуру, обеспечивающую эффективное кодирование: вариант а) код Шеннона; вариант б) код Хаффмена, и подсчет среднего количества информации на знак H; избыточности (L - H); и относительной избыточности получаемого кода (L – H) / L.*

*Для тестирования программы использовать два контрольных варианта:*

*Вариант 1*

| Буквы | Вероятности | Код Шеннона |
|---|---|---|
| **Z1** | 0,22 | 11 |
| **Z2** | 0,20 | 101 |
| **Z3** | 0,16 | 100 |
| **Z4** | 0,16 | 01 |
| **Z5** | 0,10 | 001 |
| **Z6** | 0,10 | 0001 |
| **Z7** | 0,04 | 00001 |
| **Z8** | 0,02 | 00000 H = 2,76<br>L = 2,84 L – H = 0,08 |

*Вариант 2*

| **Буквы** | **Вероятности** | **Код Шеннона** |
|---|---|---|
| **Z1** | 1/2 | 1 |
| **Z2** | 1/4 | 01 |
| **Z3** | 1/8 | 001 |
| **Z4** | 1/16 | 0001 |
| **Z5** | 1/32 | 00001 |
| **Z6** | 1/64 | 000001 |
| **Z7** | 1/128 | 0000001 |
| **Z8** | 1/128 | 0000000 H = 1 63/64 L = 1 63/64<br>L - H = 0 |

*2. Источник сообщений порождает знаки A, B, C с вероятностями 0,7; 0,2; 0,1.*

*Написать процедуру позволяющую:*

- *определить энтропию источника;*
- *указать для этих трех знаков оптимальное бинарное кодирование:*

  *вариант а) код Шеннона;*

  *вариант б) код Хаффмена,*

*и определить среднюю длину кодовых комбинаций;*

- *закодировать все пары AA, AB, …;*
- *построить для этих девяти пар оптимальный бинарный код:*

  *вариант а) код Шеннона;*

  *вариант б) код Хаффмена.*

- *увеличить блочность кода до трехсимвольных комбинаций, и построить оптимальный бинарный код:*

  *вариант а) код Шеннона;*

  *вариант б) код Хаффмена.*

*Сделать вывод об изменении избыточности кода с увеличением блочности. Как это влияет на эффективности кода?*

| Буква | Вероятность | Код | Знак | Вероятность | Код пары |
|---|---|---|---|---|---|
| **A** | 0,7 | 0 | **AA** | 0,49 | 1 |
| **B** | 0,2 | 11 | **AB** | 0,14 | 011 |
| **C** | 0,1 | 10 | **BA** | 0,14 | 010 |
| | | | **AC** | 0,07 | 0011 |
| | | | **CA** | 0,07 | 0010 |
| | | | **BB** | 0,04 | 0001 |
| | | | **BC** | 0,02 | 00001 |
| | | | **CB** | 0,02 | 000001 |
| | | | **CC** | 0,01 | 000000 |

*H = 1,1571;* $L_{знак} = 2,33;$

*L = 1,3;* $L_{символ} = L_{знак} / 2 = 1,165.$

*Вывод: при кодировании большими блоками удается уменьшить избыточность кода, теоретически минимум избыточности может быть достигнут при кодировании блоков, включающих бесконечное число букв:*

### 1.2.3. LZW – сжатие

Одним из наиболее широко используемых в настоящее время методов сжатия без потерь является алгоритм Лемпела-Зива-Уэлча *(Lempel-Ziv-Welch).*

Метод сжатия был предложен в 1977 году, усовершенствованный (*Terry Welch*) вариант был опубликован в 1984 году, и в дальнейшем неоднократно модернизировался. Этот алгоритм используется, в частности, стандартной процедурой *UNIX Compress*. Алгоритмы группы *LZ (LZ77, LZ78, LZSS)*, лежат в основе почти всех современных программных и аппаратных средств сжатия информации. *PKZIP, LHA, Stacker, SuperStor, Dblspace* и многие другие архиваторы используют ту или иную модификацию алгоритма *LZ.*

Идея сжатия, предложенная Лемпелом и Зивом заключается в следующем: если в тексте сообщения появляется последовательность из двух ранее уже встречавшихся символов, то эта последовательность объявляется новым символом, для нее назначается код, который при определенных условиях может быть значительно короче исходной последовательности. В дальнейшем, в сжатом сообщении вместо исходной последовательности записывается назначенный код.

При декодировании повторяются аналогичные действия и потому становятся известными последовательности символов для каждого кода.

**Процесс кодирования**

1. Каждому символу исходного алфавита присваивается определенный код (здесь код – порядковый номер, начиная с 0).

2. Выбирается первый символ сообщения и заменяется на его код.

3. Выбираются следующие два символа и заменяются своими кодами. Одновременно этой двухсимвольной комбинации присваивается свой код (обычно это порядковый номер, равный числу уже использованных кодов). Так, если алфавит включает 8 символов, имеющих коды от 000 до 111, то первая двухсимвольная комбинация получит код 1000, следующая – код 1001 и так далее.

4. Выбираются из исходного текста очередные 2, 3, ... N-символьные комбинации до тех пор, пока не образуется еще не встречавшаяся комбинация. Тогда этой комбинации присваивается очередной код, и поскольку совокупность из первых N-1 символов уже встречалась, то она имеет свой код, который и записывается вместо этих N-1 символов. Каждый акт введения нового кода назовем шагом кодирования.

5. Процесс продолжается до исчерпания исходного текста.

**Процесс декодирования**

При *декодировании* код первого символа, а затем второго и третьего заменяются на символы алфавита. При этом становится известным код комбинации второго и третьего символов. В следующей позиции могут быть только коды уже известных символов и их комбинаций. Процесс декодирования продолжается до исчерпания сжатого текста.

*Пример*.

Пусть исходный текст представляет собой двоичный код (первая строка таблицы 2.3), то есть исходный алфавит $A=\{1,0\}$. Коды этих символов соответственно также 0 и 1.

Образующийся по методу Лемпела-Зива *LZ*-код показан во второй строке таблицы 2.3. В третьей строке отмечены шаги кодирования, после которых происходит переход на представление кодов *A* с увеличенным числом разрядов *r*. Так, на первом шаге вводится код 10 для комбинации 00 и поэтому на следующих двух шагах *r*=2, после третьего шага *r*=3, после седьмого шага *r*=4. В общем случае $r=k$ после шага $2^{k-1}-1$.

В приведенном примере *LZ*-код оказался даже длиннее исходного кода, так как обычно короткие тексты с небольшим количеством повторяющихся символов не дают эффекта сжатия. Эффект сжатия проявляется в достаточно длинных текстах и особенно заметен, например, в графических файлах.

Таблица 1.5

| | |
|---|---|
| **Исходный текст** | 0.00.000. 01. 11. 111.1111. 110. 0000.00000. 1101. 1110. |
| ***LZ*-код** | 0.00.100.001.0011.1011.1101. 1010.00110.10010.10001.10110. |
| ***r*** | 2 3 4 |
| **Вводимые коды** | - 10 11 100 101 110 111 1000 1001 1010 1011 1100 |

Усовершенствованный алгоритм сжатия и распаковки по методу Лемпела-Зива-Уэлча приведен на рис. 1.2. На рис. 1.3 приведен пример программной реализации алгоритма *LZW*.

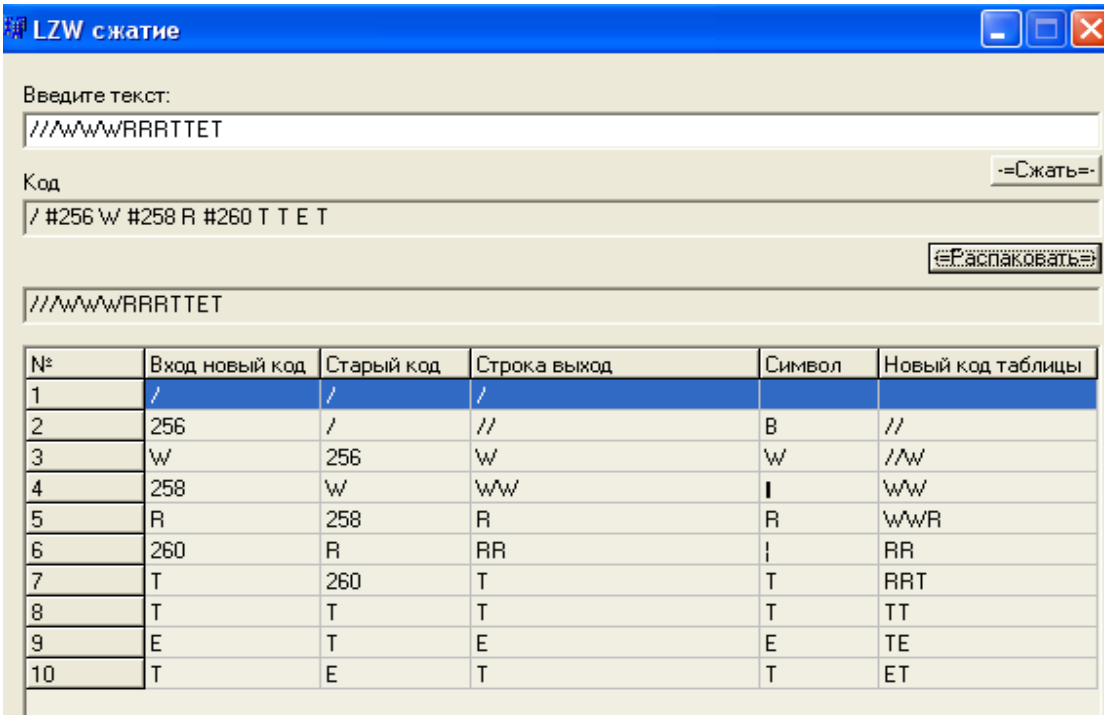
**Алгоритм компрессии**

```
СТРОКА = очередной символ из входного потока
WHILE входной поток  не пуст DO
    СИМВОЛ = очередной символ из входного  потока
    IF СТРОКА + СИМВОЛ в таблице строк THEN
       СТРОКА = СТРОКА + СИМВОЛ
    ELSE
       вывести в выходной поток код для СТРОКА
       добавить в таблицу строк СТРОКА + СИМВОЛ
       СТРОКА = СИМВОЛ
    END of IF
END of WHILE
вывести в выходной поток код для СТРОКА
```

