

MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN FEDERATION
FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION OF HIGHER EDUCATION
"NORTH CAUCASUS FEDERAL UNIVERSITY"

SUGGESTIONS FOR TEACHING
for the performance of practical work
by discipline
"Principles and Techniques of Information Security"

for students of the field of study
11.03.02 "Infocommunication Technologies and Communication Systems",
direction (profile) "Intelligent Data Processing in
Information and Communication Systems and Networks"

Stavropol, 2026

Table of Contents

Introduction.....	Theoretical	4
information.....	Laboratory work No 1. The	5
Use of Classical Cryptoalgorithms of Substitution and Permutation to Protect		
Textual Information	Laboratory	
Work No 2. Study of Various Methods of Protecting Text Information and Their		21
Strength Based on		
.....	Key Selection Lab No 3.	
Symmetric Encryption Standard <i>Aes Rijndael</i>		32
.....		
Laboratory Work No 4. Study of the structure and principle of operation of the		45
Enigma schiffing machine.....	Laboratory Work	
No 5.	Generation of Prime Numbers Used in	56
Asymmetric Encryption Systems	LaboratoryNo6.	
Electronic Digital Signature.....	Laboratory No 7. Sliding Permutation	70
Encryption		84
Laboratory Work No 8. Study of Software Products. PGP Program		
.....	Laboratory Work No 9.	95
Correction Codes. Hemming Codes.....	Laboratory Work No 10. Correction	
Codes. Cyclic Codes... Laboratory Work No 11. Compression Methods		114
According to Shannon and Huffman.....		128
		153
		158

Introduction

At present, attention to the problem of information security has sharply increased all over the world. This is due to the processes of rapid expansion of information flows that permeate all spheres of society.

Information has long ceased to be a necessary auxiliary resource for production or a by-product of any kind of activity. It has acquired a tangible value weight, which is clearly determined by the real profit received from its use, or the extent of the damage caused to the owner of information with varying degrees of probability. However, the creation of an information processing industry gives rise to a number of complex problems. One of these problems is the reliable preservation and established status of information circulating and processed in information communication systems and networks.

In these guidelines, laboratory work has been developed aimed at studying the principles of construction and parameters of the main units that ensure information security.

1. THEORETICAL INFORMATION

1.1. Problems of information protection

At present, attention to the problem of information security has sharply increased all over the world. This is due to the processes of rapid expansion of information flows that permeate all spheres of society.

Information has long ceased to be a necessary auxiliary resource for production or a by-product of any kind of activity. It has acquired a tangible value weight, which is clearly determined by the real profit received from its use, or the extent of the damage caused to the owner of information with varying degrees of probability. However, the creation of an information processing industry gives rise to a number of complex problems. One of these problems is the reliable preservation and established status of information circulating and processed in information and computing systems and networks.

The advent of global computer networks has made it easy to access information, both for individual users and for large organizations. But the ease and high speed of access to data using computer networks such as *the Internet* have also made the following threats to data security significant in the absence of protection measures:

- unauthorized access to information;
- unauthorized modification of information
- unauthorized access to networks and services;
- Other network attacks, such as replay of previously intercepted transactions and denial-of-service attacks.

With Processing any significant information using individual computer, and even more so in a network, the question arises about its protection from

unauthorized access and use. The most common method of protection in computer systems – the use of passwords – is more suitable for protecting access to computing resources than for protecting information. This is a kind of screen that separates legitimate users of the system from outsiders, passing through which a qualified user gets access to almost all information.

At present, issues related to the preservation and transfer of confidential information have become extremely important in various fields. The problems that arise in this case are solved by *cryptography* - the science of methods of transforming information in order to protect it from illegal users.

A retrospective look at the history of the development of cryptography, as a specific field of human activity, allows us to distinguish three main periods. The first, the longest, is the era of "manual cryptography". Its beginning is lost in ancient times, and its end falls on the 30s of the twentieth century. Cryptography has gone from a magical art to a quite prosaic applied specialty of officials of diplomatic and military departments.

The second period was the creation and widespread introduction into practice of mechanical, then electromechanical and electronic dialing devices, the organization of entire networks of classified communications. Its beginning can be considered the development by *G. Vernam* in 1917 of a scheme of telegraph cipher machines using *a one-time scale*, Fig. 1.1.

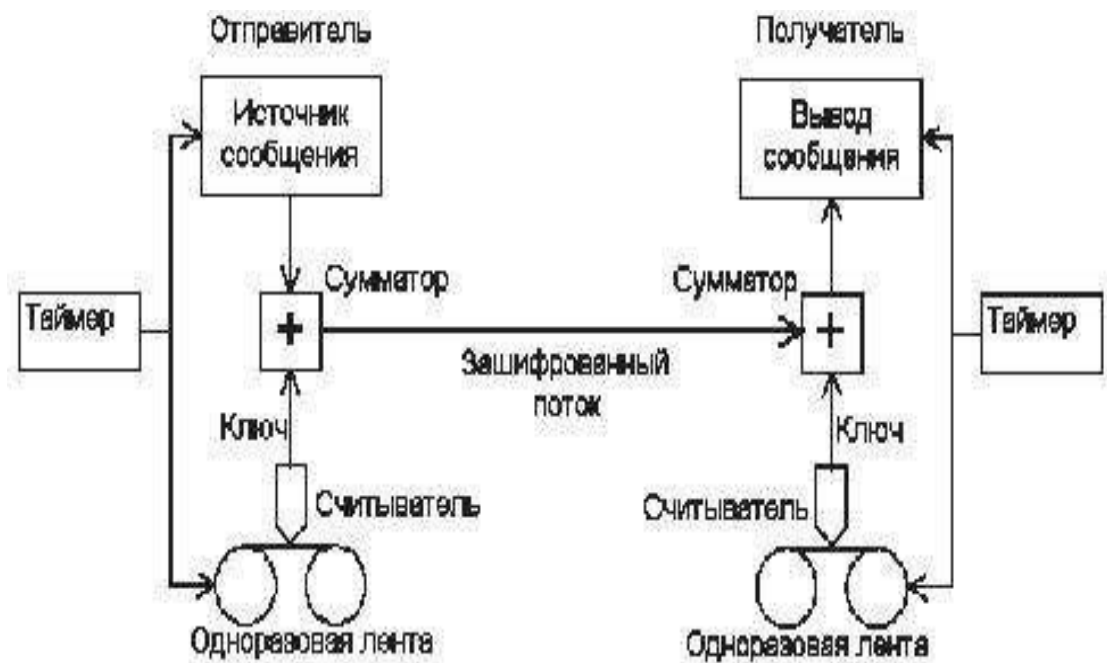


Fig. 1.1. Encryption by the Vernam method

By the mid-1970s, with the development of extensive commercial communication networks, e-mail and global information systems, new problems came to the fore – key supply problems and authentication problems.

In 1976, American scientists *W. Diffie* and *M. Hellman* proposed two new principles for organizing classified communications without pre-providing subscribers with secret encryption keys – the principle of the so-called *public encryption* and the principle of *public key distribution*.

1.2. From the history of cryptography

The concept of "security" covers a wide range of interests, both individuals and entire states. In all historical times

considerable attention was paid to the problem of information security, ensuring the protection of confidential information from

theft, modification, substitution. Cryptography deals with these issues .



Fig. 1.2. John Wallis

Cryptography is a secret writing. The term was introduced by John Wallis (1616-1703), an English mathematician. The need to encrypt and transmit encrypted messages arose a long time ago. Thus, as early as in the V-VI centuries BC, the Greeks used a special ciphering device. According to the description of Plutarch, it consisted of two cylindrical rods of the same length and thickness.

If it was necessary to transmit a message, a long tape of papyrus was wound around the scital, leaving no gap on it. Then, leaving the papyrus on the scital, they wrote everything that was necessary on it, and after writing, they removed the papyrus and sent it to the addressee without a rod. Since the letters were scattered in disorder, only those who had their own scital of the same length and thickness, having wound papyrus on it, could read the message.

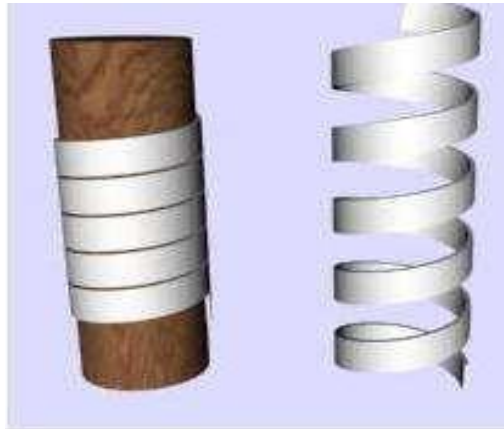


Figure 1.3. Scitala

Polybius square

In Ancient Greece (II century BC) there was a cipher called the "*Polybius square*".

This device was a 5*5 square, the columns and rows of which were numbered with numbers from 1 to 5. One letter was written in each cell (in the Greek version, one cell was empty, and in the Latin version, two letters *I, J* were placed in one cell).

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

Caesar Code

In the first century A.D., during the war with the Gauls, Julius Caesar, corresponding with his friends in Rome, replaced the first letter of the Latin alfavit (*A*) with the fourth (*D*), the second (*B*) with the fifth (*E*), and finally the last letter with the third.

¹ Polybius (200-120 BC), ancient Greek historian

⊕

ABCDEFGHIJKLMNOPQRSTUVWXYZ
 DEFGHIJKLMNOPQRSTUVWXYZABC

YHQL YLGL YLFL
Veni vidi vici - "Пришел, увидел
 победил"
 Ю.Цезарь Донесение Сенату о победе
 над понтийским царем

The Caesar cipher belongs to the so-called class of *monoalphabetic substitutions* and has many modifications.

Cardano lattice

Ciphers belonging to the permutation class are widely known, in particular, the "*cardano lattice*"². This is a rectangular card with holes, most often square, which, when superimposed on a sheet of paper, leaves only some of its parts open. The number of rows and columns on the card is even. The card is made in such a way that when it is turned sequentially, each cell of the sheet lying under it will be occupied. The card is first rotated along the vertical axis of symmetry by 180° and then along the horizontal axis also by 180°. And again repeat the same procedure.

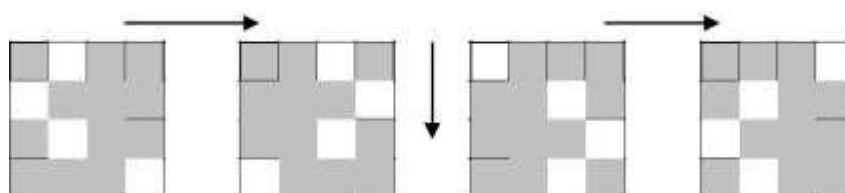


Fig. 1.4. Cardano grate

² Cardano Gerolamo (1501-1576) was an Italian mathematician, philosopher and physician

Alberti Disc

The Italian Alberti (XVI century) was the first to put forward the idea of "double encryption" – the text obtained as a result of the first cipherage was subject to repeated encryption. Alberti's treatise cited his own cipher, which he called "a cipher worthy of kings". He claimed that this cipher was indecipherable. The implementation of the cipher was carried out with the help of a cipher disk, which marked the beginning of a whole series of *multi-alphabetic substitutions*. The encryption process consisted of finding a plaintext letter on the outer disk and replacing it with the corresponding (underneath) letter of the ciphertext. After encrypting several words, the inner disk was moved one step. The key to this cipher was the order of the letters on the inner disk and its initial position relative to the outer disk.

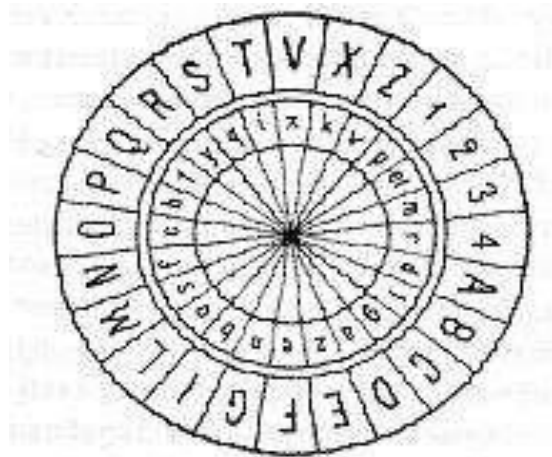


Fig. 1.5. Disk Albert

Vigenère Table

The inconvenience of the monoalphabetical substitution fonts discussed above is obvious, since in the case of using the standard alphabet, the table of frequencies of the letters of the alphabet allows you to determine one or more characters, and this is sometimes enough to break the cipher ("Dancing Men" by Conan Doyle or "The Golden Bug" by Edgar Allan Poe). Therefore, various techniques were used to complicate decryption, for example, the use *of the "Vigenère table"*, which is a square table with the number of rows and columns equal to the number of letters of the alphabet.

In order to obtain a ciphertext, we find the next sign of the logo, starting with the first in the vertical alphabet, and the corresponding sign of the message in the horizontal alphabet. At the intersection of the selected column and row, we find the first letter of the cipher. Obviously, the key to such a cipher is the slogan used.

kinulomoresirocos

e o i k s h a p y y y y s h o v h f s l s s

A

[illegible]

ЧШШЬЫУЯАБВГГЗИЙКЛМНОПРСТУФХЦ
 ШЩЬЕЮЯАБВГГЗИЙКЛМНОПРСТУФХЦЧ
 ЩЬЫААБВГГГЗИЙКЛМНОПРСТУФХЦШ
 ЙЫУЯАБВГЗИЙКЛМНОПРСТУФХЦЩ
 ЫЕЮЯАБВГДЗИЙКЛМНОПРСТУФХЦЩ
 EYUYAABVGDEZHIZIYKLMNOPRSTUFHTSCHSHY
 YUYAABVGDEZHIZIYKLMNOPRSTUFHTSHSH
 YAABVGDEZHIZIYKLMNOPRSTUFHTSHSH

³ Blaise de Vigenère (1523-1596), the French ambassador in Rome, wrote a
 large work on ciphers. Vigenère's square cipher for almost
 400 years not was deciphered, was considered undecipherable
 Code

One-time cipher pad

An example of an unbreakable cipher is the so-called *one-time cipher pad* – a cipher based on the same idea as in the Caesar cipher. Let's call *the extended alphabet* the number of letters of the alphabet and punctuation marks { . , : ; ! ? () – " <space> }, the number of characters of the extended Cyrillic alphabet in this version will be 44. Let us number the symbols of the extended alphabet with numbers from 0 to 43. In this case, any transmitted text can be considered as the sequence { a_n } of the numbers of the set $A = \{0, 1, 2, \dots, 43\}$.

Suppose that we have a random sequence { c_n } of the numbers of set A of the same length as the transmitted text – *the key*.