**Алгоритм декомпрессии**

```
Читать СТАРЫЙ_КОД
вывести СТАРЫЙ_КОД
WHILE входной поток не пуст DO
   читать НОВЫЙ_КОД
   СТРОКА = перевести НОВЫЙ_КОД
   вывести СТРОКУ
   СИМВОЛ = первый символ СТРОКИ
   добавить в таблицу  перевода СТАРЫЙ_КОД + СИМВОЛ
   СТАРЫЙ_КОД = НОВЫЙ_КОД
END of WHILE
```

Рис.1.2. Алгоритм *LZW*-компрессии и декомпрессии




| № | Вход новый код | Старый код | Строка выход | Символ | Новый код таблицы |
|---|---|---|---|---|---|
| 1 | / | / | / | | |
| 2 | 256 | / | // | B | // |
| 3 | W | 256 | W | W | //W |
| 4 | 258 | W | WW | I | WW |
| 5 | R | 258 | R | R | WWR |
| 6 | 260 | R | RR | ¦ | RR |
| 7 | T | 260 | T | T | RRT |
| 8 | T | T | T | T | TT |
| 9 | E | T | E | E | TE |
| 10 | T | E | T | T | ET |

Рис.1.3. Пример программной реализации алгоритма

## 1. Описание лабораторной работы

Программа предназначена для кодирования символов по алгоритму Хэмминга. Главное окно программы представлено на рис.9.1.

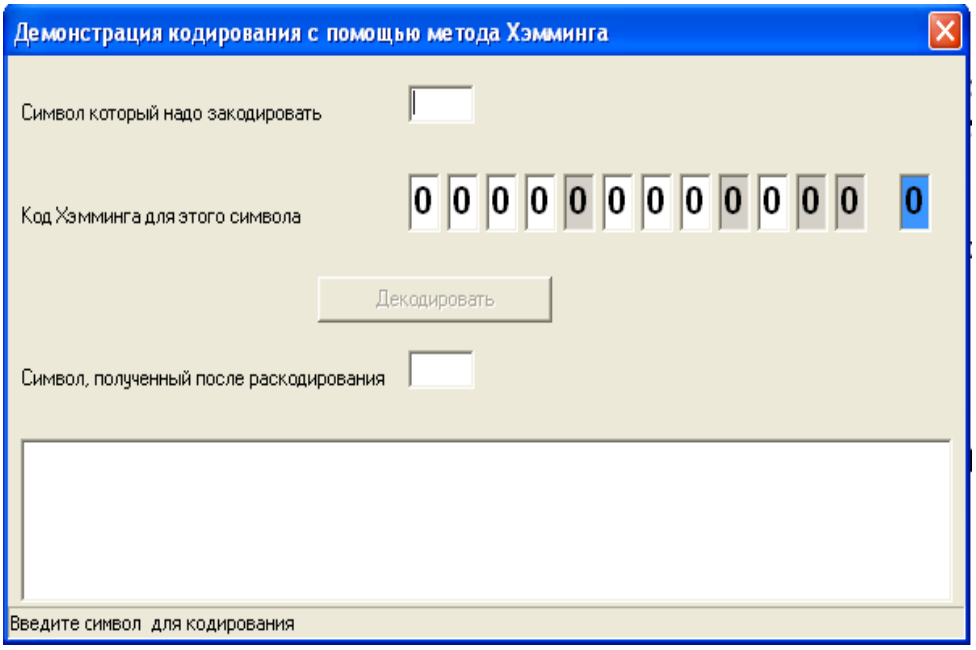



Рис. 9. 1. Главное окно программы

Кодируются любые символы *ASCII*-кодировки, с помощью тринадцатисимвольного кода Хэмминга, содержащего 8 информационных разрядов, 4 контрольных разряда и разряд общей четности.

Процессы кодирования и декодирования представлены на рис.9.2

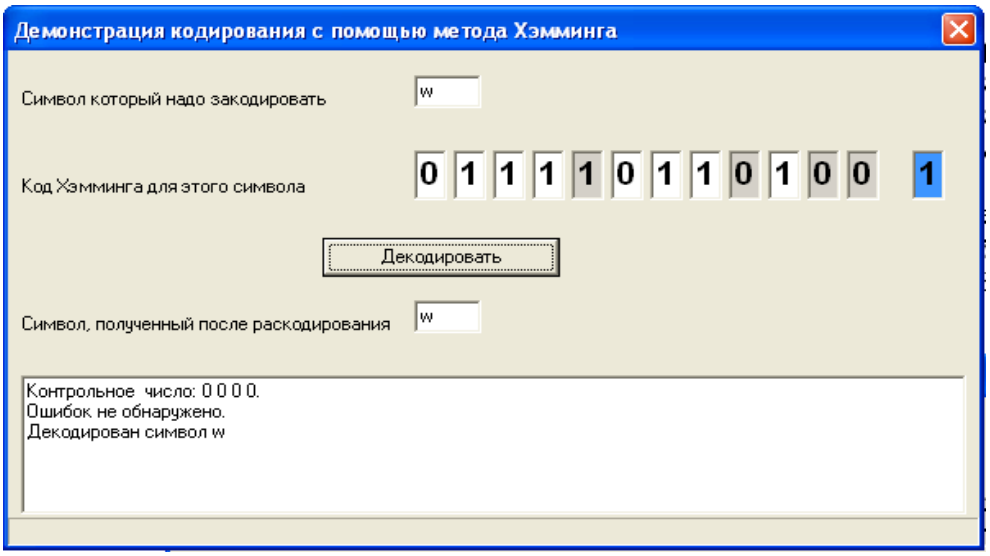



Рис. 9. 2. Процесс кодирования и декодирования

Серым цветом отмечены контрольные биты, голубым – бит общей четности. В полученный код можно вносить ошибки. Одиночные ошибки могут быть исправлены, двойные – обнаруживаются без исправления. Если ошибка была исправлена, то указывается, в каком бите она была допущена. Затем символ декодируется. Процесс исправления одиночной ошибки представлен на рис. 9.3.

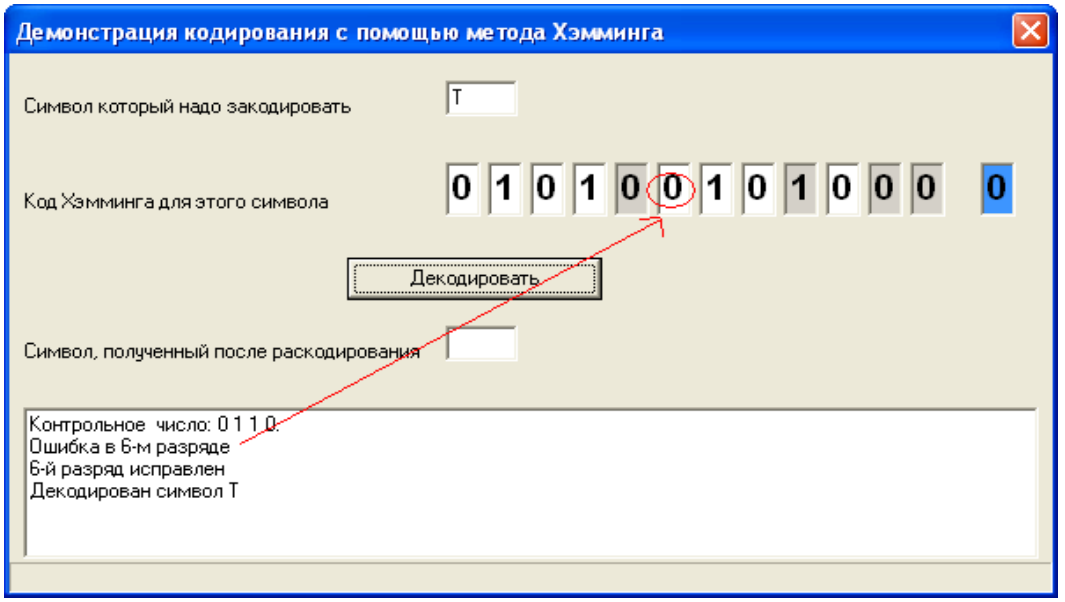



Рис. 9. 3. Процесс исправления одиночной ошибки

Если возникает ошибка двойной или более кратности – выводится сообщение о невозможности исправления кода.

## 2. Задание

***1.*** Закодировать с помощью кода Хэмминга предложенный алфавит (варианты указаны в Таблице 9.1).

***2.*** В каждую строку таблицы с закодированной информацией внести одиночную ошибку, зафиксировать в кодовой таблице результат декодирования.

***3.*** В последние две строки таблицы с закодированной информацией внести двойные ошибки, зафиксировать в кодовой таблице результат декодирования.

***4.*** Проанализировать полученные результаты и сформулировать аргументированные выводы.

Описать полученный код Хэмминга :

- количество контрольных и информационных разрядов и их номера;
- избыточность кода;
- относительная избыточность;
- минимальное кодовое расстояние;
- оценить корректирующую способность полученного кода.

***5.*** Составить из предложенного алфавита слово длиной не менее 5 символов и закодировать его с помощью полученного кода Хэмминга.

Подсчитать длину исходного текста (кодировка *ASCII*) и закодированного текста (код Хэмминга).

***6.*** Представить краткий теоретический материал, в котором описать способ кодирования и декодирования информации с помощью кода Хэмминга.

***7.*** Привести итоги проведенных экспериментов по кодированию с помощью программы *Hemming.exe.*

***8.*** Оценить результаты обнаружения и исправления одиночных и обнаружения двойных ошибок. Сделать выводы о корректирующей способности иссле-

дуемого кода.

***9.*** Привести в отчете ответы на контрольные вопросы, в соответствии с номером варианта, указанным преподавателем.

Варианты заданий представлены в Таблице 9.1

Таблица 9.1

| **Номера вариантов** | **Исходный алфавит** |
|---|---|
| 1,5,9,13,17 | Кириллица А...М |
| 2,6,10,14,18,22 | Кириллица О..Я |
| 3,7,11,15,19,23,27 | Латиница A..N |
| 4,8,12,16,20,24,28 | Латиница O..Z |
| 21,25,29,26,30 | Десятичные цифры 0..9 |

| **Номер варианта** | **Контрольные вопросы** |
|---|---|
| 1,5,7, 3,9,18,28 | Что представляет собой и как используется код Хэмминга? |
| 2,4,6,8, 20,22,24, 26,30 | Как составляется код Хэмминга? |
| 11,13,15, 10,17,19, 27 | Какие ошибки можно исправить с помощью кода Хэмминга? От каких свойств кода зависит его корректирующая способность? |
| 12,14,16 21,23.25, 29 | Какие ошибки могут возникнуть при составлении кода Хэмминга? |

# ПРАКТИЧЕСКАЯ РАБОТА № 10
# КОРРЕКТИРУЮЩИЕ КОДЫ. ЦИКЛИЧЕСКИЕ КОДЫ

**Цель работы:** Ознакомление с общими принципами построения и использования корректирующих кодов для контроля целостности информации, распространяемой по телекоммуникационным каналам. Изучение принципов построения циклических кодов.

*Примечание*. Для выполнения лабораторной работы на компьютере необходимо установить файл
*CyclicCode74.exe,* который находится в архиве *Циклический код.rar.*

## 1. Описание лабораторной работы

1. Изучить сведения о программе кодирования информации с использованием циклического кода (папка *Документация*). Запустить программу кодирования (папка *Исполняемый модуль*, файл *CyclicCode74.exe*), рис. 10.1.

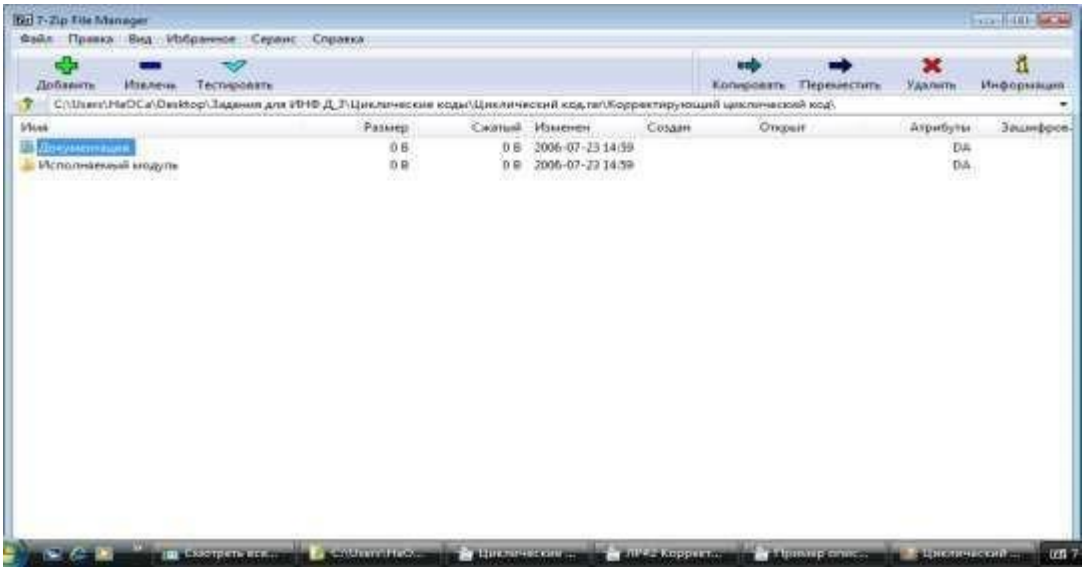


Рис. 10.1. Запуск программы *CyclicCode74*

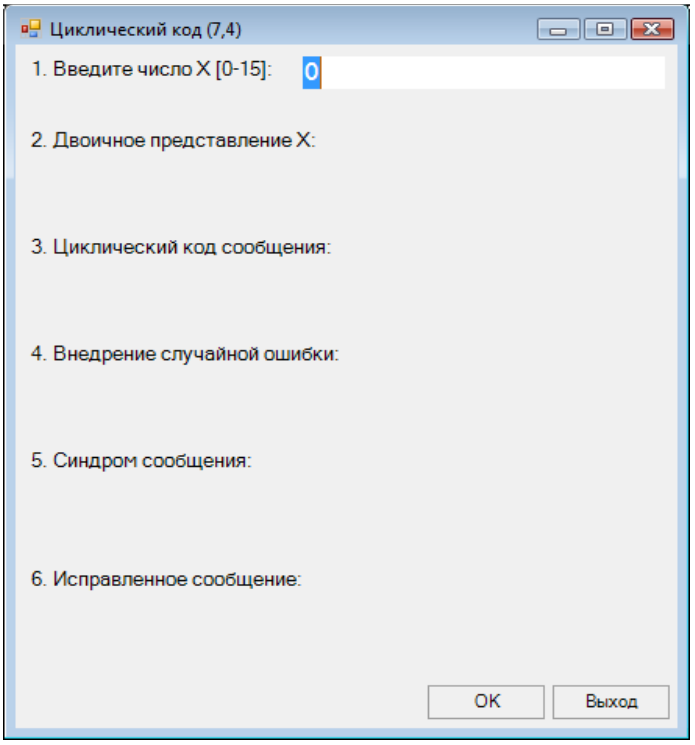



Рис.10. 2. Главное окно программы

2. Закодировать с помощью циклического кода предложенный вариант (*см. варианты заданий*). Для этого в строку *"1. Ведите число X [0-15]"* вводим заданное число. Кодовую таблицу сохранить и перенести в личную папку (для этого нажимаем одновременно Ctrl+Alt+PrtScSysRq затем Ctrl+V).


Рис. 10. 3. Пример заполнения полей главного окна программы

3. Проанализировать полученные результаты и сформулировать аргументированные выводы.

**2. Задание**

1. Получить систематический циклический код, используя приведенные в Таблице 10.1 порождающие полиномы, в соответствии с вариантом, указанным преподавателем:

Таблица 10.1

| *Вариант* | *Порождающий полином 3-ей степени* |
|---|---|
| 1,7,13,19,25 | 1011 |
| 2,8,14,20,26 | 1101 |
| 3,9,15,21,27 | 1011 |
| 4,10,16,22,28 | 1101 |
| 5,11,17,23,29 | 1011 |
| 6,12,18,24,30 | 1101 |

*Пример:* Рассмотрим кодирование восьмизначного числа 00011111. Пусть для кодирования задан порождающий полином 3-ей степени

$P(1,0) = 1101$, $P(x) = 1 + x + x^3$

Делим $x^3\,G(x)$ на $P(x)$, где информационный полином $G(x) = x^3 + x^4 + x^5 + x^6 +$

$x^7$

$x^3\,G(x) = x^6 + x^7 + x^8 + x^9 + x^{10}$

$x^{10} + x^9 + x^8 + x^7 + x^6$ | $x^3 + x + 1$

$x^{10} + x^9 + x^8 + x^7 + x^6$ | $x^7 + x^6 + x^4 + x^2 + x + 1$

$x^9 + x^6$

$x^9 + x^7 + x^6$

$x^7$

$x^7 + x^5 + x^4$

$x^5 + x^4$

$x^5 + x^3 + x^2$

$x^4 + x^3 + x^2$ | $x^4 + x^2 + x$

$x^3 + x$

$$\frac{x^3 + x + 1}{R(x) = 1}$$

Находим значение кодового полинома *F(x):*

$F(x) = ( x^6 + x^7 + x^8 + x^9 + x^{10} ) \square 1$

Таким образом, окончательно кодовая комбинация

0 1 2 3 4 5 6 7 8 9 10

*F(x)* имеет вид : ⇨ 10000011111

Контрольные разряды 100

10000011111

2. Представить в отчете краткий теоретический материал, в котором описать способ кодирования и декодирования информации с помощью циклического кода.

3. Привести итоги проведенных экспериментов по кодированию с помощью программы *CyclicCode74.exe* .

4. Оценить результаты обнаружения и исправления одиночных ошибок. Сделать выводы о корректирующей способности исследуемого кода.

5. Привести в отчете ответы на контрольные вопросы, в соответствии с номером варианта, указанным преподавателем.

Варианты заданий приведены в Таблице 10.2.

Таблица 10.2

| *Вариант* | *Число* |
|---|---|
| 1,16 | 1 |
| 2,17 | 2 |
| 3,18 | 3 |
| 4,19 | 4 |
| 5,20 | 5 |
| 6,21 | 6 |
| 7,22 | 7 |
| 8,23 | 8 |
| 9,24 | 9 |
| 10,25 | 10 |
| 11,26 | 11 |
| 12,27 | 12 |
| 13,28 | 13 |
| 14,29 | 14 |
| 15,30 | 15 |

| **Номер варианта** | **Контрольные вопросы** |
|---|---|
| | |
| 1,5,7, 3,9,18,28 | Укажите различия понятий: *кодирование информации, шифрование информации.* |
| 2,4,6,8, 20,22,24, 26,30 | Определите понятия: *равномерные коды, неравномерные коды, префиксное кодирование, разделимое кодирование, систематические коды, несистематические коды.* |
| 11,13,15, 10,17,19, 27 | От каких характеристик кода зависит его корректирующая способность? |
| 12,14,16 21,23.25, 29 | Сравните коды Хэмминга и циклические коды по эффективности контроля целостности информации. |

# ПРАКТИЧЕСКАЯ РАБОТА № 11
# МЕТОДЫ СЖАТИЯ ПО ШЕННОНУ И ХАФФМЕНУ

**Цель работы:** Ознакомление со статистическими принципами сжатия информации с использованием методов Шеннона-Фано и Хаффмена.

*Примечание*. Для выполнения лабораторной работы на компьютере необходимо установить файл *Shannon-Huffman.exe*, который находится в архиве *Методы сжатия по Шеннону и Хаффмену.rar.*

## 1. Описание лабораторной работы

Работа выполняется на персональном компьютере с использованием программы *Shannon-Huffman.exe.* Программа предназначена для демонстрации методов сжатия информации с использованием алгоритмов Шеннона-Фано и Хаффмена (рис.11.1).