Adding modulo 44 the number from the transmitted text a_n with the corresponding number from the set of the key c_n :

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

Get a sequence of { b_n } characters of ciphertext. To get the transmitted text, you can use the same key:

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

Two subscribers who are in secret correspondence have two identical notebooks. In each of them, a random sequence of numbers of the set A is printed on several sheets. The sender encrypts his text in the way indicated above, with the help of the first page of the notebook. After encrypting the message, he destroys the used page and sends the text of the message to the second subscriber, the recipient of the ciphertext decrypts it and also destroys it.

used sheet of notebook. It is not difficult to see that a one-time cipher is not disclosure in principle, since a symbol in the text can be replaced by any other symbol and this choice is completely random.

1.3. Encryption Methods

1.3.1. Single-Alphabet Method

This method is perhaps the oldest of all known methods. It is based on a simple method of encryption: the sender and recipient of the encrypted document agree in advance on a certain displacement of letters relative to their usual location in the alphabet. For example, for the Cyrillic alphabet, if the offset is 1, then "A" corresponds to the letter "B", "B" - "V", and so on, and when the alphabet comes to an end, they begin to take letters from the beginning of the list. And it turns out, for example, the following: from the word "ENCODING" we get "LPEYSPGBBOYZH".

A special case of this method is the previously considered Caesar cipher. Obviously, an arbitrary cipher from the class of single-alphabetic methods is not a Caesar cipher (if the cardinality of the text alphabet is n , then the number of Caesar ciphers is n , and the number of all single-alphabet ciphers is $n!$). However, even for such methods, it is easy to propose decryption methods based on the statistical properties of ciphertexts, since open and closed texts have the same static characteristics.

In Laboratory Paper No 1, two variants of the single-alphavit method with a fixed displacement and with an arbitrary (specified) displacement are considered.

1.3.2. Character permutation encryption

The essence of this method is that the characters of the text are rearranged according to certain rules, using only the symbols of the original (unencrypted) text.

Permutations in classical cryptography are usually obtained by writing the source text and reading the ciphertext along different paths of a geometric figure.

The simplest example of permutation is writing the original text on the rows of a matrix and reading it on the columns of this matrix.

The sequence of filling in rows and reading columns can be any and is specified by the key. Thus, for a matrix of 8 x 8 size (block length 64 characters), $1.6 \cdot 10^9$ keys are possible, which makes it possible to decrypt the given text on modern computers by brute force. However, for a matrix of 16 x 16 size (block length 256 characters) there is

$1.4 \cdot 10^{26}$ keys, and it is very difficult to brute force them with the help of modern computing tools.

An example of the permutation method can also be an eight-element table with a set of routes bearing the name of Hamilton routes. The sequence of filling in the table each time corresponds to the numbering of its elements. If the length of the ciphered text is not a multiple of the number of elements, then at the last filling, an arbitrary character is entered into the free elements. Sampling from the table for each fill can be carried out according to its own route, and routes can be used both sequentially and in the order specified by the key.

Permutation methods are characterized by the simplicity of the algorithm, the possibility of software implementation, and a low level of security, since with a long length of the source text, statistical patterns of the key appear in its encrypted version, which allows it to be quickly revealed. Another disadvantage of these methods is easy disclosure, if successful.

For example, if the length of a block in the source text is equal to K characters, then in order to reveal the key, it is enough to pass through the encryption system $K-1$ blocks of the source text, in which all characters, except one, are the same.

1.3.3. Encryption with inverse symbols (in addition to 255)

This method of encryption is a special case of unilateral power 256. The essence of the method is to replace the *ASCII* encoding character with the number i with the symbol with the number $255-i$. The decryption operation is carried out in a similar way.

1.3.4. Multi-Alphabet Encryption Methods

Multi-alphabetic encryption (multi-alphabet replacement) consists in the fact that single-alphabet methods with different keys are used for sequential characters of the ciphered text.

For example, the first character is replaced by the Caesar method with an offset of 14, the second with an offset of 10, and so on until the end of the given key. Then the procedure continues periodically. A more common situation is when not the Caesar cipher is used, but a sequence of arbitrary substitutions corresponding to single-alphabetic methods.

A more illustrative example of such encryption is the gamma method. This method of transformation consists in the fact that the characters of the closed text are sequentially added to the symbols of some special sequence called the gamma. This transformation is sometimes called the overlay of the gamma on the plaintext.

The actual application procedure can be carried out in one of two ways:

1) The characters of the closed text and gamma are replaced by numerical equivalents, and then added modulo K , where K is the number of characters of the alphabet,

$Tsh = (To \oplus Tg) \bmod K$ where Sh – ciphertext,
 To is an open text,
 Tg – scale.

2) Text and gamma symbols are represented in binary codes, and then each pair of binary bits is added modulo 2.

The strength of gamma encryption is determined mainly by the quality of the gamma, which is determined by two characteristics: the length of the period and the randomness of the distribution over the period.

The length of a gamma period is the minimum number of symbols after which the sequence begins to repeat. The randomness of the distribution of symbols over a period means that there are no regularities between the appearance of different symbols within a period.

In Laboratory Paper No 1, three variants of the multi-alpha method are considered: with a fixed key, with a key of a fixed length, and with a key of arbitrary length.

1.3.5. Basic requirements for information encryption methods

1) The complexity and complexity of encryption and decryption procedures should be determined depending on the degree of secrecy of the data being protected.

2) The security of the closure should be such that secrecy is not violated even if the intruder knows the method of closure.

3) The method of closure and the set of service data used (key settings) should not be too complicated. The costs of security transformations should be acceptable for a given level of information preservation.

4) The implementation of the forward and reverse transformation procedures should be formal and as simple as possible.

5) Forward and reverse conversion procedures should not depend on the length of the message.

6) Errors that occur during the conversion process should not propagate through the system and cause loss of information. Due to the occurrence of errors in the transmission of an encrypted message through communication channels, the possibility of reliable decryption of the text at the receiving end should not be excluded.

7) The redundancy of messages introduced by closing should be as low as possible.

8) The volume of the key should not make it difficult to remember and send.

1.3.6. Text histogram

One of the most well-known methods of cryptanalysis is the study of the statistical characteristics of ciphertexts. A graphical representation of the set of frequencies of occurrence of characters in a text is called a histogram of this text.

Suppose that we are dealing with the method of single-alphabetic lettering. Knowing the frequency of occurrence of letters in the alphabet, it is possible to assume which letter has been replaced by a given one. For example, the frequently encountered letter "O" is replaced by the rare letter "И".

For clarity, in Laboratory Paper No 1, double histograms are used, displaying the frequency of occurrence of characters in the source and ciphertext.

Note that the histogram type for the standard allocation depends on the source text type as follows:

The source text contains Cyrillic and Latin characters, then the statistical distribution for Cyrillic and Latin is displayed, if only Cyrillic (Latin), then the statistical distribution for Cyrillic (Latin) is displayed.

Practical lesson No 1

Use of classical crypt algorithms for substitution and permutation to protect textual information

The purpose of the work is to form students' practical skills in researching classical cryptographic algorithms of multi-alphabetic substitution, multi-alphabetic substitution and permutation for the protection of textual information and the formation of histograms that display the frequency of occurrence of characters in the text for the cryptanalysis of classical schiffs.

1. Job description

To perform the work, you need to run the L_LUX.EXE program . A window appears on the display screen with a text editor placed in its center (for displaying encrypted and decrypted texts), in the top bar of the window there is the main menu that allows the user to perform the required action, just below the main menu there is a toolbar (for controlling quick command buttons and other "hot" controls), and at the very bottom of the window, under the text editor, there is a status bar in which a hint is indicated and additional information is displayed. The toolbar keys, for convenience, are equipped with tooltips.

In order to get to the main menu, you need to press the *F10 keyboard*. Moving through the main menu is carried out by moving the cursor keys. To call a menu item, you need to press the "ENTER" key, return to the main menu or exit it altogether - "ESC".

And now let's take a closer look at each of the main menu items.

1.1. Editor

This item of the main menu contains sub-items: create a document, open a file, save a file, exit the program.

In advance, immediately after starting the program, the text editor is not available, as well as almost all the main menu items (except for creating a document, opening a file, exiting the program, program information) and most of the control panel keys (except for creating a document, opening a file and exiting the program).

Create document (Ctrl+N) – this sub-item makes the text editor available for work (the user gets the right to create his own text file, which can later be used when working with the program), it is also possible to use all the inaccessible items of the main menu and the keys of the control panel.

Open file (Ctrl+L) – when this option is selected, a dialog box will appear, allowing you to select the file to be uploaded. The contents of the file will be displayed in the text editor window.

Similar to the "Create document" item, a text editor with the displayed text becomes available for work, and it is also possible to use all previously inaccessible items of the main menu and control panel keys.

Save file (Ctrl+S) - when you select this option, a dialog box appears that allows you to save the contents of the text editor to disk.

Exit the program (Ctrl+X) - when this option is selected, a dialog box appears that allows you to exit the program.

1.2. Histogram

Displays two histograms showing the frequency of characters in the text.

***Attention!** It makes no sense to call the histogram before encrypting and decrypting, since the texts for which the histogram will be viewed have not yet been formed.*

You can view the following combinations of histograms:

- histograms of the original and encrypted file,
- histograms of the encrypted and decrypted file,
- histograms of the standard distribution and encrypted text hundred,
- Histograms standard Distributions and deciphered text.

Histograms use the left and right mouse keys to zoom.

For example, after encrypting a large amount of text, the user wants to see histograms of the original and encrypted files. Since the text sizes are quite large, two histograms with a large number of columns in each (a column corresponds to one character of the text) will be displayed, but it will be difficult to say which of these columns corresponds to a particular text character, and what is the frequency of occurrence of this character. Therefore, the user has the opportunity to zoom in on any of the two histograms in order to more accurately determine the required value of the frequency of occurrence of a particular character. To do this, you need to hover the mouse pointer over the left border of the part of the histogram that you want to zoom in, then press the left mouse button, and without releasing it, stretch the rectangle so that its lower right corner coincides with the right border of the enlarged section of the histogram. After that, you should release the left mouse button and an enlarged image of the desired area will appear on the screen.

Moreover, by clicking and not releasing the right mouse button, you can move the histogram in any direction in order to study the entire resulting distribution on an enlarged scale.

In order to return from the enlarged scale to the original view, you need to hover the mouse pointer over the histogram, then press the left mouse button, and, without releasing it, stretch a small rectangle from bottom to top, after which you should release the left mouse button and the original image of the histogram will appear on the screen.

1.3. Encryption

Encrypting a text file is done by one of the seven methods discussed in the lab.

1. Single-alphabet method (fixed offset).
2. Single-alphabet method with a specified offset (from 2 to 20).
3. Transposition of symbols.
4. In addition, up to 255 (inverse method).
5. Multi-alphabet method (with a fixed key).
6. Multi-alphabetic method with a fixed-length key.
7. Multi-alphabetic method with a key of arbitrary length.

The encryption method is selected both with the mouse and with the cursor movement keyboards and the "ENTER" key.

Then a window appears, in which, depending on the encryption method, you need to specify certain parameters, and either confirm the encoding process or refuse it. After that, the ciphertext will be displayed in the editor window.

1.4. Decryption

Similar to the previous step, the decryption method is selected (must match the method used to encrypt the file).

A window appears again, in which, depending on the decryption method, you need to specify certain parameters, and either confirm the decryption process or refuse it.

After that, the deciphered text will be displayed in the editor window. If the transcription is correct, the resulting text matches the original text.

1.5. More information

The program provides the ability to view additional information ('Help Ctrl+F9'), reference information about the encryption methods used ('About Ctrl+F10 methods'), information about the program ('About Ctrl+F11').

2. Emerging competencies

This laboratory Work sent by on formation of the following competencies of a bachelor's graduate:

OPK-1: the ability to install software and hardware for information and automated systems;

PC-5: the ability to interface hardware and software as part of information and automated systems.

3. Theoretical part

To perform the laboratory work, you need to start the L_LUX program. A window will appear on the display screen with a text editor placed in its center. In the top bar of the window there is the main menu, which allows the user to perform the required action. Just below the main menu there is a toolbar, and at the very bottom of the window, below the text editor, there is a status bar in which a hint is indicated and additional information is displayed. Keyboard

Vichy toolbars are equipped with tooltips for convenience.

In order to get to the main menu, you need to press the F10 keyboard. Moving through the main menu is carried out with the mouse.

An example of working with the program.

Let's consider single-alphabet encryption with a fixed key. Load the source text (any) from 10 to

15 sentences. Save it for further work with this file. Then call the ENCRYPTION menu item, select the single-alphabet method (with a fixed offset). In the window that appears, click the ENCRYPT key. After the encryption is completed, you can view the ciphertext in the editor.

Go to the HISTOGRAM menu item. Select a histogram type that displays the source and ciphertext histograms. Analyze the histograms. They should look more or less the same.

To find out the cipher key (the offset of the second alphabet relative to the first), it is necessary to find the characters that have the same frequency of occurrence in the histograms. For example, the most frequent character in the first histogram should turn into the most frequent character in the second histogram during encryption. Thus, having found the two most common characters in both histograms, you can calculate the offset using the standard ASCII code table. Knowing the offset and the symbol encoding table, the text can be easily deciphered. By calling the DECRYPTION menu, you can perform the same actions in automatic mode.

Note. When encrypting and decrypting from the encoding table, symbols with codes 176-223 and 240-255 are not used, that is, during manual decryption, these symbols should be omitted and it should be assumed that, for example, the symbol I has the code 159, and 223, similar to P not 175, but 239.

Sometimes the histograms below the columns showing the frequency of the characters encountered do not show the symbols themselves, but their tabular codes in square brackets.

4. Equipment and materials

Laboratory hardware includes computer classrooms with network equipment installed, as well as

- Windows operating system;
- the program L_LUX in the C++ environment.

5. Safety Instructions

Safety requirements before starting work:

- DO NOT: change clothes, use fire, smoke, eat in the laboratory;
- make sure that electrical outlets and connectors are intact. It is forbidden to work with a faulty power tool, break it and repair it yourself;
- in the laboratory, you must wear a change of shoes;
- turn on the computer only after obtaining permission for the work performed and permission from the teacher or laboratory assistant.

Safety requirements during work:

- when performing laboratory work, students are required to use only computer equipment, peripheral equipment, connecting cables, measuring equipment and diskettes directly related to this laboratory lesson;
- Connection and disconnection of the components of the computing complex shall be carried out only when the voltage is completely removed from all components of the computing complex.

Safety requirements at the end of work:

- at the command of the teacher, students are obliged to turn off the computer and the components of the computer complex;
- put the workplace in order.