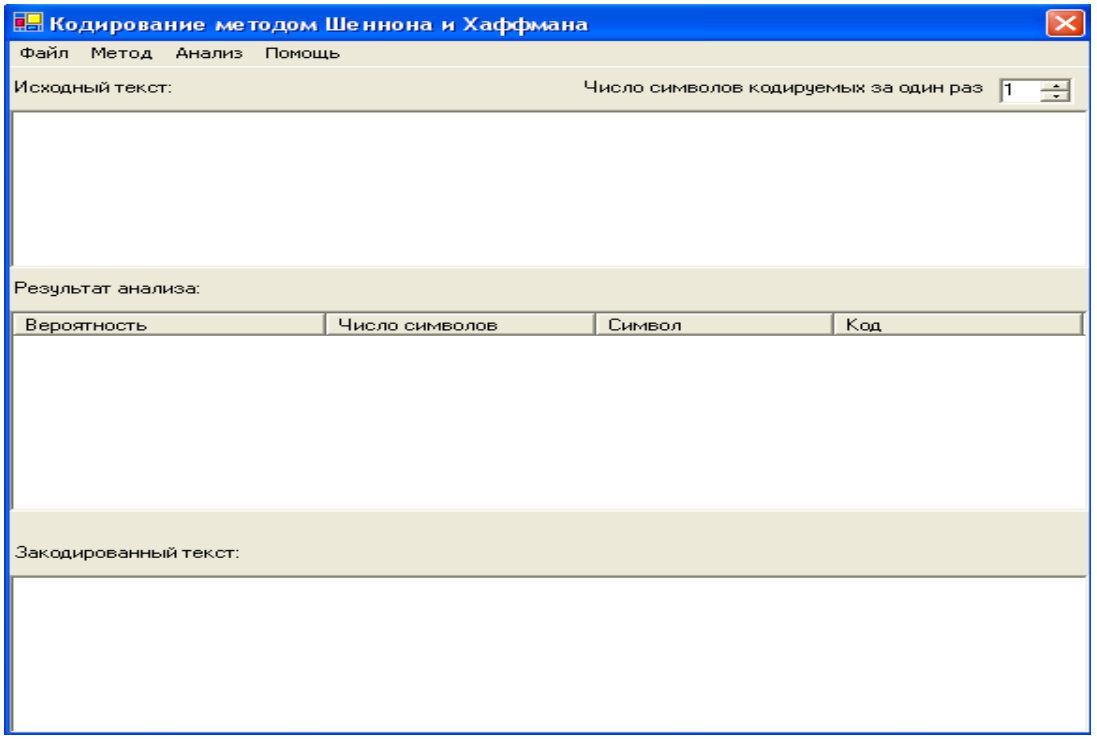



Рис. 11. 1. Главное окно программы

Для работы с программой пользователем выбирается требуемый метод сжатия, рис. 11.2.

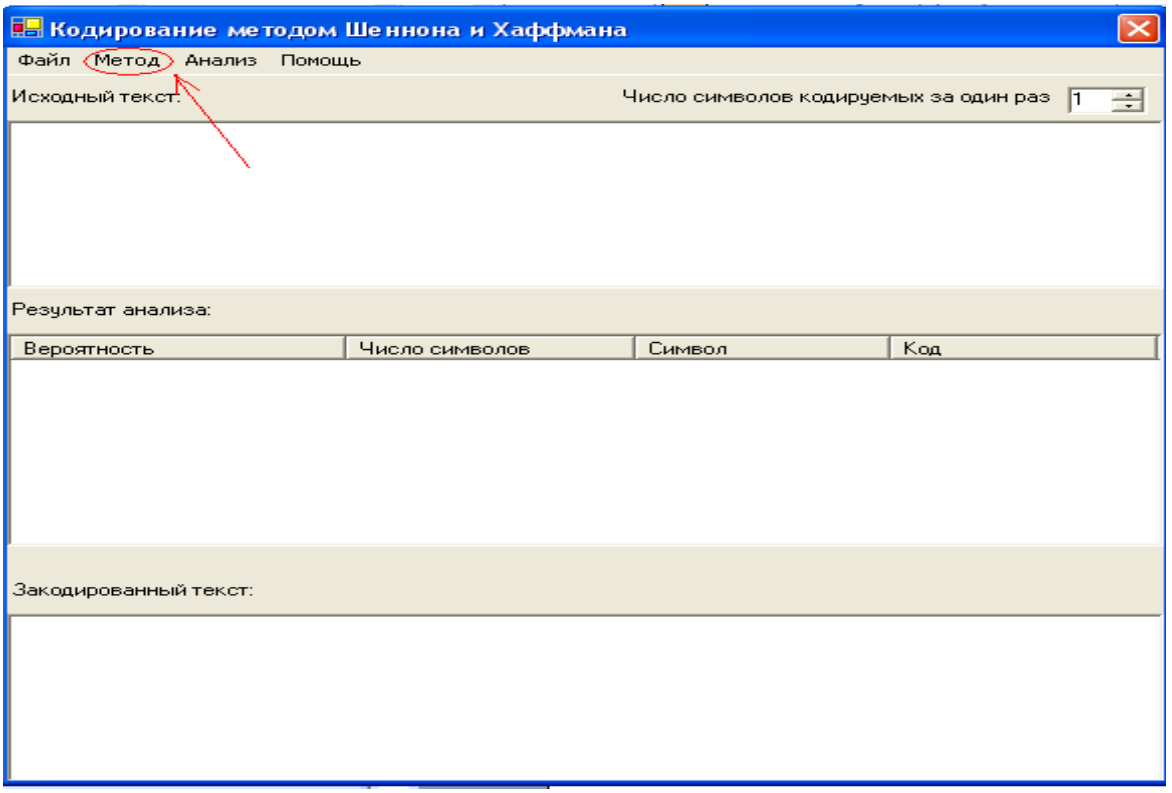
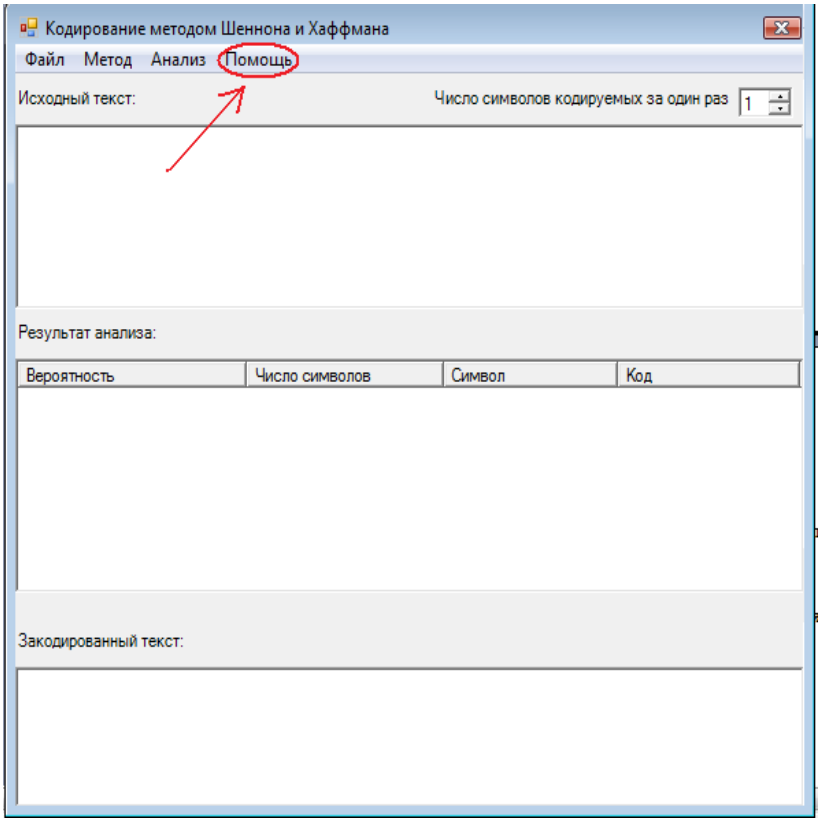



Рис. 11. 2. Выбор метода сжатия

Далее в окне программы *Исходный текст* записывается сообщение произвольной длины (или загружается сообщение из заранее подготовленного файла в формате *.txt*). После этого необходимо указать число символов, кодируемых за один раз.

Для подсчета вероятности появления букв во введенном сообщении *P*, определения энтропии источника сообщений *H*, среднего числа символов при кодировании одной буквы сообщения *L*, необходимо выбрать закладку *Анализ.*




Для определения эффективности кодирования рассчитывается избыточность кода *(L-H)*.

Чтобы выйти из данной программы достаточно выбрать закладку *Файл* и далее *Выход.*

*Пример 1.*

Исходное сообщение *аббввввггггдддддеееее*, состоящее из символов шестибуквенного алфавита

*А = {а, б, в, г, д, е},* закодируем, используя метод Хаффмена. Определяем вероятность появления символа в исходном сообщении

$P = n/N,$

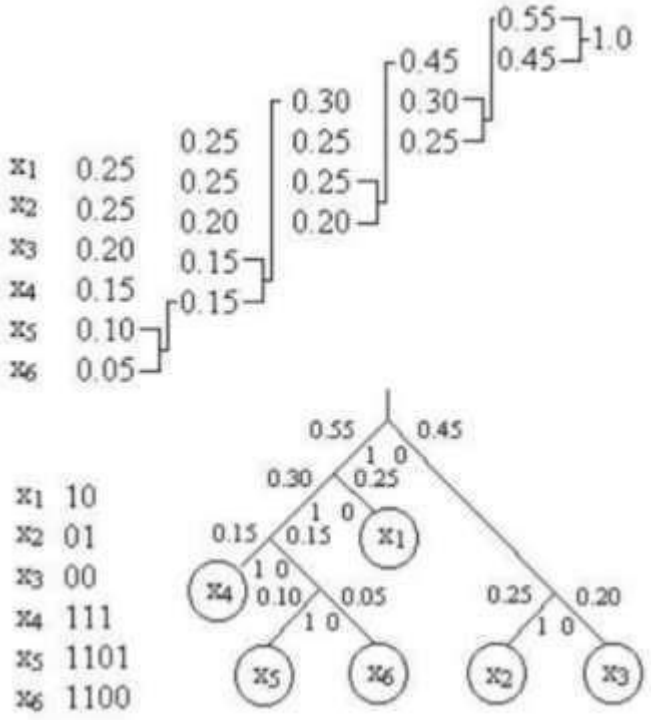
где $n$ – число повторов символа в сообщении, $N$ – длина сообщения.