6. Quests

Task No1. For the single-alphabetic method with a fixed offset, determine the offset set in the program. To do this, you should:

- view your text file pre-created using the editor;
- Encrypt this file.
- View the encrypted file in the editor.
- view histograms of the source and ciphertext;
- describe histograms (similarities, differences) and determine the offset with which the encryption was performed;
- Decrypt the ciphertext:
 - Using a histogram, then check the correctness of the decryption in the editor;
 - Manually use histograms to describe and explain the decryption process.

For each encryption method, the report describes the sequence of the actions performed, the names of all files used, the histograms obtained, the found offset, and the decryption process.

The teacher is provided with a report on the work done and all used and created files.

Task No2. For a single-alphabetic method with a given offset (Caesar cipher), you should:

- Perform arbitrary offset encryption on your source text.
- view and describe histograms of the original and encrypted

texts, determine the offset for several characters;

- decrypt the text using the program;
- decrypt the text encrypted with the Caesar cipher using the program using the method of brute force displacement; indicate in the report with what displacement the file was encrypted.

Task No3. For the method of transposing symbols, decrypt the encrypted file. To do this, you need to determine the law of transposition of plaintext characters. Create a small file of a few characters with the text you know, encrypt it. Look at the histograms, describe them, and answer whether it is possible to extract information useful for decryption from them. Use the editor to compare your source text and the ciphertext and determine the law of character permutation.

Decrypt the file:

- manually and explain your actions;
- with the help of the program.

Task No4. For inverse coding (up to 255 additions)

- Encrypt your arbitrary file.
- look at the histograms of the source and ciphertext, describe the histograms and determine the offset for several characters;
- Decrypt the ciphertext, check the decryption in the editor.

Task No5. For multi-alphabetic encryption with a fixed key, determine how many single-alphabetic methods are used in the program and with what offset. To do this, you need to create your own file consisting of a string of identical characters, encrypt it and determine the encryption method and the set of offsets from the histogram.

Task No6. For multi-alphabetic encryption with a fixed-length key:

- Perform encryption and determine from the histogram what offset each character receives for a file consisting of a string of the same

symbols;

- Encrypt and decrypt a freetext file.
- view and describe the histograms of the source and ciphertext; answer

what information can be obtained and the histogram.

Task No7. For multi-alphabetic encryption with an arbitrary password, the task is completely similar to task 6.

7. Contents of the report

The report must contain:

1. the name of the work;
2. the purpose of the work;
3. working diagrams;
4. basic ratios and calculation formulas;
5. tables;
6. Conclusions on the work.

8. Control questions

Give answers to control questions in the report in accordance with the number of the option specified by the teacher (Table 1.1).

Table 1.1 – Control questions

Variant Number	Control questions
1, 5, 7, 3, 9, 18, 28	What cryptographic protection methods do you know? files?
2, 4, 6, 8, 20, 22, 24, 26, 30	What are the advantages and disadvantages of single-alphabetic me- Todov?
11, 13, 15, 10, 17, 19, 27	If it is necessary to encrypt a text containing an important information about which of the methods you have considered choose? Justify your choice
12, 14, 16, 21, 23, 25, 29	Is it advisable to reuse for already stiffened? text:

	(a) Multi-alphabetic encryption; b) Caesar's method?
--	---

9. List of references recommended for use on this topic

1. Zhuk, A.P., Zhuk, E.P., Lepeshkin, O.M., Timoshkin, A.I. Information Protection: Textbook for Bachelors [Text] / A.P. Zhuk et al. – M.: INFRA-M, 2015. – 392 p.

2. Shangin, V.F. Zashchita komp'yuternoy informatsii [Protection of computer information]. Effective methods and means [Text] / V.F. Shangin – Moscow: DMK Press, 2012. – 544 p.

Practical lesson No 2

Study of Various Methods of Text Information Protection and Their Durability Based on Key Guessing

The aim of the work is to study the methods of encrypting/decrypting by transmutation of symbols, substitution, gamma, the use of the Vigenère table, to compare the strength of various methods based on attacks by brute force of all possible keys.

In the laboratory work, a method of cracking a cipher based on the search for all variants of the key is considered. The criterion for the correctness of the variant is the presence of a "probable word" in the text.

A set of all possible keys is searched, the ciphertext is deciphered on each key. In the resulting "pseudo-clear" text

a probable word is searched. If there is no such word, the current text is rejected and the next key is moved. If such a word is found, a variant of the key is displayed on the screen. Then the search for keys continues until all the many variants are exhausted. It is possible to find several keys in which there is a probable word in the "pseudo-plaintext".

After completing the search, it is necessary to decipher the text on the found keys.

"Pseudo-open text" is displayed on the screen for visual control.

If the operator recognizes the text as open, then the work on opening is completed. Otherwise, this version of the key is rejected and the transition to the next key is carried out.

1. Emerging competencies

This laboratory Work sent by on formation of

the following competencies of a bachelor's graduate:

OPK-1: the ability to install software and hardware for information and automated systems;

PC-5: the ability to interface hardware and software as part of information and automated systems.

2. Theoretical part

To perform laboratory work, you need to run the *LAB_RAB.exe* program used to encrypt/decrypt and decrypt (by probable word pulling) files. The system implements the following functions:

- entering, deleting and selecting user keys;
- key list support;
- encryption and decryption of text;
- decryption of the text by selecting keys, by pulling a probable word.

The system supports the following methods of cryptographic transformation of information:

- replacement;
- permutation;
- gamma;
- Vigenère's table.

When the encryption and decryption utilities are launched, the user is asked to confirm the correctness of the chosen method for work and the compliance of the specified key with the user's goals (also, when a file is changed in a text editor, a request is always given to save the changes at each step, the further development of which will lead to irreversible changes in the file and the loss of the original information).

Interface description.

- A text editor window with a wide range of additional functions.
- A table of all keys entered into the system with quick access to enter, delete, or select the current key.
- A list of all encryption methods for quick and easy switching between them.

- Main menu (at the top of the screen).
- An additional menu (accessed by right-clicking the shi).
- A set of auxiliary buttons for quick and easy interaction face.
- Fields for displaying the current state of the system:
- Current key
- probable word;
- The power of the wrench for broaching.

Example of working with the program

***Attention!** Be careful when setting the operating parameters, since in the process of calculation in the course of work, these parameters can no longer be changed. After starting the program, absolutely all working fields are empty and it is necessary to make initial settings for the system to work.*

1. Enter a list of keys.
2. Enter a probable word (not necessarily at the beginning, before entering it, all the broach start menus are still unavailable).
3. The required encryption method is selected.
4. The original or encrypted file is downloaded (the corresponding menus for encryption and decryption open).
5. The required process is started:
 - encryption;
 - decryption;
 - stretching a probable word;
 - text conversion.
6. Continue in any order as described.

7. When you finish your work, remember to save the necessary results (when closing and loading new files, the system automatically asks for confirmation of the record).

Encryption

1. Open the file.
2. Make the necessary changes.
3. Configure the appropriate parameters:
 - type of encryption;
 - key;
 - others.
4. Start the encryption process via the menu item
UTILITIES \ ENCRYPT F5.

***Attention!** When encrypting a file, all changes made by the user up to the current point in time will be saved to the hard drive.*

Decryption

1. Open the file.
2. Make the necessary changes.
3. Configure the appropriate parameters:
 - type of encryption;
 - others.
4. Start the decryption process via the menu item
UTILITIES \ DECRYPT F6.

***Attention!** When decrypting a file, all changes made by the user up to this point in time will be saved to the hard drive.*

Probable word broaching (decryption)

***Attention!** The power of the key will be set in advance in the "key strength" option. The length of the key significantly affects the time it takes to draw a probable word (in the worst case, we are dealing with a logarithmic algorithm).*

1. A probable word (1(3) to 9) is entered
2. To separate the newly found keys from the previous ones, the inscription "selection" is added between them.

3. Enumeration of keys.
4. The first line of text is decrypted by the current key.
5. In portions equal to the length of the probable word, the content of this line is compared with the meaning of the probable word.
6. If at least one match is found, remember the key.
7. Let's move on to the new key.
9. The results must be contained in the key list. If no matches are found, nothing is added to the key list.

Key operations

The following actions can be performed with the key database:

- Add a new key.
- Delete a single record.
- Change the active record.
- clear the entire table of entered keys.

<i>Note. The words "working with the key table" refer to the keys introduced for use in two methods (gamma and Vigenère table).</i>
--

Keys for rearrangement

There can be only one current permutation key in the system at a time.

Rules for entering the key for permutation:

1. When switching to the "Permutation" item in the list of encryption methods supported by the system, the permutation key entry window is called. The window consists of two buttons (Cancel and Exit unchanged and the Enter button – confirmation of the set key length) and a window for setting the key length for permutation.

2. In the key length window, select the required length (the parameters are replaced within 1 ... 9), and confirm your desire to use the key of this length.

3. After confirmation, the window will display buttons with numbers on the front side (in an amount equal to the length of the key), when the button is pressed, the button is fixed (discolored) for

the impossibility of its further use (since all the digits in the permutation key must be non-repeating).

4. After enumerating all the buttons, the system remembers the entered key, displays it in the key input field and exits the switch key entry window to the main program window.

4. Equipment and materials

Laboratory hardware includes computer classrooms with network equipment installed, as well as

- Windows operating system;
- The program *LAB_RAB* in the C++ environment.

5. Safety Instructions

Safety requirements before starting work:

- DO NOT: change clothes, use fire, smoke, eat in the laboratory;
- make sure that electrical outlets and connectors are intact. It is forbidden to work with a faulty power tool, break it and repair it yourself;
- in the laboratory, you must wear a change of shoes;
- turn on the computer only after obtaining permission for the work performed and permission from the teacher or laboratory assistant.

Safety requirements during work:

- when performing laboratory work, students are required to use only computer equipment, peripheral equipment, connecting cables, measuring equipment and diskettes directly related to this laboratory lesson;
- Connecting and Disconnecting Computing Components of the complex should be carried out only when the voltage is completely removed from all components of the computer complex.

Safety requirements at the end of work:

- by team teacher Students Obligated turn off the computer and components of the computer complex;
- put the workplace in order.

6. Quests

To perform the lab work, you need to run the *LAB_RAB.exe* program used to encrypt/decrypt and decrypt (probable word pulling) the files.

The system implements the following functions:

- entering, deleting and selecting user keys;
- key list support;
- encryption and decryption of text;
- decryption of the text by selecting keys, by pulling a probable word.

The system supports the following methods of cryptographic transformation of information:

- replacement;
- permutation;
- gamma;
- Vigenère's table.

When the encryption and decryption utilities are launched, the user is asked to confirm the correctness of the chosen method for work and the compliance of the specified key with the user's goals (also, when a file is changed in a text editor, a request is always given to save the changes at each step, the further development of which will lead to irreversible changes in the file and the loss of the original information).

Interface description:

1. A text editor window with a wide range of additional functions.

2. A table of all keys entered into the system with quick access to enter, delete, or select the current key.

3. A list of all encryption methods for quick and easy switching between them.

4. Main menu (at the top of the screen).

5. An additional menu (accessed by right-clicking the shi).

6. A set of auxiliary buttons for quick and easy interaction face.

7. Fields for displaying the current state of the system:

- Current key
- probable word;
- The power of the wrench for broaching. Example of working with the program

Attention! Be careful when setting the operating parameters, since these parameters cannot be changed during the calculation process in the course of work.

After starting the program, absolutely all working fields are empty and it is necessary to make initial settings for the system to work.

1. Enter a list of keys.

2. Enter a probable word (not necessarily at the beginning, before entering it, all the broach start menus are still unavailable).

3. The required encryption method is selected.

4. The original or encrypted file is downloaded (the corresponding menus for encryption and decryption open).

5. The required process is started:

- encryption;
- decryption;
- stretching a probable word;

– text conversion.

6. Continue in any order as described.

7. When you finish your work, remember to save the necessary results (when closing and loading new files, the system automatically asks for confirmation of the record).

Encryption

1. Open the file.

2. Make the necessary changes.

3. Configure the appropriate parameters:

– type of encryption;

– key;

– others.

4. Start the encryption process via the menu item

UTILITIES \ ENCRYPT F5.

Attention! When encrypting a file, all changes made by the user up to this point in time will be saved to the hard drive.

Decryption

1. Open the file.

2. Make the necessary changes.

3. Configure the appropriate parameters:

– type of encryption;

– others.

4. Start the decryption process via the menu item

UTILITIES \ DECRYPT F6.

Attention! When decrypting a file, all changes made by the user up to this point in time will be saved to the hard drive.

Probable word broaching (decryption)

Attention! The power of the key is set in advance in the "key strength" option. The length of the key significantly affects the time of pulling the probable word

(in the worst case, we are dealing with a logarithmic algorithm).

1. A probable word (1(3) to 9) is entered
2. To separate the newly found keys from the previous ones, the inscription "selection" is added between them.
3. Enumeration of keys.
4. The first line of text is decrypted by the current key.
5. In portions equal to the length of the probable word, the content of this line is compared with the meaning of the probable word.
6. If at least one match is found, remember the key.
7. Let's move on to the new key.
8. Let's move on to the next line.
9. The results must be contained in the key list. If no matches are found, nothing is added to the key list.

Key operations

The following actions can be performed with the key database:

- Add a new key.
- Delete a single record.
- Change the active record.
- clear the entire table of entered keys.

Note. Under the words "working with the key table" there are the keys that have been introduced for use in two methods (gamma and Vigenère table).

Keys for rearrangement

There can be only one current permutation key in the system at a time.

Rules for entering the key for permutation:

1. When you switch to the "Permutation" item in the list of encryption methods supported by the system, the window for entering the key is called

setting. The window consists of two buttons (Undo and Exit unchanged and the Enter button – confirmation of the set key length) and a window for specifying the key length for permutation.

2. In the key length window, select the required length (the parameters are replaced within 1 ... 9), and confirm your desire to use the key of this length.

3. After confirmation, the window will display buttons with numbers on the front side (in an amount equal to the length of the key), when the button is pressed, the button is fixed (discolored) to prevent its further use (since all digits in the permutation key must be non-repeating).

4. After enumerating all the buttons, the system remembers the entered key, displays it in the key input field and exits the switch key entry window to the main program window.

Task No1.

1. Familiarize yourself with the description of the laboratory work and the task.

2. Configure the program: select an encryption method; enter keys for all methods; enter a probable word; perform all other system settings.

Task No2. For the replacement method (single-alphabet method):

- select this algorithm in the list of available encryption methods.

;

- set the required offset;
- open an arbitrary file;
- View the contents of the original file.
- Encrypt this file (if necessary

you can specify the name of the encrypted file);

- View the encrypted file in the editor.
- enter a probable word;
- enter the probable key length (except for the replacement method);

- pick up the key;
- decrypt with all found keys;
- Find a correctly decrypted keyword in any of the decrypted files.
- decrypt the file with the original key;
- check the result.

Task No3. For the permutation method:

- choose the method of permutation;
- in the window that opens, first specify the length of this key, and then from the buttons that appear, make the necessary combination for the key by clicking on the buttons in the specified order; in this case, the already used buttons become unavailable to prevent their re-entry;

▪ Further, the actions fully correspond to those described in the previous paragraph of the task.

Task No4. For the gamma method:

- choose the gamma method;
- enter the gamma key;
- Repeat the second point from task 3 in full.

Task No5. For the Vigenère table, all actions are repeated from the rear. 4 (gamma method).

For each encryption method, the report describes the sequence of the actions performed, the names of all files used, the source and found keys, and the deshuffling process is described.

The teacher is provided with a report on the work done and all the files used.

7. Contents of the report

The report for each encryption method describes the sequence of actions performed, indicating the names of all the

files, source and found keys, the deshuffling process is described.

The teacher is provided with a report on the work done and all the files used.

8. Control questions

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. List of references recommended for use on this topic

1. Zhuk, A.P., Zhuk, E.P., Lepeshkin, O.M., Timoshkin, A.I. Information Protection: Textbook for Bachelors [Text] / A.P. Zhuk et al. – M.: INFRA-M, 2015. – 392 p.

2. Shangin, V.F. Zashchita komp'yuternoy informatsii [Protection of computer information]. Effective methods and means [Text] / V.F. Shangin – Moscow: DMK Press, 2012. – 544 p.

Practical lesson No 3

Aes Rijndael *symmetric encryption standard*

The aim of the research is to study the principles of encryption used in the *AES RIJNDAEL symmetric encryption algorithm* and the ability to draw independent conclusions based on the results of the research.

2. Emerging competencies

This Work sent by on formation of the following competencies of a bachelor's graduate:

OPK-3: the ability to install software and hardware for information and automated systems;

PC-5: the ability to interface hardware and software as part of information and automated systems.

3. Theoretical part

State, encryption key, and number of cycles. Rijndael is an iterative block cipher with variable block lengths and different key lengths. The key length and block length can be 128, 192, or 256 bits independently.

A variety of transformations work with an intermediate result called State.

The state can be represented as a rectangular array of bytes. This array has 4 rows, and the number of columns is denoted as N_b and is equal to the length of the block divided by 32.

The encryption key is also represented as a rectangular mass with four rows. The number of columns is denoted as N_k and is equal to the length of the key divided by 32. This is shown in Fig. 1.

In some cases, the encryption key is shown as a linear mass of four-byte words. Words consist of 4 bytes that are in a single column (when presented as a rectangular array).

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

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

Figure 1. Example representation of State (Nb=6) and Key Siff- (Nk=4)

The inputs to the cipher are denoted as state bytes in the order $a_{0,0}$, $a_{1,0}$, $a_{2,0}$, $a_{3,0}$, $a_{0,1}$, $a_{1,1}$, $a_{2,1}$, $a_{3,1}$... After the cipher expires, the output is obtained from the state bytes in the same order. The number of cycles is denoted as Nr and depends on the values of Nb and Nk.

Table 4.1. Number of cycles (Nr) as a function of key length and block length

Nr	Nb = 4	Nb = 6	Nb = 8
Nk = 4	10	12	14
Nk = 6	12	12	14
Nk = 8	14	14	14

Cyclic transformation. Cyclic transformation consists of four different transformations: byte replacement, row shifting, column shuffling, and adding a loop key. The latter cycle differs from a simple loop only in the absence of column mixing.

```

Round (State, RoundKey)
{
ByteSub(State); ShiftRow(State) byte
replacement; MixColumn(State) row shifting;
Column shuffling
AddRoundKey(State, RoundKey); Adding a Loop Key
}

```

The last cipher cycle is slightly different:

```
FinalRound(State, RoundKey)
{
  ByteSub(State); byte replacement
  ShiftRow(State); line shift
  AddRoundKey(State, RoundKey); Adding a Loop Key
}
```

As you can see, the latter cycle differs from a simple cycle only in the absence of column mixing. Each of the above transformations is discussed in detail below.

Column Kneading Conversion (MixColumn)

The transformation is a multiplication of the state by the ME matrix for encryption or the MD matrix for decryption:

ME $\begin{bmatrix} & \\ & \end{bmatrix}$ State

MD $\begin{bmatrix} & \end{bmatrix}$ State

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

$$b1 = (b1 * 2) \text{ XOR } (b2 * 3) \text{ XOR } (b3 * 1) \text{ XOR } (b4 * 1)$$

Two bytes are multiplied according to the following algorithm:

- If one of the bytes is 0, the result is 0;
- if one of the bytes is 1, the result will be another byte;
- in other cases, each byte is replaced according to table L, the replaced bytes are added, if necessary, 255 is subtracted to fall into the interval [0, 255] and replaced according to table E, which gives the result.

```

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
};

```

Adding a Cycle Key

In this operation, the cycle key is added to the state using simple EXOR. The cycle key is generated from the encryption key using a key schedule algorithm.

The length of the cycle key is equal to the length of the block Nb.

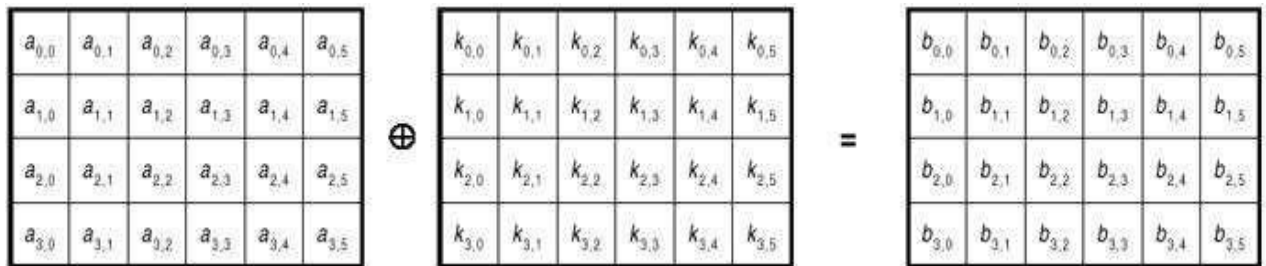


Fig. 4. 5. When you add a key, the cycle key is folded by EXOR with the state

In encryption, parts of the extended key are selected from beginning to end, in decryption – from end to beginning.

Key Expansion

The extended key is a linear array of 4-byte words and is denoted as $W[Nb*(Nr+1)]$. The first Nk words contain the encryption key. All other words are determined recursively from words with lower indices. The algorithm for generating keys depends on the value of Nk : below is a version for Nk equal to or less than 6 and a version for Nk greater than 6.

For $Nk < 6$ or $Nk = 6$ we have:

```
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];
    }
}
```

As you can see, the first Nk words are filled in with the cipher key. Each subsequent word $W[i]$ is obtained by EXOR of the previous word $W[i-1]$ and the word on Nk positions before $W[i-Nk]$. For words whose position is a multiple of Nk , the conversion to $W[i-1]$ is applied before EXOR, and then a cyclic constant is added. The transformation contains a cyclic byte shift in the word, denoted as $Rotl$, followed by $SubByte$ - the application of byte replacement.

For $Nk > 6$ we have:

```
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];
}
}

```

The difference for the scheme at $N_k > 6$ is the use of SubByte for every 4th byte of N_k .

The cyclic constant is independent of N_k and is determined by the following order:

```

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

```

Encryption

The Rijndael cipher includes the following conversions:

- Initial addition of a loop key
- $N_r - 1$ cycles;
- final cycle.

On pseudo-C, it looks like this:

```

Rijndael (State, CipherKey)
{
KeyExpansion(CipherKey, ExpandedKey); AddRoundKey(State,
ExpandedKey) key extension; For (i=1 ; i<Nr ; i++)
Round(State,ExpandedKey+Nb*i) FinalRound(State,
ExpandedKey+Nb*Nr) loops; final loop
}

```

If the key extension procedure has been performed beforehand, the procedure will look like this:

```

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

```

Description of the demo program The program is made in C# and consists of two elements - a Rijndael.dll file containing the implementation of the encryption algorithm, and a demo application RijndaelDemo.exe. For the application to work, you need Windows OS with .NET Framework v1.1 installed. In the main window of the demo program, you can specify the length of the key, the length of the block, you can see the extended encryption key, calculated in accordance with the specified encryption key.

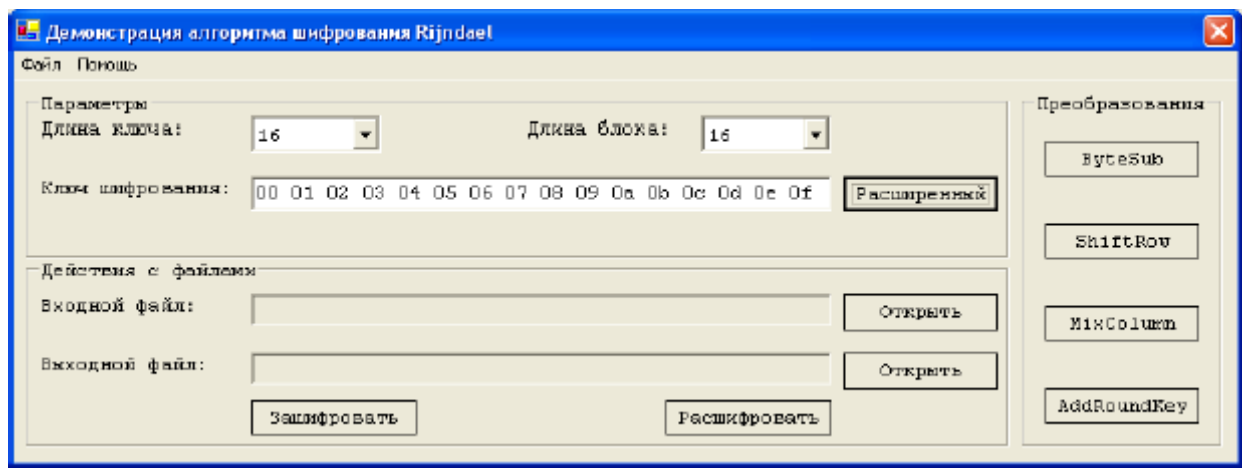


Figure 2. Main window of the demo program

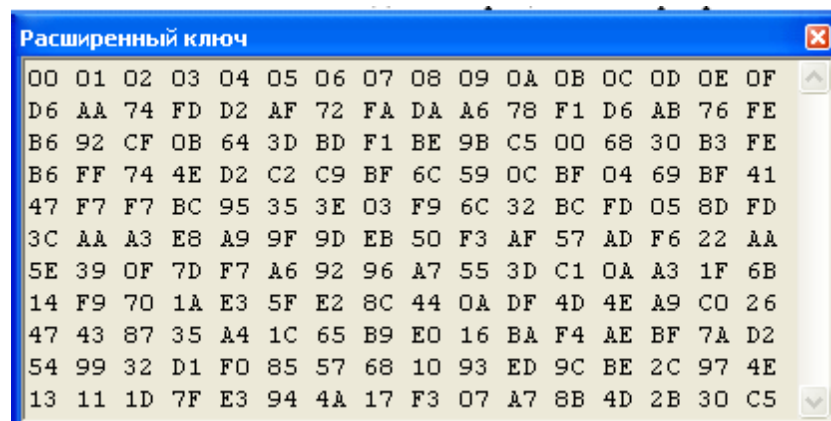


Figure 3. Extended key window

It is also possible to take a closer look at the effect of all cyclic transformations (ByteSub, ShiftRow, MixColumn, AddRoundKey) both in encryption and decryption, Fig. 4, 5.

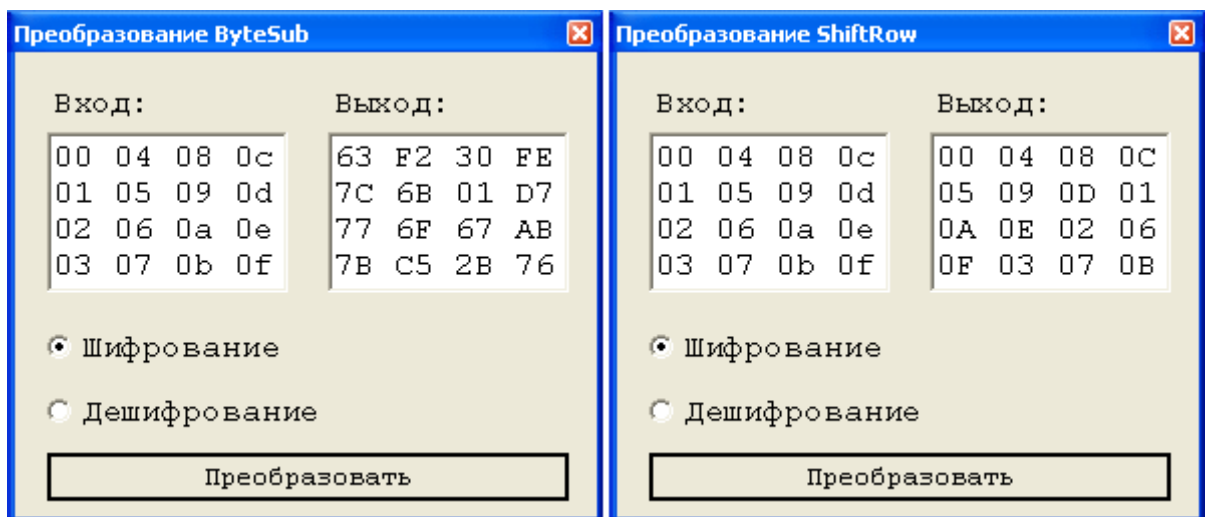


Figure 4. ByteSub and ShiftRow conversion windows

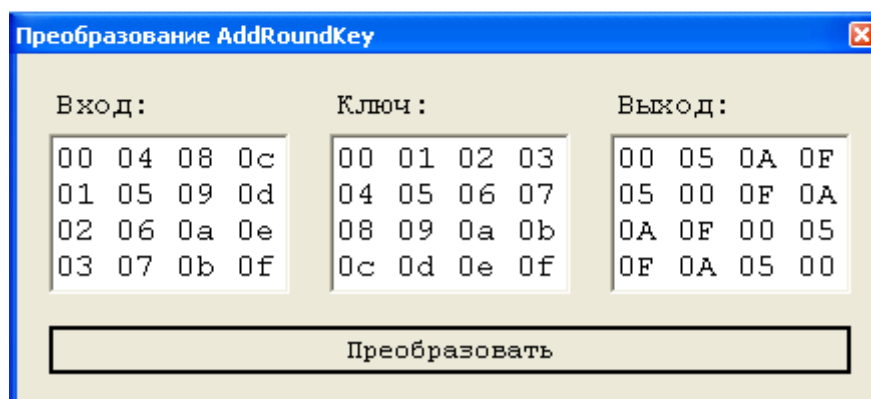
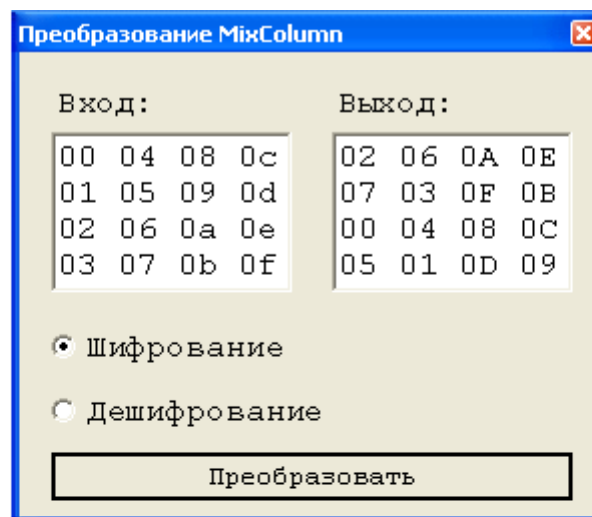


Figure 5. MixColumn and AddRoundKey transformation windows

When encrypting, it is proposed to select the source file and the file where the encryption result will be placed, when decrypting, respectively, the encrypted file and the file intended for placing the result of the encryption

tata decryption. In the process, the encryption key and the length of the key and block specified in the main window of the program are used.