Выписываем в столбец все символы алфавита в порядке убывания вероятности их появления в тексте (Таблица 11.1).

Таблица 11.1

| символ | число повторений в данном сообщении | вероятность |
|---|---|---|
| *д* | 5 | 0,25 |
| *е* | 5 | 0,25 |
| *г* | 4 | 0,20 |
| *в* | 3 | 0,15 |
| *б* | 2 | 0,10 |
| *а* | 1 | 0,05 |

Последовательно объединяя два символа с наименьшими вероятностями появления в новый составной символ, вероятность появления которого полагается равной сумме вероятностей составляющих его символов, в конце концов мы построим дерево, каждый узел которого имеет суммарную вероятность всех узлов, находящихся ниже него. Прослеживаем путь к каждому листу дерева помечая направление к каждому узлу (например, направо - 0, налево -1).



Среднее количество символов на букву сообщения:

$$L= \sum_{i=1}^{n} P(i)n(i) = 2*2*0.25+2*0.20+3*0.15+4*(0.10+0.05)=2.45,$$

где

$n(i)$ — количество знаков в кодовой комбинации $i$-го символа алфавита

$P(i)$ – вероятность появления $i$-го символа алфавита.

Энтропия:

$$H = \sum_{i=1}^{n} P(i) \log \frac{1}{P(i)} = 2.425$$

Значение избыточности кода (L-H) = 0.025.

Сравниваем полученные данные с результатами работы программы (рис.11.4)

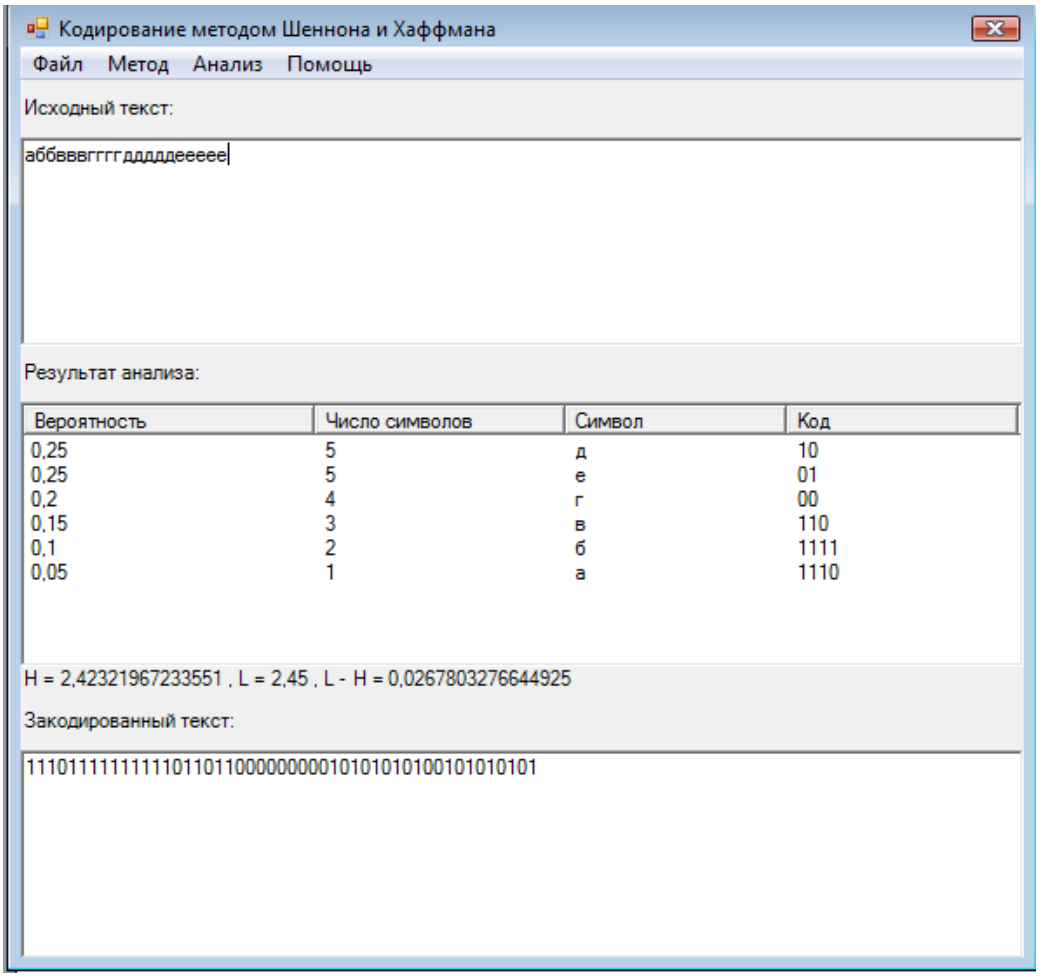



Рис. 11. 4. Пример кодирования

*Пример 2.*

Закодируем исходное сообщение (см. *Пример 1*), но используя метод Шеннона. Выписываем в столбец все символы в порядке убывания вероятности появления их в тексте;

Последовательно делим множество символов на два подмножества так, чтобы сумма вероятностей появления символов одного подмножества была примерно равна сумме вероятностей появления символов другого. Для нижнего подмножества каждому символу приписываем "0", для верхнего – "1". Дальнейшие разбиения повторяются до тех пор, пока все подмножества не будут состоять из одного элемента (Таблица 11.2).

Таблица 11.2

| символ | число повторений в данном сообщении | вероятность | Код Шеннона | | |
|---|---|---|---|---|---|
| *д* | 5 | 0,25 | 1 | 1 | |
| *е* | 5 | 0,25 | 1 | 0 | |
| *г* | 4 | 0,20 | 0 | 1 | 1 |
| *в* | 3 | 0,15 | 0 | 1 | 0 |
| *б* | 2 | 0,10 | 0 | 0 | 1 |
| *а* | 1 | 0,05 | 0 | 0 | 0 |

*Средняя длина полученного кода будет равна*

Среднее количество символов на букву сообщения:

$$L=\sum_{i=1}^{n} n(i)*P(i)=0.25*2+0.25*2+0.2*3+0.15*3+0.1*3+0.25*3=2.5$$

Энтропия: $$H=\sum_{i=1}^{n} P(i)\log\frac{1}{P(i)}=2.425$$

Сравниваем полученные данные с результатами работы программы (рис.11.5)

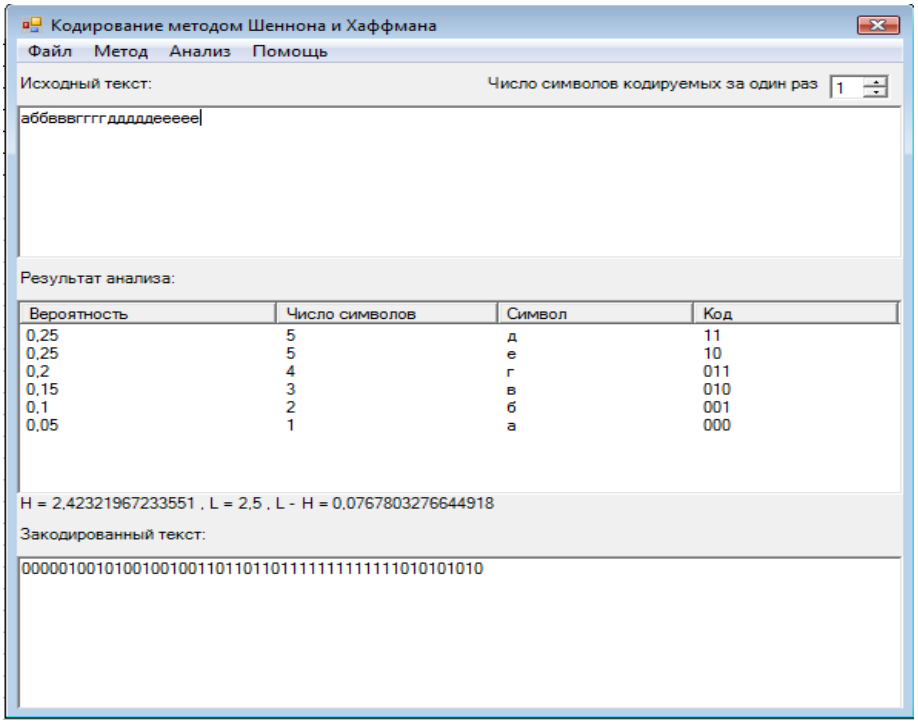



Рис. 11. 5. Пример кодирования

## 2. Задание

Ознакомиться со сведениями о программе *Shannon-Huffman.exe,*и рассмотреть примеры эффективного кодировании информации.

1. Запустить программу *Shannon-Huffman.exe*, предназначенную для демонстрации методов сжатия информации с использованием алгоритмов Шеннона-Фано и Хаффмена.

2. Выполнить предложенные задания:

1). Число символов алфавита $k = m$ ($m$– номер варианта).

Составить такое исходное сообщение, чтобы

a) символы алфавита встречались в сообщении с равными вероятностями;

b) символы алфавита встречались в сообщении с разными вероятностями.

2). Ввести произвольный связный текст на русском языке.

Это может быть пословица, стихотворение или произвольный текст. Используя результаты работы программы, следует проанализировать алфавит введенного сообщения: подсчитать количество символов алфавита, значение энтропии *H*, среднее количество символов на знак *L* при целочисленном кодировании.

3. Сохранить в отчете экранные формы, демонстрирующие процесс сжатия информации.

4. Отчет по лабораторной работе должен содержать результаты анализа программы *Shannon-Huffman.exe* при кодировании; алгоритм построения кода Хаффмена и Шеннона с применением знаний по данной тематике; результаты сравнения различных методов кодирования; выводы об эффективности используемых методов сжатия.

5. Привести в отчете ответы на контрольные вопросы, в соответствии с номером варианта, указанным преподавателем.

Включить в отчет о лабораторной работе выполненные задания 1, 2, 3.

**Варианты заданий**

*Задание1*.

Сообщение состоит из последовательности двух букв A и B, вероятности появления каждой из которых не зависят от того, какая была передана раньше, и равны 0,8 и 0,2 соответственно. Произведите кодирование по методу Шеннона: а) отдельных букв; б) блоков, состоящих из двухбуквенных сочетаний; в) блоков, состоящих из трехбуквенных сочетаний. Сравните полученные коды по их эффективности.

*Задание 2.*

Составьте текст, который бы соответствовал данным, приведенным в Таблице 11.3.

Используя программу *Shannon-Huffman.exe* закодируйте текст методом Хаффмена.

Таблица 11.3

| Номер Варианта | вероятность | символ | число символов |
|---|---|---|---|
| 1,5,7, 3,9,12,14,18,28 | 0,333333333 | о | 2 |
| | 0,166666667 | г | 1 |
| | 0,166666667 | р | 1 |
| | 0,166666667 | д | 1 |
| | 0,166666667 | а | 1 |
| | | | |
| 2,4,6,8,16,21 20,22,24, 30 | 0,25 | е | 2 |
| | 0,25 | т | 2 |
| | 0,125 | о | 1 |
| | 0,125 | п | 1 |
| | 0,125 | р | 1 |
| | 0,125 | а | 1 |
| | | | |
| 11,13,15, 10,17,19,23,25 | 0,25 | р | 2 |
| | 0,25 | а | 2 |
| | 0,25 | с | 2 |
| | 0,125 | т | 1 |
| | 0,125 | е | 1 |

*Задание 3.*

Для вариантов А),Б),В) (см.рис.11.6 и Таблица 4) составьте код Хаффмена. Сделайте подсчет среднего количества символов на знак; избыточности *(L – H)*; и относительной избыточности полученного кода *(L – H) / L*. Сравните полученные значения с *L, H, (L-H)* для кода Шеннона, сделайте выводы.

А)

Результат анализа:

| Вероятность | Число символов | Символ | Код |
|---|---|---|---|
| 0,4375 | 7 | а | 11 |
| 0,1875 | 3 | б | 10 |
| 0,1875 | 3 | г | 01 |
| 0,125 | 2 | в | 001 |
| 0,0625 | 1 | д | 000 |

H = 2,05242128382936 , L = 2,1875 , L - H = 0,135078716170635

Б)

Результат анализа:

| Вероятность | Число символов | Символ | Код |
|---|---|---|---|
| 0,357142857142857 | 5 | к | 11 |
| 0,285714285714286 | 4 | е | 10 |
| 0,142857142857143 | 2 | ж | 011 |
| 0,142857142857143 | 2 | з | 010 |
| 0,0714285714285714 | 1 | и | 00 |

H = 2,1209520310264 , L = 2,28571428571429 , L - H = 0,164762254687883

В)

Результат анализа:

| Вероятность | Число символов | Символ | Код |
|---|---|---|---|
| 0,363636363636364 | 4 | л | 11 |
| 0,272727272727273 | 3 | о | 10 |
| 0,181818181818182 | 2 | м | 01 |
| 0,0909090909090909 | 1 | н | 001 |
| 0,0909090909090909 | 1 | п | 000 |

H = 2,11807820934971 , L = 2,18181818181818 , L - H = 0,0637399724684737

Рис. 11. 6. Варианты заданий

Таблица 11.4

| Номер варианта | Задание |
|---|---|
| 1,5,7, 3,9, 12,14,18, 28 | *Задание 3*. Вариант А). |

| | |
|---|---|
| 2,4,6,8, 20,22,23, 24,25 30 | *Задание 3*. Вариант Б). |
| 11,13,15, 16 21,10,17, 19 | *Задание 3*. Вариант В). |

| **Номер варианта** | **Контрольные вопросы** |
|---|---|
| 1,5,7, 3,9,18,28 | Какие коды позволяют производить однозначное декодирование даже без использования разделительного символа? Приведите примеры таких кодов. |
| 2,4,6,8, 20,22,24, 26,30 | Назовите условия построения оптимальных кодов. |
| 11,13,15, 10,17,19, 27 | С какой целью используются эффективные коды, и какие из них вам известны? |
| 12,14,16 21,23.25, 29 | Перечислите основные методы сжатия информации без потерь. |

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное

учреждениевысшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

# МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению самостоятельной работы

по дисциплине «**ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ В ИНФОКОММУНИКАЦИЯХ**»

для студентов направления подготовки

11.04.02 Инфокоммуникационные технологии и системы связи

Ставрополь, 2026

Методические указания по организации самостоятельной работы студентов составлено в соответствии с требованиями программы дисциплины «Обеспечение информационной безопасности в инфокоммуникациях» для студентов направления подготовки 11.04.02 Инфокоммуникационные технологии и системы связи. Данные указания посвящены планируемой учебной работе студентов, выполняемой во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия, при частичном, непосредственном участии преподавателя.

## СОДЕРЖАНИЕ

## 1. Самостоятельная работа как важнейшая форма учебного процесса

Самостоятельная работа - планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Самостоятельная работа студентов в ВУЗе является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения. Обучение в ВУЗе включает в себя две, практически одинаковые по объему и взаимовлиянию части – процесса обучения и процесса самообучения. Поэтому СРС должна стать эффективной и целенаправленной работой студента.

Концепцией модернизации российского образования определены основные задачи профессионального образования - "подготовка квалифицированного работника соответствующего уровня и профиля, конкурентоспособного на рынке труда, компетентного, ответственного, свободно владеющего своей профессией и ориентированного в смежных областях деятельности, способного к эффективной работе по специальности на уровне мировых стандартов, готового к постоянному профессиональному росту, социальной и профессиональной мобильности".

Решение этих задач невозможно без повышения роли самостоятельной работы студентов над учебным материалом, усиления ответственности преподавателей за развитие навыков самостоятельной работы, за стимулирование профессионального роста студентов, воспитание творческой активности и инициативы.

К современному специалисту общество предъявляет достаточно широкий перечень требований, среди которых немаловажное значение имеет наличие у выпускников определенных способностей и умения самостоятельно добывать знания из различных источников, систематизировать полученную информацию, давать оценку конкретной финансовой ситуации. Формирование такого умения происходит в течение всего периода обучения через участие студентов в лабораторных занятиях, выполнение контрольных заданий и тестов, написание курсовых и выпускных квалификационных работ. При этом самостоятельная работа студентов играет решающую роль в ходе всего учебного процесса.

Формы самостоятельной работы студентов разнообразны. По данной дисциплине «**ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ В ИНФОКОММУНИКАЦИЯХ**» предуссмотрены следующие:

### Использование материала учебно-методического комплекса дисциплины

На первом этапе необходимо ознакомиться с обязательным минимумом содержания основной образовательной программы по изучаемой дисциплине. Далее необходимо изучить рабочую программу дисциплины, в которой рассмотрено содержание тем дисциплины лекционного курса, взаимосвязь тем лекций с лабораторных занятий, темы и виды самостоятельной работы. По каждому виду самостоятельной работы предусмотрены определённые формы отчетности, представленные в учебной программе по дисциплине «**ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ В ИНФОКОММУНИКАЦИЯХ**».

### Работа с литературой

Для успешного освоения дисциплины, необходимо самостоятельно детально изучить представленные темы по рекомендуемым источникам информации, которые представлены в учебной программе.

Самостоятельная работа приобщает студентов к научному творчеству, поиску и решению актуальных современных проблем.

## 2. Цели и основные задачи СРС

Ведущая цель организации и осуществления СРС должна совпадать с целью обучения студента – подготовкой магистра. При организации СРС важным и необходимым условием становятся формирование умения самостоятельной работы для приобретения знаний, навыков и возможности организации учебной и научной деятельности.

Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности. Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами СРС являются:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;

- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений;
- использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на лабораторных занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

## 3. Виды самостоятельной работы

В образовательном процессе высшего профессионального образовательного учреждения выделяется два вида самостоятельной работы – аудиторная, под руководством преподавателя, и внеаудиторная. Тесная взаимосвязь этих видов работ предусматривает дифференциацию и эффективность результатов ее выполнения и зависит от организации, содержания, логики учебного процесса (межпредметных связей, перспективных знаний и др.):

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа выполняется студентом по заданию преподавателя, но без его непосредственного участия.

Основными видами самостоятельной работы студентов без участия преподавателя по данной дисциплине являются:

- формирование и усвоение содержания конспекта лекций на базе рекомендованной лектором учебной литературы, включая информационные образовательные ресурсы (электронные учебники, электронные библиотеки и др.);
- подготовка к лабораторным работам, их оформление.

Основными видами самостоятельной работы студентов с участием преподавателей являются:

- текущие консультации;
- прием и защита лабораторных работ (во время проведения л/р).

## 4. Организация СРС

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей дисциплины «Проектирование траектории профессионального роста и личностного развития», объема часов на ее изучение, вида заданий для самостоятельной работы студентов, индивидуальных качеств студентов и условий учебной деятельности.

Процесс организации самостоятельной работы студентов включает в себя следующие этапы:

- подготовительный (определение целей, составление программы, подготовка методического обеспечения, подготовка оборудования);
- основной (реализация программы, использование приемов поиска информации, усвоения, переработки, применения, передачи знаний, фиксирование результатов, самоорганизация процесса работы);
- заключительный (оценка значимости и анализ результатов, их систематизация, оценка эффективности программы и приемов работы, выводы о направлениях оптимизации труда).

### Деятельность студентов по формированию и развитию навыков учебной самостоятельной работы

В процессе самостоятельной работы студент приобретает навыки самоорганизации, самоконтроля, самоуправления, саморефлексии и становится активным самостоятельным субъектом учебной деятельности.