4. Equipment and materials

To perform the work, you will need:

- class of personal computers not lower than Pentium-4, blackboard, chalk;
- Microsoft Windows operating system, software

Remedy

RijndaelDemo ,

5. Safety Instructions

Safety requirements before starting work:

- DO NOT: change clothes, use fire, smoke, eat in the laboratory;
- make sure that electrical outlets and connectors are intact. It is forbidden to work with a faulty power tool, break it and repair it yourself;
- in the laboratory, you must wear a change of shoes;
- turn on the computer only after obtaining permission for the work performed and permission from the teacher or laboratory assistant.

Safety requirements during work:

- when performing laboratory work, students are required to use only computer equipment, peripheral equipment, connecting cables, measuring equipment and diskettes directly related to this laboratory lesson;
- Connection and disconnection of the components of the computing complex shall be carried out only when the voltage is completely removed from all components of the computing complex.

Safety requirements at the end of work:

- at the command of the teacher, students are obliged to turn off the computer and the components of the computer complex;

- put the workplace in order.

6. Quests

Task No1. Read the information about *the RijndaelDemo* program in section 3. Start the RijndaelDemo.exe module.

Task No2. To study the methods of encryption and decryption using the *Rijndael* algorithm using the example of ordinary text files. To consider in detail the effect of all cyclic transformations (ByteSub, ShiftRow, MixColumn, AddRoundKey), both in encryption and decryption. The source text for encryption can be prepared earlier and saved in the *.txt file.

Task No3. Save in the report screen forms demonstrating the process of encrypting and decrypting information, analyze the results obtained.

Task No4. Include in the laboratory report the answers to the control questions selected in accordance with the option number specified by the teacher.

7. Contents of the report

The report must contain:

1. Name of the work, purpose.
2. The sequence of actions in the performance of each task.
3. Conclusions on the implementation of laboratory work.

8. Control questions

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

9. List of references recommended for use on this topic

1. Zhuk, A.P., Zhuk, E.P., Lepeshkin, O.M., Timoshkin, A.I. Information Protection: Textbook for Bachelors [Text] / A.P. Zhuk et al. – M.: INFRA-M, 2015. – 392 p.
2. Shangin, V.F. Zashchita komp'yuternoy informatsii [Protection of computer information]. Effective methods and means [Text] / V.F. Shangin – Moscow: DMK Press, 2012. – 544 p.

LABORATORY WORK №4

STUDY OF THE STRUCTURE AND PRINCIPLE OF OPERATION OF THE ENIGMA CIPHER MACHINE (*Enigma*)

Objective: To study the principles of information encryption/decryption used in the Enigma cipher machine. To get acquainted with the general principles of the Enigma cipher machine on the example of an emulator.

1. Description of the Enigma

laboratory work, device and principle of

opera



Fig. 3.1. Appearance of the Enigma cipher machine

Enigma4 was a portable cipher machine used to encrypt and decrypt secret messages. More precisely, Enigma is a whole family of electromechanical rotary machines, used since the 20s of the XX century.

⁴ Enigma Emulators <http://www.attlabs.att.co.uk/andyc/enigma>

Enigma was used commercially, as well as in military and government services in many countries around the world, but was most widely used in Germany during the Second World War. It is the *Wehrmacht Enigma*, the German military model, that is most often the subject of study.

Although the Enigma cipher, from the point of view of cryptography, was quite weak, in practice only a combination of this factor with others, such as errors in the op

erators, procedural flaws, and enigma instance captures

and cipher books, which allowed English cryptanalysts to open messages encrypted with the Enigma cipher.

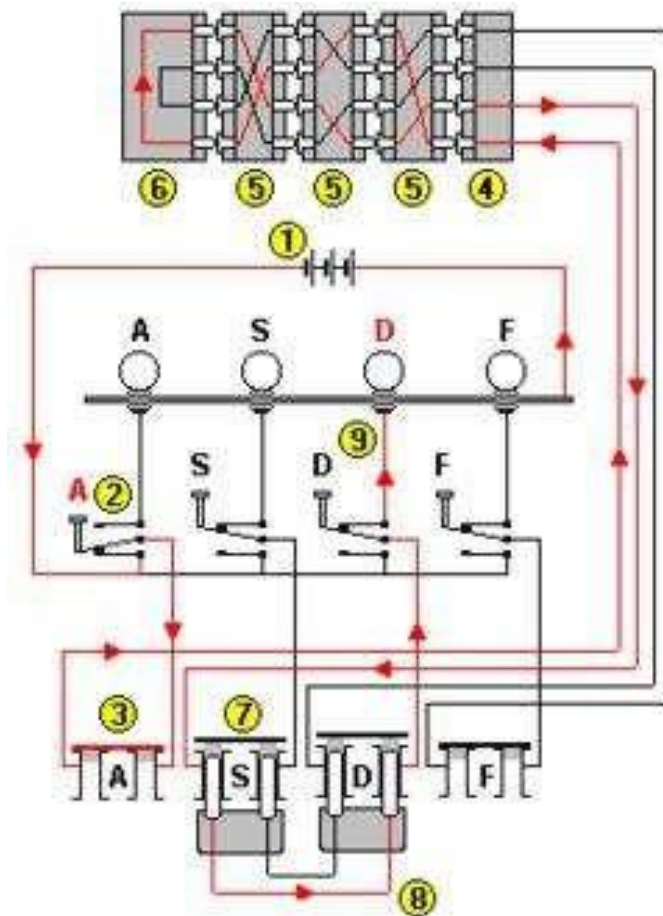


Fig. 3.2. Enigma wiring diagram (replacing the letter 'A' with the letter 'D' in the text)

The encrypting effect of Enigma is shown for two keys pressed in succession: the current flows through the rotors, is "reflected" from the reflector, and then returns again through the rotors. The gray lines in the figure show other possible electrical circuits inside each rotor. The letter "A" is replaced in the ciphertext in different ways by successive key presses, first by "G", then by "C". The signal takes a different route by turning the rotor.

Like other rotary machines, Enigma consisted of a combination of mechanical and electrical systems. The mechanical part included a keyboard, a set of rotating disks (rotors) located along the shaft, and a stepped mechanism that drives one or more rotors with each keystroke. The movement of the rotors results in different variations of character substitution each time a key is pressed on the keyboard.

The mechanical parts moved to form a changing electrical circuit—that is, in effect, the encryption was done electrically. When the keys were pressed, the circuit was closed, the current passed through various components, and finally turned on one of the many light bulbs that displayed the output letter. For example, when encrypting a message beginning with "ANX...", the operator first pressed the "A" button, and the "Z" light bulb lit up, that is, the "Z" became the first letter of the cryptogram. The operator continued to encrypt by pressing "N" on the keypad, and so on, until the end of the original message.

The constant change in the electrical circuit through which the current flowed, due to the rotation of the rotors, made it possible to implement a multi-alphabetic substitution cipher, which gave high resistance for that time.

Rotors



Fig. 3.3. Left side of the Enigma rotor, planar electrical contacts are visible



Fig. 3.4. Right side of the rotor, pin contacts are visible

Rotors are the heart of Enigma. Each rotor is a disc about 10 cm in diameter, made of solid rubber or bakelite, with spring pin contacts on one side of the rotor, arranged around the circumference; on the other side, there is an appropriate number of flat electrical contacts. The pin and flat contacts correspond to the letters in the alphabet; usually it is 26 letters "A"..."Z". When contacting, the contacts of adjacent rotors close the electrical circuit. Inside the rotor, each pin contact is connected to some flat contact. The order of connection may vary.

The rotor itself reproduces encryption by simply replacing the symbols. For example, the pin responsible for the letter "E" can be connected to the pin of the letter "T" on the other side of the rotor. But when using non-

how many rotors in a bundle (usually three or four), due to their constant movement, a more stable type of multi-alphabetic encryption is obtained.

The rotor can occupy one of 26 positions in the machine. It can be turned manually using a knurled finger wheel that protrudes outwards, as shown in Fig. 3.5. In order for the operator to always be able to determine the position of the rotor, there is an alphabetic ring on each rim; one of the letters is visible through a window. In early Enigma models, the alphabet ring was fixed; later versions introduced a sophisticated design with the possibility of adjusting it. Each rotor contains a notch (or several notches) used to control the movement of the rotors.

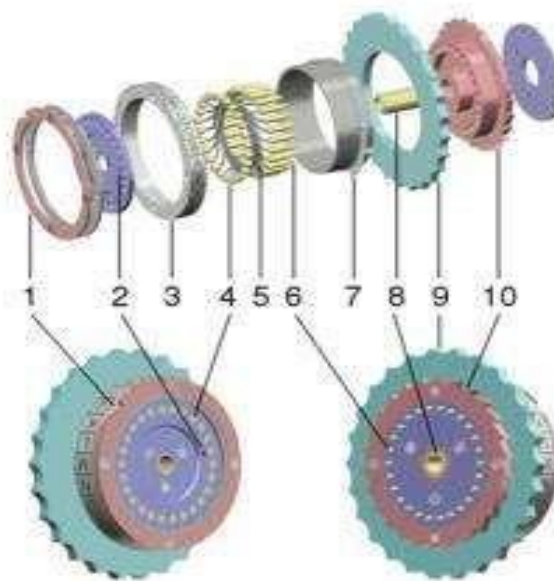


Fig. 3.5. Disassembled rotor

1. Ring with notches
2. marking point for pin "A"
3. Alphabet Ring
4. Contacts
5. Electrical wiring
6. Pin contacts
7. Spring lever for ring adjustment
8. Bushing
9. Finger Ring
10. ratchet wheel

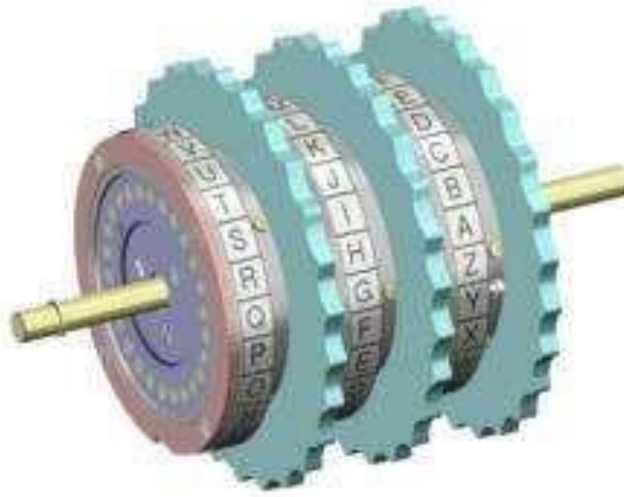


Fig. 3.6. Three rotors connected in series

Military versions of the Enigma were produced with several rotors; the first model contained only three. In 1938 there were five, but only three of them were used simultaneously in the machine. These types of rotors were marked with the Roman numerals I, II, III, IV, V, and all with one notch located at different points in the alphabet ring.

In the *Wehrmacht Enigma*, each rotor is attached to an adjustable ring with notches. The five base rotors (I-V) had one notch each,

while naval machines with additional rotors (VI-VIII) have two notches. At a certain point, the recess falls in front of the pawl, allowing it to hook the ratchet of the next rotor when the key is subsequently pressed. When the pawl does not fall into the recess, it simply slips along the surface of the ring without catching the gears. In a single-notch system, the second rotor moves forward one position while the first moves 26 steps. Similarly, the third rotor moves one step while the second moves 26 steps. A special feature was that the second rotor also turned if the third rotor turned; this means that the second rotor could turn twice with two successive key presses, the so-called "two-step movement", which leads to a shorter encryption period.

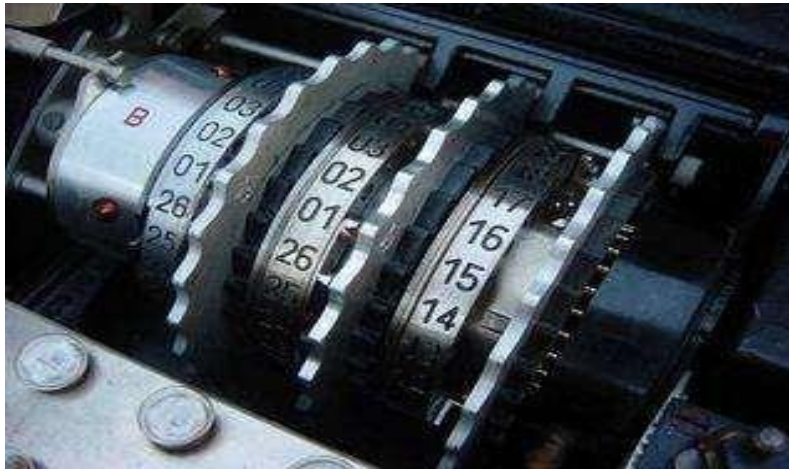


Fig. 3.7. Enigma rotors in the assembled state. Three movable rotors are placed between two fixed parts: the inlet ring and the reflector (marked "B" on the left)

Input Wheel

The input wheel (*Eintrittswalze* in German), or input stator, connects the patch panel, or (if it is not available) the keyboard and the lamp panel, to the rotors. Although the fixed connection of the wires is relatively of little importance from the point of view of safety, it proved to be a bit of an obstacle in the work of the Polish cryptanalyst Marian Regzowski when he tried to determine how the wires were switched inside the rotors. The commercial version of Enigma connected the letters in the order in which they appeared on the keyboard: QA, WB, EC, and so on.