*Выполняя самостоятельную работу под контролем преподавателя студент должен:*

–освоить минимум содержания, выносимый на самостоятельную работу студентов и предложенный преподавателем в соответствии с образовательными стандартами высшего образования;

– планировать самостоятельную работу в соответствии с графиком самостоятельной работы, предложенным преподавателем;

–самостоятельную работу студент должен осуществлять в организационных формах, предусмотренных учебным планом и рабочей программой преподавателя;

– выполнять самостоятельную работу и отчитываться по ее результатам в соответствии с графиком представления результатов, видами и сроками отчетности по самостоятельной работе студентов.

*студент может:*

сверх предложенного преподавателем (при обосновании и согласовании с ним) и минимума обязательного содержания, определяемого программой по данной дисциплине:

–самостоятельно определять уровень (глубину) проработки содержания материала;

– предлагать темы и вопросы для самостоятельной проработки;

– в рамках общего графика выполнения самостоятельной работы предлагать обоснованный индивидуальный график выполнения и отчетности по результатам самостоятельной работы;

– предлагать свои варианты организационных форм самостоятельной работы;

– использовать для самостоятельной работы методические пособия, учебные пособия, разработки сверх предложенного преподавателем перечня;

– использовать не только контроль, но и самоконтроль результатов самостоятельной работы в соответствии с методами самоконтроля, предложенными преподавателем или выбранными самостоятельно.

Самостоятельная работа студентов должна оказывать важное влияние на формирование личности будущего магистра, она планируется студентом самостоятельно. Каждый студент самостоятельно определяет режим своей работы и меру труда, затрачиваемого на овладение учебным содержанием по каждой дисциплине. Он выполняет внеаудиторную работу по личному индивидуальному плану, в зависимости от его подготовки, времени и других условий.

## 5. Общие рекомендации по организации самостоятельной работы

Основной формой самостоятельной работы студента является изучение конспекта лекций, их дополнение, рекомендованной литературы, активное участие на практических занятиях. Но для успешной учебной деятельности, ее интенсификации, необходимо учитывать следующие субъективные факторы:

1. Знание школьного программного материала, наличие прочной системы зияний, необходимой для усвоения основных вузовских курсов. Это особенно важно для математических дисциплин. Необходимо отличать пробелы в знаниях, затрудняющие усвоение нового материала, от малых способностей. Затра-

тив силы на преодоление этих пробелов, студент обеспечит себе нормальную успеваемость и поверит в свои способности.

2. Наличие умений, навыков умственного труда:

а) умение конспектировать на лекции и при работе с книгой;

б) владение логическими операциями: сравнение, анализ, синтез, обобщение, определение понятий, правила систематизации и классификации.

3. Специфика познавательных психических процессов: внимание, память, речь, наблюдательность, интеллект и мышление. Слабое развитие каждого из них становится серьезным препятствием в учебе.

4. Хорошая работоспособность, которая обеспечивается нормальным физическим состоянием. Ведь серьезное учение - это большой многосторонний и разнообразный труд. Результат обучения оценивается не количеством сообщаемой информации, а качеством ее усвоения, умением ее использовать и развитием у себя способности к дальнейшему самостоятельному образованию.

5. Соответствие избранной деятельности, профессии индивидуальным способностям. Необходимо выработать у себя умение саморегулировать свое эмоциональное состояние и устранять обстоятельства, нарушающие деловой настрой, мешающие намеченной работе.

6. Овладение оптимальным стилем работы, обеспечивающим успех в деятельности. Чередование труда и пауз в работе, периоды отдыха, индивидуально обоснованная норма продолжительности сна, предпочтение вечерних или утренних занятий, стрессоустойчивость на экзаменах и особенности подготовки к ним,

7. Уровень требований к себе, определяемый сложившейся самооценкой.

Адекватная оценка знаний, достоинств, недостатков - важная составляющая самоорганизации человека, без нее невозможна успешная работа по управлению своим поведением, деятельностью.

Одна из основных особенностей обучения в высшей школе заключается в том, что постоянный внешний контроль заменяется самоконтролем, активная роль в обучении принадлежит уже не столько преподавателю, сколько студенту.

Зная основные методы научной организации умственного труда, можно при наименьших затратах времени, средств и трудовых усилий достичь наилучших результатов.

Эффективность усвоения поступающей информации зависит от работоспособности человека в тот или иной момент его деятельности.

*Работоспособность* - способность человека к труду с высокой степенью напряженности в течение определенного времени. Различают внутренние и внешние факторы работоспособности.

К внутренним факторам работоспособности относятся интеллектуальные особенности, воля, состояние здоровья.

К внешним:

- организация рабочего места, режим труда и отдыха;
- уровень организации труда - умение получить справку и пользоваться информацией;
- величина умственной нагрузки.

Выдающийся русский физиолог Н. Е. Введенский выделил следующие условия продуктивности умственной деятельности:

- во всякий труд нужно входить постепенно;
- мерность и ритм работы. Разным людям присущ более или менее разный темп работы;
- привычная последовательность и систематичность деятельности;
- правильное чередование труда и отдыха.

Отдых не предполагает обязательного полного бездействия со стороны человека, он может быть достигнут простой переменой дела. В течение дня работоспособность изменяется. Наиболее плодотворным является *утреннее время (с 8 до 14 часов),* причем максимальная работоспособность приходится на период с 10 до 13 часов, затем *послеобеденное* - (с 16 до 19 часов) и *вечернее* (с 20 до 24 часов). Очень трудный для понимания материал лучше изучать в начале каждого отрезка времени (лучше всего утреннего) после хорошего отдыха. Через 1-1,5 часа нужны перерывы по 10 - 15 мин, через 3 - 4 часа работы отдых должен быть продолжительным - около часа.

Составной частью научной организации умственного труда является овладение техникой умственного труда.

Время, которым располагает студент для выполнения учебного плана, складывается из двух составляющих: одна из них - это аудиторная работа в вузе по расписанию занятий, другая - внеаудиторная самостоятельная работа. Задания и материалы для самостоятельной работы выдаются во время учебных занятий по расписанию, на этих же занятиях преподаватель осуществляет контроль за самостоятельной работой, а также оказывает помощь студентам по правильной организации работы.

Самостоятельные занятия потребуют интенсивного умственного труда, который необходимо не только правильно организовать, но и стимулировать. При этом очень важно уметь поддерживать устойчивое внимание к изучаемому материалу. Выработка внимания требует значительных волевых усилий. Именно поэтому, если студент замечает, что он часто отвлекается во время самостоятельных занятий, ему надо заставить себя сосредоточиться. Подобную процедуру необходимо проделывать постоянно, так как это является тренировкой внимания. Устойчивое внимание появляется тогда, когда человек относится к делу с интересом.

Следует правильно организовать свои занятия по времени: 50 минут - работа, 5-10 минут - перерыв; после 3 часов работы перерыв - 20-25 минут. Иначе нарастающее утомление повлечет неустойчивость внимания. Очень существенным фактором, влияющим на повышение умственной работоспособности, являются систематические занятия физической культурой. Организация активного отдыха предусматривает чередование умственной и физической деятельности, что полностью восстанавливает работоспособность человека.

## 6. Самостоятельная работа студента - необходимое звено становления исследователя

Прогресс науки и техники, информационных технологий приводит к значительному увеличению научной информации, что предъявляет более высокие требования не только к моральным, нравственным свойствам человека, но и в особенности, постоянно возрастающие требования в области образования – обновление, модернизация общих и профессиональных знаний, умений специалиста.

Всякое образование должно выступать как динамический процесс, присущий человеку и продолжающийся всю его жизнь. Овладение научной мыслью и языком науки является необходимой составляющей в самоорганизации будущего специалиста исследователя. Под этим понимается не столько накопление знаний, сколько овладение научно обоснованными способами их приобретения. В этом, вообще говоря, состоит основная задача вуза.

Специфика вузовского учебного процесса, в организации которого самостоятельной работе студента отводятся все больше места, состоит в том, что он является как будто бы последним и самым адекватным звеном для реализации этой задачи. Ибо во время учебы в вузе происходит выработка стиля, навыков

учебной (познавательной) деятельности, рациональный характер которых будет способствовать постоянному обновлению знаний высококвалифицированного выпускника вуза.

Однако до этом пути существуют определенные трудности, в частности, переход студента от синтетического процесса обучения в средней школе, к аналитическому в высшей. Это связано как с новым содержанием обучения (расширение общего образования и углубление профессиональной подготовки), так и с новыми, неизвестными до сих пор формами: обучения (лекции, семинары, лабораторные занятия и т.д.). Студент получает не только знания, предусмотренные программой и учебными пособиями, но он также должен познакомиться со способами приобретения знаний так, чтобы суметь оценить, что мы знаем, откуда мы это знаем и как этого знания мы достигли. Ко всему этому приходят через собственную самостоятельную работу.

Это и потому, что самостоятельно приобретенные знания являются более оперативными, они становятся личной собственностью, а также мотивом поведения, развивают интеллектуальные черты, внимание, наблюдательность, критичность, умение оценивать. Роль преподавателя в основном заключается в руководстве накопления знаний (по отношению к первокурсникам), а в последующие годы учебы, на старших курсах, в совместном установлении проблем и заботе о самостоятельных поисках студента, а также контролирования за их деятельностью. Отметим, что нельзя ограничиваться только приобретением знаний предусмотренных программой изучаемой дисциплины, надо постоянно углублять полученные знания, сосредотачивая их на какой-нибудь узкой определенной области, соответствующей интересам студента. Углубленное изучение всех предметов, предусмотренных программой, на практике является возможным, и хорошая организация работы позволяет экономить время, что создает условия для глубокого, систематического, заинтересованного изучения самостоятельно выбранной студентом темы.