Reflector

With the exception of the early A and B models, the last rotor was followed by a reflector (*Umkehrwalze* in German), a patented feature that distinguished the Enigma family from other rotary machines developed at the time. The reflector connects the contacts of the last rotor in pairs, switching the current through the rotors in the opposite direction, but along a different route. The reflector ensures that the Enigma conversion is an involuntarily

i.e. the decryption process is symmetrical to the encryption process. In addition, the reflector gives Enigma the property that no letter can be encrypted by itself. This was a serious conceptual flaw, which was subsequently exploited by codebreakers.

Patch Panel



Fig. 3.8. Patch panel at the front of the machine. Up to 13 connections could be used. In the photo, two pairs of letters are switched ("S"-"O" and "J"-"A")

The patch panel, which allowed the operator to vary the connections of the wires, first appeared in German military versions in 1930 and was soon successfully used in naval versions. The patch panel contributed immensely to the complexity of the Enigma cipher, even more so than the introduction of an additional rotor. With Enigma without a patch plate – the "*unsteckered*" Enigma – the decoder could be manipulated almost manually, but these methods proved helpless when a patchboard was added, and the crackers were forced to construct special decryption machines. A cable placed on the patch panel connects the letters in pairs, for example, "E" and "Q" can be paired together. The effect is to rearrange these letters before and after the signal passes through the rotors. For example, when the operator pressed "E", the signal was routed to "Q" and only then to the input rotor.

Accessories

A handy part used on the M4 Enigma was the "*Schreibmax*", a small printing device that could print all 26 letters on a small sheet of paper.

Enigma Procedures

During the Second World War, German operators used the cipher book only to set the rotors and set the rings. For each message, they selected a random starting position, acceptable "WZA", and a random message key, such as "SXT". Next, the operator set the rotors to the starting position "WZA", and encrypted the message key "SXT". Suppose that the key encryption results in "UHL". Next, the operator set the message key "SXT" as the starting position of the rotors and encrypted the message. The operator then sent the start position "WZA" and the encrypted key "UHL" along with the message. The recipient set the starting position in accordance with the first three-gram "WZA" and, deciphering the second trigram "UHL" Give answers to control questions in the report, in accordance with the number of the variant indicated by the teacher.

Recognized the original "SXT" key. Next, the recipient used this key as a starting point to decrypt the message. Usually, the keys were valid for one day.

The military version of Enigma used only 26 letters. Other characters were replaced by rare combinations of letters. The space was omitted or replaced with an "X". The "X" symbol was also used to indicate the point or end of a message. Some special characters were used

in some armed units, for example, *the Wehrmacht* replaced the comma with the two characters "ZZ" and the question mark with "FRAGE" or "FRAQ".

And *Kriegsmarine* replaced the comma - "Y" and the question mark - "UD". Two, three or four zeros were replaced by CENTA, MILLE and MYRIA respectively.

When performing laboratory work, the Enigma3S emulator program is used to study the Enigma3S cipher .

2. Task

1. Launch the Enigma3S Enigma *Emulator* from the folder specified by the instructor. Consult the help file: menu option

Help=>Help. In the program menu, select Settings => Reset, Fig.3.9.

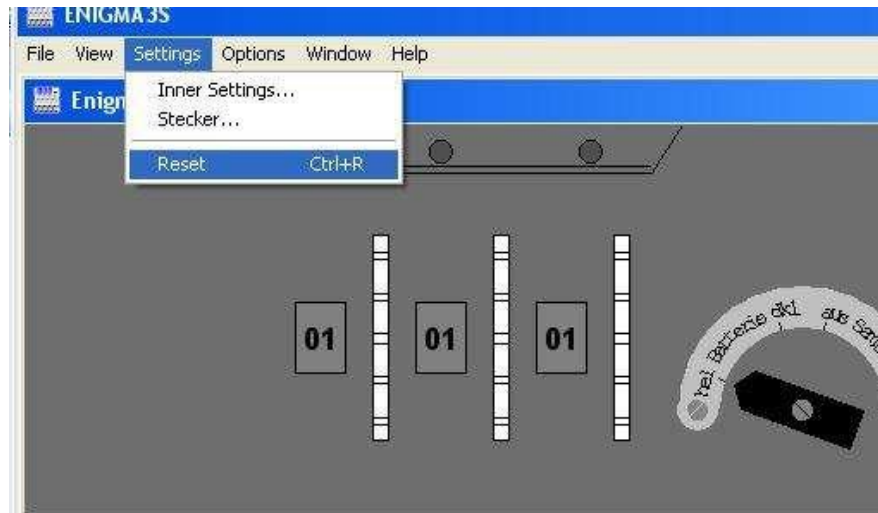


Fig. 3.9. Initial Enigma Emulator Settings

3. Set the values for rings 01 01 01 by selecting the menu item

View=>Open Cover. In the Settings=>Inner Settings menu, set the following values: Reflector – B, Left – I, Middle – II, Right – III, Ringstellung – A-A-A (Let's consider this as the starting position),

Fig.3.10, 3.11.

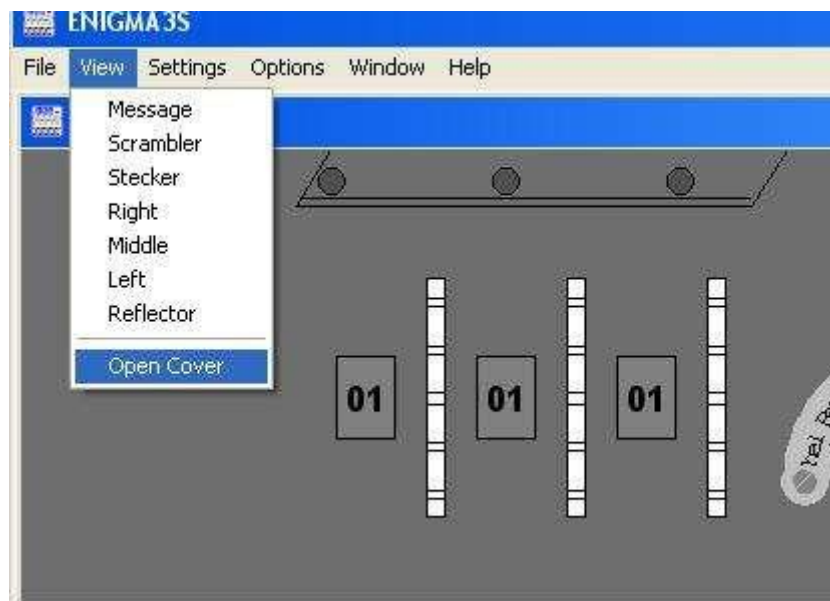


Fig. 3.10. Viewing the initial settings

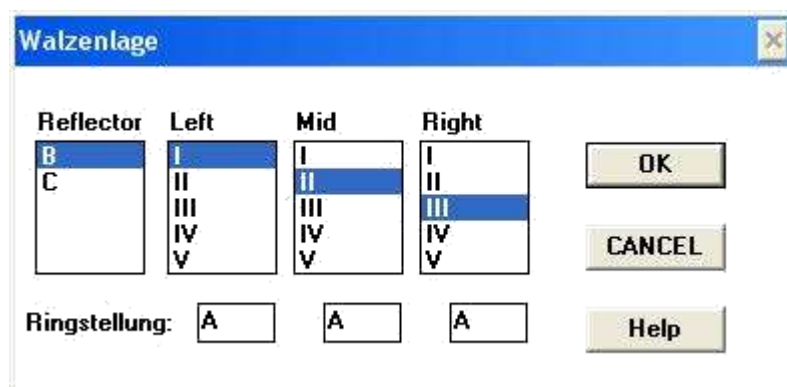


Fig. 3.11. Reflector, Left, Mid, Right Settings

4. Enter the message "SECRET MESSAGE" on the Enigma keyboard. What message is received at the output?
5. Reverse step 3 by changing the settings from Ringstellung – A-A-A to Ringstellung – A-B-C
6. Enter the message "SECRET MESSAGE" on the Enigma keyboard. What message is received at the output? How different is it from the message received in point 4?
7. Save the ciphertext obtained in step 6 using the menu option File=>Save CT as, Fig. 3.12.



Fig. 3.12. Saving Ciphertext

8. Create a .pln file in the root of the emulator folder, open it for editing in Notepad and write the open text to encrypt it.

9. In the program menu, select the File=>Open Pt option and select the file created in step 9. Get the ciphertext from the plaintext by selecting the Options=>Encipher Text menu option.

10. Set the emulator to the initial position. Select Window=> Scrambler from the program menu. Use the keyboard to enter a 22-character message (as shown in *Table 3.1*), noting the position of the rings. Enter the 22-character sequence again. How has the position of the rings changed? Is the new encrypted sequence different from the initial sequence? Why has the output sequence changed?

11. Using the Scrambler window, observe how the scythrottext changes depending on the position of the contact wheels. What makes it possible to adjust the order of the contact wheels?

In the program window, select the View=>Right option, Fig. 3.13.

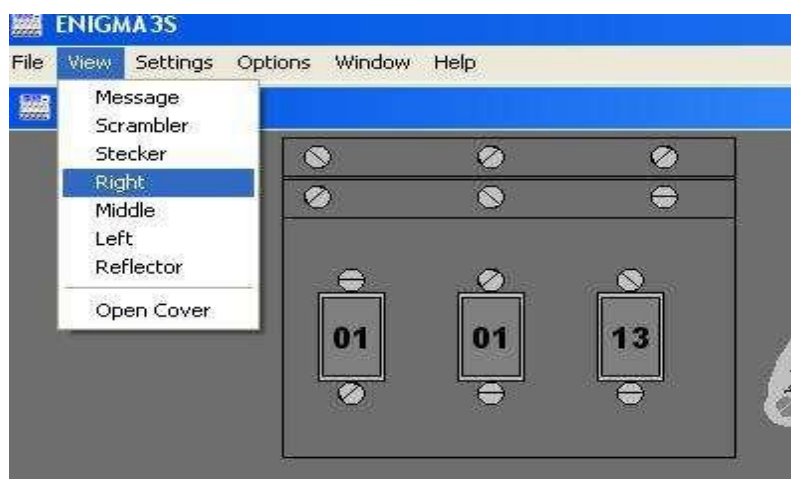


Fig. 3.13. Viewing the settings for the task

Fill in the correspondence table for 5 letters, the options are indicated in Table 3.1) for the first 12 angular positions of the right wheel.

Table 3.1

Letters on Entran ce	Angular wheel positions											
	11	22	33	44	55	66	77	88	99	110	111	112

13. Knowing that the emulator is set to the initial position, decipher the text in accordance with the version number indicated by the teacher (the varieties of the control tasks are indicated in *Tables 3.2, 3.3, 3.4*).

Table 3.2

Variant Numbers	Original alphabet
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

Table 3.3

Variant Numbers	Ciphertext
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

Table 3.4

Variant Numbers	Ciphertext
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

Laboratory Work No5
Generation of prime numbers used in asymmetric systems
Encryption Max

The aim of the research is to study the generation of prime numbers used in public-key encryption systems and the ability to draw independent conclusions based on the results of research.

2. Emerging competencies

This laboratory work is aimed at the formation of the following competencies of a bachelor's graduate:

OPK-1: ability to install software and hardware for information and automated systems;

PC-5: the ability to interface hardware and software as part of information and automated systems.

3. Theoretical part

Asymmetric encryption systems

Asymmetric cryptosystems (public key systems) – the meaning of these cryptosystems is that different transformations are used for encryption and decryption. One of them – encryption – is absolutely open to everyone. The other – decryption – remains secret due to the secrecy of the decryption key. Thus, anyone who wants to encrypt something uses an open transformation. But only the one who owns the secret key can decrypt and read it.

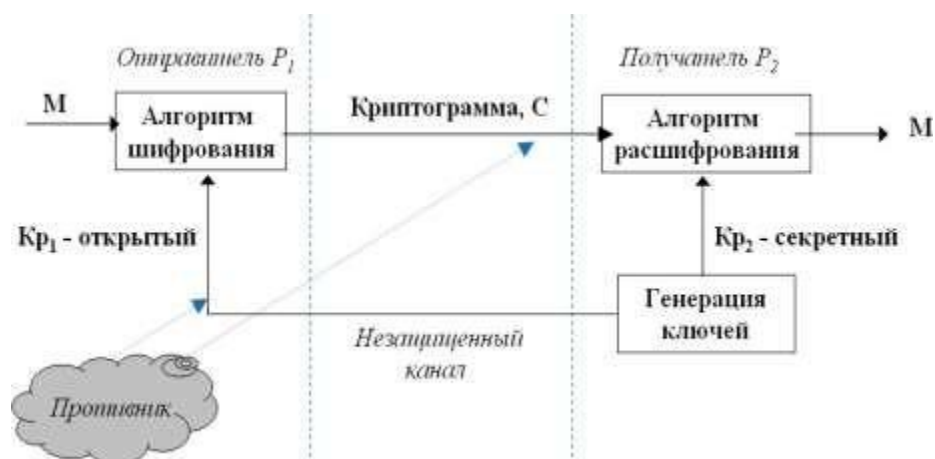


Fig. 1.1. Generalized diagram of an asymmetric cryptosystem

At the moment, in many asymmetric cryptosystems, the type of transformation is determined by the key. A user has two keys – a secret key and a public key. The public key is published in a public place, and anyone who wants to send a message to this user encrypts the text with the public key. Only the mentioned user with the secret key can decrypt it. Thus, the problem of transmitting the secret key, as in symmetric systems, disappears. However, despite all their advantages, these cryptosystems are quite time-consuming and slow. The strength of asymmetric cryptosystems is based mainly on the algorithmic difficulty of solving a task in a reasonable time. If an attacker manages to build such an algorithm, then the entire system and all messages encrypted with this system will be discredited. This is the main danger of asymmetric cryptosystems in contrast to symmetric ones.

Diffie-Hellman algorithm

The Diffie-Hellman algorithm uses a discrete exponentiation function. First, two large prime numbers, n and q , are generated. These two numbers do not have to be kept secret.

partners *P1* generates a random number x and sends the value to another participant in future *P2* exchanges

$$A = qx \bmod n$$

Upon receipt of A , partner *P2* generates a random number y and sends the computed value to the exchange participant *P1*

$$B = qy \bmod n$$

Partner *P1*, having received B , calculates $Kx = Bx \bmod n$, and partner *P2* calculates $Ky = Ay \bmod n$. The algorithm ensures that the numbers Ky and Kx are equal and can be used as a secret key for encryption. After all, even if the numbers A and B are intercepted, it is difficult to calculate Kx or Ky . Schematically, the work of the Diffie-Hellman algorithm is shown in Fig. 1.2.

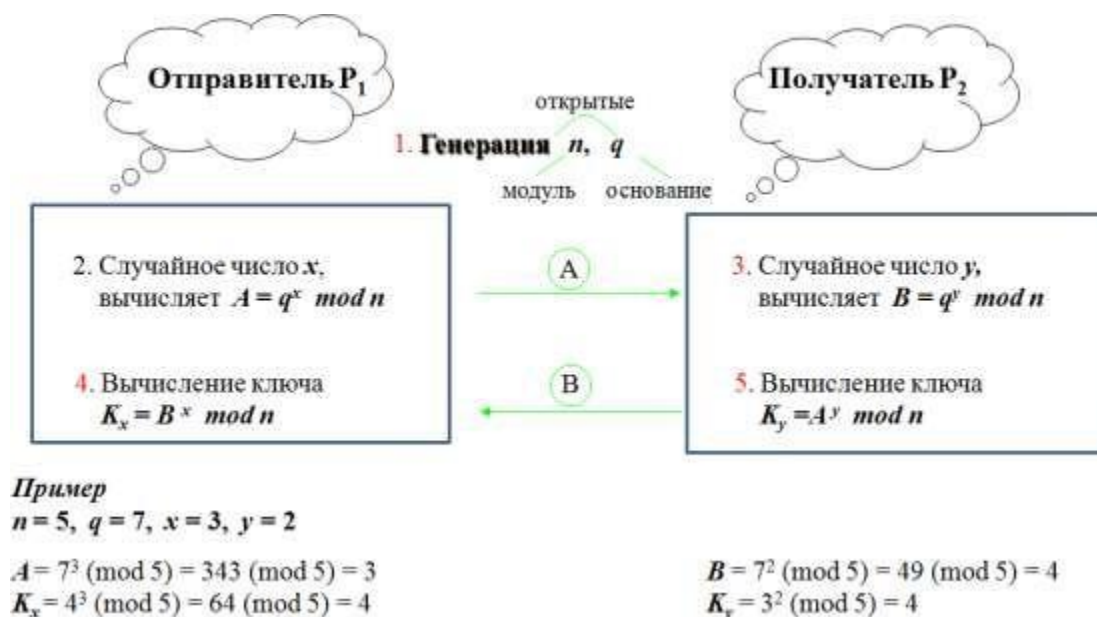


Fig. 1.2. Diffie-Hellman algorithm

RSA Algorithm

The first practical embodiment of the principle of open encryption was in the *RSA system*, developed in 1977 at the Massachusetts Institute of Technology (USA) and named after the first letters of the authors' surnames: *R. Rivest, A. Shamir, L. Adleman*.

The authors' idea was that by taking an integer N as the product of two large prime numbers $N = P * Q$, it is easy to select a pair of numbers Y and X , such that for any integer M less than N , the ratio is true:

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

The public encryption key in *RSA* is the *key* Y and the *module* N , and the secret key for decrypting messages is *the number* X .

The procedure for encrypting message M , considered as an integer (this assumption is possible due to the fact that any content can be represented in numerical form when processed in computer technology), less than N (if necessary, a long message is divided into segments that can be encrypted independently), consists in calculating the value of:

$$C = MY \bmod N$$

Decryption is carried out in the same way using the secret key X :

$$M = CX \bmod N$$

Mathematically, it can be strictly proved that the determination of *the secret key* X by a pair of numbers (N, Y) is no simpler than the prime factorization of the number N , that is, the finding of P and Q . The problem of factorization of an integer has been studied in mathematics since ancient times and is known as a complex computational problem. At the moment, the expansion of a number of several hundred decimal places will require hundreds of years of continuous operation from modern computers.

Schematically, the *RSA* algorithm is shown in Fig. 1.3.

Генерация ключей

Получатель 1. P, Q - простые, $N = P \cdot Q$
2. $\varphi(N) = (P-1) \cdot (Q-1)$, $\varphi(N)$ - функция Эйлера

Выбор открытого ключа Y :

$$1 < Y \leq \varphi(N), \text{НОД}(Y, \varphi(N)) = 1$$

Вычисление секретного ключа X :

$$X \cdot Y \equiv 1 \pmod{\varphi(N)}$$

$(N, Y) \rightarrow \text{отправителю}$

Отправитель шифрование M ($M_1 = 0, 1, 2, \dots, N-1$)
3. $C_1 = M_1^Y \pmod{N}$

Получатель расшифрование C ($C_1, C_2, \dots, C_p, \dots$)
4. $M_1 = C_1^X \pmod{N}$

Пример

Генерация ключей

1. $P = 3, Q = 11, N = P \cdot Q = 33$
2. $\varphi(N) = (P-1) \cdot (Q-1) = 2 \cdot 10 = 20$
 $Y = 7, \text{НОД}(Y, \varphi(N)) = 1$
 $X \cdot Y \equiv 1 \pmod{20}, 7 \cdot 3 \equiv 1 \pmod{20}, X = 3$

Сообщение: $M_1 M_2 \rightarrow 32; M_1 = 3 < 33, M_2 = 2 < 33$

Шифрование $C_1 = M_1^Y \pmod{N}$

3. $C_1 = 3^7 \pmod{33} = 2187 \pmod{33} = 9$
 $C_2 = 2^7 \pmod{33} = 128 \pmod{33} = 29$

Шифротекст 9,29

Расшифрование $M_1 = C_1^X \pmod{N}$

4. $M_1 = 9^3 \pmod{33} = 729 \pmod{33} = 3$
 $M_2 = 29^3 \pmod{33} = 24389 \pmod{33} = 2$

Восстановленный текст 3,2

Fig. 1.3. RSA algorithm

1.1. Methods for Checking Numbers for Simplicity

One of the main problems of asymmetric encryption is the generation of large prime numbers. The simplest method for checking the primality of a natural number N is the method of trial divisions: for $d=2, 3, 4 \dots$ we check the fulfillment of condition $(d, N) > 1$ (here (d, N) is the largest common divisor of the numbers d and N). The number of operations required for this method is of the root order of N .

Therefore, it is not applicable for numbers of the order of 1030–1040.

In contrast to such "deterministic" tests, there are also "probabilistic" tests for checking the simplicity. For the number under study, the

If any of these conditions are not met, then N is composite.

If all the conditions are met, then with some probability we can say that N is a prime number. This probability is closer to 1 the more random numbers we check.

Normally, these conditions are based on Fermat's small theorem, which states that for any positive number b not greater than some prime number p :

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

For example, $26 = 64 = 63 + 1 = 1(\text{mod } 7)$. If you want to determine whether an integer r is prime, you can choose any positive integer b less than r and check if the equation is true

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

If the equation is not satisfied, then on the basis of Fermat's theorem, we can be absolutely sure that r is not a prime number. If the equation holds, then we can only assume that r is a prime number and therefore call it "pseudoprime on base b ". The probability $P(x)$ that the composite number x will be pseudoprime on a random basis decreases with the increase of x .

Unfortunately, there are so-called Carmichael numbers – composite numbers that have the property:

$$b^{(r-1)} = 1(\text{mod } r) \text{ for all } b \text{ of the interval } [1, r] \text{ that are reciprocal with } r.$$

An example of a Carmichael number is the number $561 = 3 \cdot 11 \cdot 17$.

The classic result of number theory, Chebyshev's theorem, shows that the fraction of positive integers smaller than an integer m that are prime is close to $1/(\ln m)$. For example, the fraction of integers less than 10100 that are prime is close to $1/(\ln 10100) = 1/230$.

Thus, if we choose a randomly large integer positive odd number x and sequentially check for primeness $x, x+1, x+2, \dots$, then, on average, we will encounter a prime number for the first time in the step numbered $\ln x$.

4. Equipment and materials

To perform the work, you will need:

- class of personal computers not lower than Pentium-4, blackboard, chalk;
- Microsoft Windows operating system, software

Remedy

L_PROST.EXE.

5. Safety Instructions

Safety requirements before starting work:

- DO NOT: change clothes, use fire, smoke, eat in the laboratory;
- make sure that electrical outlets and connectors are intact. It is forbidden to work with a faulty power tool, break it and repair it yourself;
- in the laboratory, you must wear a change of shoes;
- turn on the computer only after obtaining permission for the work performed and permission from the teacher or laboratory assistant.

Safety requirements during work:

- when performing laboratory work, students are required to use only computer equipment, peripheral equipment, connecting cables, measuring equipment and diskettes directly related to this laboratory lesson;

- Connection and disconnection of the components of the computing complex shall be carried out only when the voltage is completely removed from all components of the computing complex.

Safety requirements at the end of work:

- at the command of the teacher, students are obliged to turn off the computer and the components of the computer complex;

- put the workplace in order.

Description of the laboratory work

To perform laboratory work, you need to start the L_PROST.EXE program . A window appears on the display screen with a text editor placed in its center (for displaying information about the actions performed by the program), in the top bar of the window there is the main menu, just below the main menu there is a toolbar (for controlling quick command buttons and other "hot" controls), and at the very bottom of the window, under the text editor, there is a status bar in which a hint is indicated and additional information is displayed. The toolbar keys, for convenience, are equipped with tooltips.

In order to get to the main menu, you need to press the F10 keyboard. Moving through the main menu is carried out by moving the cursor keys. To call a menu item, you need to press the "ENTER" key, "ESC" - exit the main menu.

For convenience, the program provides work with the mouse. In this case, the pointer is brought to the desired item of the main menu or to the desired button on the toolbar and the left mouse button is pressed, to refuse, it is enough to press the "ESC" key.

In addition to the main functions of the main menu ("Generate a simple P", "Search in an interval", "Check for simplicity", and so on), the toolbar contains a key that displays information about the program.

Generating a simple P

The ability to generate a prime number; the number of digits of the generated number is set by the user (from 1 to 5).

Search in the interval

Ability to search for prime numbers in a given interval. The user sets: the beginning of the interval is the value of x , then the length of the interval is the value of L . The search will be carried out in the interval $(x, x+L)$.

When checking for the primality of each number of the interval, the test of trial divisions for the first primes in order is performed first, and then the check according to the Fermat test. To specify the methods of checking for simplicity, you need to select the checkbox next to the name of the desired method with the left mouse button, and then specify all the necessary data to start the search.

In the method of trial divisions, the initial data is the number of the first prime numbers to be divided, and in the Fermat test, the number of bases and their values must be indicated.

At the end of the calculation, the found prime numbers and their number will be displayed on the screen. The full picture of the results of the work can be viewed in the menu item "Output of results".

Checking for simplicity

The ability to check the primality of any number. It is necessary to enter the number and calculation parameters in the same way as searching in the interval.

Output of results

Ability to view all the results of the last processing. When you log in to the program when no calculations have yet been made, you can view the source file of the first primes used for the test divisions.

More information about the program

1. When the utilities for generating a prime number, searching in the interval and checking for simplicity are launched, the user is asked to confirm the correctness of the selected method for work.

2. During the operation of long-term procedures, the progress of the process is started and the text editor window is extinguished. By the progress bar, you can observe and evaluate the approximate speed of the algorithmic rhythm and the time of the end of the current process.

3. Be careful when setting the operating parameters, as these parameters cannot be changed during the calculation as you work.

4. Description of "hotkeys":

Ctrl+F1 - 'Generate a simple P'
Ctrl+F2 - 'Search in an interval'
Ctrl+F3 - 'Check for simplicity'
Ctrl+F4 - 'Output results' Ctrl+X -
'Exit the program'

5. In laboratory work, due to the long counting time, it is recommended to use numbers of no more than five digits and the length of the interval to choose no more than 500, the number of bases for the Fermat test is no more than 5.

6. For the program to function properly, the *prost.txt* and *work.txt* files must be in the working directorate (together with the *l_prost.exe* file).

6. Tasks

Task No1.

1. Check for primality two arbitrary integers with a bit of at least 5.
2. Distribution of prime numbers.

2.1. An interval of the form $[x, x+L]$ is given. Calculate the number of $P(x, L)$ of simple numbers in the interval and compare it with the value of $L/\ln(x)$. Under what conditions is $P(x, L)/L$ close to $1/\ln(x)$?

$x=2000, L=500,$

- Number of prime numbers to divide: 5-15;
- Number of bases: 1-2.

2.2. Find all the primes in a given interval. Let $L(i)$ be the difference between two adjacent primes. Construct a histogram for $L(i)$. Calculate the sample mean of L media. Compare with the value of $\ln(x)$, where x is the middle of the interval.

Interval: (1000, 1000+300):

- Number of prime numbers to divide: 5-20;
- Number of bases: 1-3.

2.3. For a given set of numbers $\{k\}$, estimate the relative error of the formula for *the* k prime:

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

3. Methods for generating prime numbers.

3.1. In a given interval, plot the relative number of natural numbers that pass through the "sieve of Eratosthenes" (i.e., not divisible by the first k primes).

Interval: (500, 500+200).

The calculation is made for all $k \leq 10$.

3.2. For a given interval:

a) calculate the exact number of $P0$ prime numbers in the interval, i.e. when checking, set only the divisibility test.

The number of the first prime numbers to be divided is determined from the calculation: the maximum number to be divided is equal to the square root of the maximum value of the interval.

b) compose a test with a small number of trial divisions and one base in the Fermat test. Calculate the number of probably simple numbers $P1$ that satisfy this test.

c) compose a test with a greater number of trial divisions than in the previous case and two or three bases in the Fermat test. Subtract the number of *probably prime numbers* $P2$ that satisfy this test.

Interval: (1500, 1500 + 300).

Analyze the data obtained.

3.3. It is known that there are Carmichael numbers in a given interval.
Find them. Variants of intervals: (1050, 1050 + 100) (1700, 1700 + 100)
(2400, 2400 + 100)

4. Give answers to control questions in the report, in accordance with the number of the option specified by the teacher.

7. Contents of the report

The report must contain:

1. Name of the work, purpose.
2. The sequence of actions in the performance of each task.
5. Conclusions on the implementation of laboratory work.

8. Control questions

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: $1812 \pmod{13}$, $127 \pmod{7}$.
12,14,16 21,23,25, 29	Сформулируйте суть теста на простоту с использованием пробных делений.

9. List of references recommended for use on this topic

1. Zhuk, A.P., Zhuk, E.P., Lepeshkin, O.M., Timoshkin, A.I. Information Protection: Textbook for Bachelors [Text] / A.P. Zhuk et al. – M.: INFRA-M, 2015. – 392 p.
2. Shangin, V.F. Zashchita komp'yuternoy informatsii [Protection of computer information]. Effective methods and means [Text] / V.F. Shangin – Moscow: DMK Press, 2012. – 544 p.

Lab No 6 Electronic Digital Signature

The purpose of the work is to study the principles of secure electronic document management in telecommunication networks, algorithms for placing an electronic digital signature (EDS) and the ability to make independent conclusions based on the results of research.

2. Emerging competencies

This laboratory Work sent by on formation of the following competencies of a bachelor's graduate:

OPK-1: the ability to install software and hardware for information and automated systems;

PC-5: the ability to interface hardware and software as part of information and automated systems.