Конечно, все советы, примеры, рекомендации в этой области, даваемые преподавателем, или определенными публикациями, или другими источниками, не гарантируют никакого успеха без проявления собственной активности в этом деле, т. е. они не дают готовых рецептов, а должны способствовать анализу собственной работы, ее целей, организации в соответствии с индивидуальными особенностями. Учитывая личные возможности, существующие условия жизни и работы, навыки, на основе этих рекомендаций, возможно, выработать индивидуально обоснованную совокупность методов, способов, найти свой стиль или усовершенствовать его, чтобы изучив определенный материал, иметь

время оценить его значимость, пригодность и возможности его применения, чтобы, в конечном счете, обеспечить успешность своей учебе с будущей профессиональной деятельности

## 7. Методические рекомендации для студентов по отдельным формам самостоятельной работы

Система вузовского обучения подразумевает значительно большую самостоятельность студентов в планировании и организации своей деятельности.

**Работа с литературой**

При работе с литературой необходимо подобрать литературу, научиться правильно ее читать, вести записи. Для подбора литературы в библиотеке используются алфавитный и систематический каталоги.

Важно помнить, что рациональные навыки работы с книгой - это всегда большая экономия времени и сил.

Правильный подбор учебников рекомендуется преподавателем, читающим лекционный курс. Необходимая литература может быть также указана в методических разработках по данному курсу.

Изучая материал по учебнику, следует переходить к следующему вопросу только после правильного уяснения предыдущего, описывая на бумаге все выкладки и вычисления (в том числе те, которые в учебнике опущены или на лекции даны для самостоятельного вывода).

При изучении любой дисциплины большую и важную роль играет самостоятельная индивидуальная работа.

Особое внимание следует обратить на определение основных понятий курса. Студент должен подробно разбирать примеры, которые поясняют такие определения, и уметь строить аналогичные примеры самостоятельно. Нужно добиваться точного представления о том, что изучаешь. Полезно составлять опорные конспекты. При изучении материала по учебнику полезно в тетради (на специально отведенных полях) дополнять конспект лекций. Там же следует отмечать вопросы, выделенные студентом для консультации с преподавателем.

Выводы, полученные в результате изучения, рекомендуется в конспекте выделять, чтобы они при перечитывании записей лучше запоминались.

Опыт показывает, что многим студентам помогает составление листа опорных сигналов, содержащего важнейшие и наиболее часто употребляемые формулы и понятия. Такой лист помогает запомнить формулы, основные положения лекции, а также может служить постоянным справочником для студента.

Различают два вида чтения; первичное и вторичное. *Первичное* - эти внимательное, неторопливое чтение, при котором можно остановиться на трудных местах. После него не должно остаться ни одного непонятного олова. Содержание не всегда может быть понятно после первичного чтения.

Задача *вторичного* чтения полное усвоение смысла целого (по счету это чтение может быть и не вторым, а третьим или четвертым).

Чтение научного текста является частью познавательной деятельности. Ее цель – извлечение из текста необходимой информации. От того на сколько осознанна читающим собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия.

Выделяют ***четыре основные установки в чтении научного текста***:

1. информационно-поисковый (задача – найти, выделить искомую информацию)
2. усваивающая (усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить как сами сведения излагаемые автором, так и всю логику его рассуждений)
3. аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему)
4. творческая (создает у читателя готовность в том или ином виде – как отправной пункт для своих рассуждений, как образ для действия по аналогии и т.п. – использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

С наличием различных установок обращения к научному тексту связано существование и нескольких ***видов чтения***:

1. библиографическое – просматривание карточек каталога, рекомендательных списков, сводных списков журналов и статей за год и т.п.;
2. просмотровое – используется для поиска материалов, содержащих нужную информацию, обычно к нему прибегают сразу после работы со списками литературы и каталогами, в результате такого просмотра читатель устанавливает, какие из источников будут использованы в дальнейшей работе;
3. ознакомительное – подразумевает сплошное, достаточно подробное прочтение отобранных статей, глав, отдельных страниц, цель – познакомиться с характером информации, узнать, какие вопросы вынесены автором на рассмотрение, провести сортировку материала;

4. изучающее – предполагает доскональное освоение материала; в ходе такого чтения проявляется доверие читателя к автору, готовность принять изложенную информацию, реализуется установка на предельно полное понимание материала;

5. аналитико-критическое и творческое чтение – два вида чтения близкие между собой тем, что участвуют в решении исследовательских задач. Первый из них предполагает направленный критический анализ, как самой информации, так и способов ее получения и подачи автором; второе – поиск тех суждений, фактов, по которым или в связи с которыми, читатель считает нужным высказать собственные мысли.

Из всех рассмотренных видов чтения основным для студентов является изучающее – именно оно позволяет в работе с учебной литературой накапливать знания в различных областях. Вот почему именно этот вид чтения в рамках учебной деятельности должен быть освоен в первую очередь. Кроме того, при овладении данным видом чтения формируются основные приемы, повышающие эффективность работы с научным текстом.

**Основные виды систематизированной записи прочитанного:**

1. Аннотирование – предельно краткое связное описание просмотренной или прочитанной книги (статьи), ее содержания, источников, характера и назначения;
2. Планирование – краткая логическая организация текста, раскрывающая содержание и структуру изучаемого материала;
3. Тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала;
4. Цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора;
5. Конспектирование – краткое и последовательное изложение содержания прочитанного.

Конспект – сложный способ изложения содержания книги или статьи в логической последовательности. Конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

**Методические рекомендации по составлению конспекта:**

1. Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта.

2. Выделите главное, составьте план.

3. Кратко сформулируйте основные положения текста, отметьте аргументацию автора.

4. Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно.

5. Грамотно записывайте цитаты. Цитируя, учитывайте лаконичность, значимость мысли.

В тексте конспекта желательно приводить не только тезисные положения, но и их доказательства. При оформлении конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной последовательности, отвечающей логической структуре произведения. Для уточнения и дополнения необходимо оставлять поля.

Овладение навыками конспектирования требует от студента целеустремленности, повседневной самостоятельной работы.

**Практические занятия**

Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться на практических занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью выполнения заданий лабораторной работы, поставленных задач. При этих условиях студент не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно) для активной проработки лекции.

При самостоятельном выполнении заданий нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение задач или примеров следует излагать по-

дробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что выполнение каждого учебного задания должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Выполнение заданий данного типа нужно продолжать до приобретения твердых навыков в их решении.

**Самопроверка**

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих заданий на практических занятиях и самостоятельно студенту рекомендуется, используя лист опорных материалов, воспроизвести по памяти определения, выводы формул, формулировки основных положений и доказательств.

В случае необходимости нужно еще раз внимательно разобраться в материале.

Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала - умение решать задачи или пройти тестирование по пройденному материалу. Однако следует помнить, что правильное решение задачи может получиться в результате применения механически заученных формул без понимания сущности теоретических положений.

**Консультации**

Если в процессе самостоятельной работы над изучением теоретического материала или при выполнении заданий у студента возникают вопросы, разрешить которые самостоятельно не удается, необходимо обратиться к преподавателю для получения у него разъяснений или указаний. В своих вопросах студент должен четко выразить, в чем он испытывает затруднения, характер этого затруднения. За консультацией следует обращаться и в случае, если возникнут сомнения в правильности ответов на вопросы самопроверки.

**Подготовка к зачету.**

Изучение многих общепрофессиональных и специальных дисциплин завершается зачетом. Подготовка к зачету способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к зачету, студент ликвидирует

имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания.

Для получения зачета студенту необходимо полностью выполнить задания практических работ занятий, подготовить отчет по каждой практической работе и отчет по результатам самостоятельного изучения тем, согласно учебной программе. При наличии задолженностей по текущей аттестации по данной дисциплине студент не получает зачет.

Систематическая подготовка к занятиям в течение семестра позволит использовать время экзаменационной сессии для систематизации знаний.

Текущая аттестация студентов проводится преподавателями, ведущими практические занятия по дисциплине в следующих формах: собеседование при защите практической работы с предоставлением результатов выполнения заданий.

Критерии оценивания собеседования приведены в Фонде оценочных средств по данной дисциплине.

## 8. Учебно-методическое и информационное обеспечение дисциплины

По данной дисциплине преподаватель рекомендует следующую литературу:

| | |
|---|---|
| Основная литература | 1. Чистякова, С. Н. От учебы к профессиональной карьере : учеб. пособие / С. Н. Чистякова, Н. Ф. Родичев. – М. : Академия, 2012. – 171 с. : цв. ил. – (Профессиональная ориентация). – Библиогр.: с. 167. – ISBN 978-5-7695-8399-5<br>2. Ameличкин, О.А. Управление карьерой в коммерческих учреждениях / О.А. Ameличкин. - Москва : Лаборатория книги, 2012. - 112 с. - ISBN 978-5-504-00529-4 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=142538<br>3. Жданко, Т. А. Управление личной карьерой будущего педагога. Практическое руководство для бакалавров и магистров [Электронный ресурс] : учебное пособие по самоменеджменту для бакалавров и магистров направления подготовки 050100 «Педагогическое образование» / Т. А. Жданко. — Электрон. текстовые данные. — Иркутск : Иркутский государственный лингвистический университет, 2013. — 92 с. — 978-5-88267-364-1. — Режим доступа: http://www.iprbookshop.ru/50705.html<br>4. Социально-педагогические основы развития образовательных траекторий личности в системе непрерывного образования : коллективная монография / под науч. ред. Т.Ю. Лома- |

| | |
|---|---|
| | киной ; Российская академия образования, Федеральное государственное научное учреждение, Институт теории и истории педагогики и др. - Москва : Институт эффективных технологий, 2013. - 260 с. - ISBN 978-5-904212-15-5 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=232311 |
| Дополнительная литература | 1. Парнов, Д.А. Кем быть?: Секреты выбора профессии. Книга, с которой начинается карьера / Д.А. Парнов. - Москва : Книжный мир, 2014. - 256 с. : табл. - ISBN 978-5-8041-0695-0 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=274577<br>2. Громова, Е.М. Профессиональная карьера: путь к успеху : научно-методическое пособие / Е.М. Громова, Д.И. Беркутова, Т.А. Горшкова ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Ульяновский государственный педагогический университет имени И.Н. Ульянова». - Ульяновск : УлГПУ, 2012. - 124 с. : ил., табл. - ISBN 978-5-86045-543-6 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=278064 |