3. Theoretical part

When exchanging electronic documents over a communication network, the costs of processing and storing documents are significantly reduced, and their search is accelerated. But at the same time, there is a problem of authentication of the author of the document and the document itself, i.e. establishing the authenticity of the author and the absence of changes in the received document. When processing documents in electronic form, traditional methods of establishing authenticity by a handwritten signature and a seal impression on a paper document are completely unsuitable. The first EDS scheme - *RSA* - was developed in the late 1970s. However, the problem of confirming authorship has become relevant.

It is for this reason that the digital signature standards in Russia and the United States were adopted practically simultaneously, in 1994. Of the digital signature schemes proposed by cryptologists, *RSA* and El-Gamal's scheme turned out to be the most successful. But the first of them was patented in the United States and a number of other countries (the patent for *RSA* expired quite recently). In the second scheme, there are a large number of its possible modifications, and it is very difficult to patent all of them. It is for this reason that the El-Gamal EDS scheme has remained largely free of patents. In addition, this scheme has certain practical advantages: the size of the blocks used by the algorithms, and, accordingly, the size of the EDS in it turned out to be much smaller than in *RSA*, with the same stability. That is why the EDS standards of Russia and the USA are based on the El-Gamal scheme. More than 60 countries have laws on EDS today. Russia is also on this list. The Law "On Electronic Digital Signature", adopted in 2002, should have a stimulating effect on the development of domestic e-commerce, especially if other regulatory legal acts are brought into line with it in a timely manner.

- The Government of the Russian Federation financially supports the implementation of the federal target program "Electronic Russia".
- The law "On Amendments to Article 80 of Part One of the Tax Code of the Russian Federation" was adopted.
- As early as January 2001, the Board of the Pension Fund by its resolution "On the Introduction of Cryptographic Protection of Information and Electronic Digital Signature in the System of the Pension Fund of the Russian Federation" regulated the

registration and connection of legal entities and individuals to the system of its electronic document management.

- In 2002, the Ministry of Taxes and Levies issued an order "On Approval of the Procedure for Submitting a Tax Return in Electronic Form via Telecommunication Channels", thanks to which today any individual or legal entity can contact the tax inspectorate using secure e-mail.

- In 2004, amendments to Articles 13 and 15 of the Law "On Accounting" were APPROVED BYd, according to which the accounting statements of the enterprise can be kept, stored and submitted to the regulatory authorities in electronic form.

The principle of EDS construction. The asymmetry of the roles of the sender and the recipient in EDS schemes requires the presence of two closely related keys: *a secret*, or signature key, and *a public*, or signature verification key. Any EDS scheme must define the following three algorithms:

- algorithm for generating a key pair for the signature and its verification;
- signature algorithm;
- signature verification algorithm.

The standards of Russia and the United States are very similar, they differ only in some numerical parameters and certain details of the key pair development, calculation and verification of the signature. Indeed, both standards are variants of the same El-Gamal EDS scheme. EDS is used to authenticate texts transmitted via telecommunication channels. Functionally, it is similar to an ordinary handwritten signature and has its main advantages:

- certifies that the signed text comes from the signatory;

- does not give the person the opportunity to waive the obligations associated with the signed text;

- guarantees the integrity of the signed text.

EDS is a relatively small amount of additional digital information transmitted along with the signed text, and includes two procedures:

- a signature procedure that uses the secret key of the sender of the message;

- A signature verification procedure that uses the sender's public key.

Signature procedure. When generating an EDS, the sender, first of all, calculates the hash function $h(M)$ of the signed text M . The calculated value of the hash function $h(M)$ is one block of information m , which characterizes the entire text M as a whole. Then the value m is encrypted with the sender's secret key. The resulting pair of numbers is the digital signature for this text *Moscow*.

Signature verification procedure. When verifying the EDS, the recipient of the message again calculates the hash function $m = h(M)$ of the text M received via the channel, after which he checks with the help of the sender's public key whether the received signature corresponds to the calculated value of *the m* hash function.

The fundamental point in the EDS system is the impossibility of forging the user's EDS without knowing his secret key.

Each signature typically contains the following information:

- date of signature;
- expiration date of the key of this signature;
- information about the person who signed the text;
- Signer ID (public key name)

- the digital signature itself.

Unidirectional hash functions. A hash function is designed to compress a signed document M to several tens or hundreds of bits. Hash function $h(.)$. It should be noted that the hash value of $h(M)$ is complexly dependent on document M and does not allow the document M itself to be recovered. A hash function must satisfy a number of conditions:

- should be sensitive to all kinds of changes in the text M ;
- must have the property of irreversibility, i.e. the task of selecting the M_1 , which would have the desired hash value, must be computationally unsolvable;
- The probability that the hash values of two different documents will coincide should be negligible. Most hash functions are based on the unidirectional function $f(.)$, which forms an output value of length n when two input values of length n are specified.

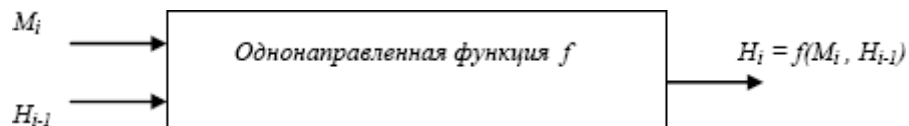


Figure 1. Hash function formation

DSA digital signature algorithm . Digital signature algorithm

DSA (Digital Signature Authorization) was proposed in 1991 in the United States and is a

development of the El-Gamal digital signature algorithm. The sender and recipient of an electronic document use large integers in the calculation:

G, P – prime numbers of L -bit each ($L = 512 \dots 1024$ bits); q – a prime number of 160 bits in length divisor of the number $(P - 1)$.

G, P, q are open and can be common to all network users.

Algorithm description

1. The sender selects a random integer X , $1 < X < q$. Number X is the sender's secret key for generating an electronic signature.

2. The sender calculates the value of

$$Y = GX \bmod P.$$

The Y number is the public key for verifying the sender's signature and is shared with all recipients of the document.

3. In order to sign a document M , the sender hashes it into an integer hash value m :

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

It then generates a random integer K , $1 < K < q$, and calculates the numeric number of the

Lo

R:

$$r = (GK \bmod P) \bmod q.$$

4. Using the secret key X , the sender calculates the number s :

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

A pair of numbers r, s form a digital signature $S = (r, s)$ under the document

Tom M .

5. The message delivered to the recipient, together with the signature, is a triple of numbers $[M, r, s]$. First of all, the recipient checks the fulfillment of the ratios:

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

6. Next, the recipient calculates the values:

$$w = 1/s \bmod q;$$

$$m = h(M) - \text{hash value}; u1$$

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

$$u2 = (r \cdot w) \bmod q.$$

Then, using the public key Y , the value of

$$v = ((Gu1 \cdot Yu2) \bmod P) \bmod q,$$

and the fulfillment of the equation $v = r$ is checked. If it is met, then the signature is recognized as authentic, since it can be strictly mathematically proved that the last equation will be fulfilled if and only if the signature $S = (r, s)$ under document M is obtained using the secret key X from which the public key Y was obtained.

New EDS standards. The latest achievements in cryptography have shown that the general problem of logarithm in discrete fields, which is the basis of the specified EDS scheme, cannot be considered a sufficiently solid foundation. For example, the size of blocks considered "safe" is growing at a relatively fast rate. As a result, this led to the fact that in 2001 the EDS standards of Russia and the USA were updated: they were transferred to elliptic curves. At the same time, EDS schemes remained the same, but the numbers they operate with are no longer the elements of the finite field $GF(2^n)$ or $GF(p)$, but elliptic numbers - solutions of the equation of elliptic curves over the specified finite

In the updated standards, the role of exposing a number in a finite field is performed by the operation of taking a multiple point of an elliptic curve – "multiplying" a point by an integer.

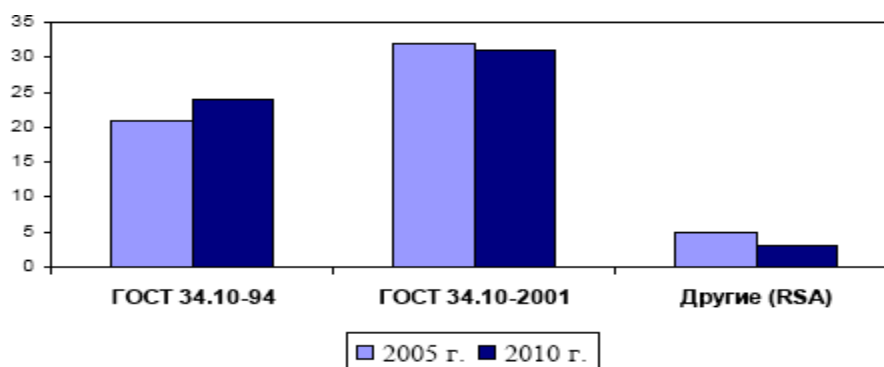


Figure 2. EDS algorithms used in the Russian Federation

The old Russian digital signature standard operates with 1024-bit blocks, and the new one, based on elliptic curves, operates with 256-bit blocks, and at the same time has greater stability. The strength of the GOST R34.10-94 signature scheme is based on the complexity of solving the problem of discrete logarithm in a simple field. In GOST R34.10-2001, the durability of the EDS schema is based on the complexity of solving the problem of discrete logarithm in a group of points of an elliptic curve. With the correct choice of curve parameters, the most effective methods for solving it are more labor-intensive r - and l -Pollard methods. Thus, according to various estimates of specialists, the labor intensity of breaking the old and new digital signature standards in Russia is about 10^{26} and 10^{38} multiplication operations in the basic $GF(p)$ field, respectively.

4. Equipment and materials

To perform the work, you will need:

- class of personal computers not lower than Pentium-4, blackboard, chalk;
- Microsoft Windows operating system, software Remedy
labWork6.exe

5. Safety Instructions

Safety requirements before starting work:

- DO NOT: change clothes, use fire, smoke, eat in the laboratory;
- make sure that electrical outlets and connectors are intact. It is forbidden to work with a faulty power tool, break it and repair it yourself;
- in the laboratory, you must wear a change of shoes;
- turn on the computer only after obtaining permission for the work performed and permission from the teacher or laboratory assistant.

Safety requirements during work:

- when performing laboratory work, students are required to use only computer equipment, peripheral equipment, connecting cables, measuring equipment and diskettes directly related to this laboratory lesson;
- Connection and disconnection of the components of the computing complex shall be carried out only when the voltage is completely removed from all components of the computing complex.

Safety requirements at the end of work:

- by team teacher Students Obligated turn off the computer and components of the computer complex;
- put the workplace in order.

6. Quests

Task No1. Get acquainted with the main areas of work within the framework of the Federal Target Program "Electronic Russia", as well as with the information on the procedure for using and existing algorithms for setting up electronic digital signatures, set out in paragraph 3. Launch a *labWork6.exe* **program** designed to demonstrate the procedure for setting and verifying an electronic digital signature.

Task No2. Generate and send to the participants of the exchange keys for encrypting the source document and keys for signing the document. The source text for encryption is typed directly in the program window.

Task No3. Encrypt the original message and sign it with the sender's secret key.

Task No4. Send the encrypted and signed message to the recipient. Check the correctness of the EDS and restore the original text of the message.

Task No5. Save in the report screen forms demonstrating the process of generating and distributing keys; the process of encrypting the original document and setting up an EDS.

Task No5. Give answers to control questions in the report, in accordance with the number of the option specified by the teacher.

7. Contents of the report

The report must contain:

1. Name of the work, purpose.
2. The sequence of actions in the performance of each task.
3. Conclusions on the implementation of laboratory work.

8. Control questions

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 (<i>RSA</i> , <i>El-Gamal</i> , <i>DSA</i>)

9. List of references recommended for use on this topic

1. Zhuk, A.P., Zhuk, E.P., Lepeshkin, O.M., Timoshkin, A.I. Information Protection: Textbook for Bachelors [Text] / A.P. Zhuk et al. – M.: INFRA-M, 2015. – 392 p.
2. Shangin, V.F. Zashchita komp'yuternoy informatsii [Protection of computer information]. Effective methods and means [Text] / V.F. Shangin – Moscow: DMK Press, 2012. – 544 p.

Lab No 7 Sliding Permutation Encryption

The purpose of the work is to study the sliding permutation cipher using the *XY-Move software implementation*. and the ability to make independent conclusions based on the results of research.

2. Emerging competencies

This laboratory Work sent by on formation of the following competencies of a bachelor's graduate:

OPK-1: the ability to install software and hardware for information and automated systems;

PC-5: the ability to interface hardware and software as part of information and automated systems.

3. Theoretical part

Stable Patterns of Plaintext and Their Use in Deciphering Ciphers of Simple Substitution and Permutation

The possibility of deciphering a cipher depends to a large extent on the extent to which cryptographic transformations destroy the probabilistic-statistical regularities present in the open content text. Thus, in meaningful texts of any natural language, different letters occur with different frequencies, while the relative frequencies of letters in different texts of the same language are close to each other. The same can be said about the frequencies of pairs or triples of letters of plaintext. In addition, any natural language has the following

called redundancy, which makes it possible to "guess" the meaning of the message with a high probability, even if some of the letters in the message are not known.

Table 1. Relative Frequencies of Appearance of Letters of the Russian Alphabet in the Text

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	ь,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		

Similar tables are given in different books. They are obtained on the basis of frequency calculations on large volumes of plaintext. Taking into account, that For experiments, different source material is taken, the values of the values of the probability are slightly different from each other. If we order the letters in descending probabilities, we get the following variation series:

**O,E,A,I,N,T,S,R,V,L,K,M,D,P,U,Z,Y,B,I,G,CH,Y,H,ZH,YU,SH,C,SH,S
H, SH, SH, SH,**

E,F

For example, the word **SENOVALITR** contains 10 of the most common letters.

The frequencies of the signs of the alphabet depend not only on the language, but also on the character of the text. For example, in a text on cryptography, the probability of the letters Φ , III will be increased (because of the common words "cipher", "cryptography"). In some mathematical texts, the frequency of the letter Φ may be overestimated (because of the words "function",

"function", etc.). In standard text files, the most common symbol is the "space" character. The frequency diagram of the content of

is a stable characteristic of the text. It follows from the theory of probability that with sufficiently weak restrictions on the probability properties of a random process, the law of large numbers is valid, i.e. the relative frequencies of θ/N signs converge in probability to the values of their probabilities p_k

$$P\left\{\left|\frac{g_k}{N} - p_k\right| > \varepsilon\right\} \xrightarrow{N \rightarrow \infty} 0.$$

Permutation and simple substitution ciphers do not completely destroy the very-statistical properties of the open message.

When deciphering a text encrypted with a simple substitution cipher, the frequency characteristics of the plaintext are used. Namely, if we calculate the frequencies of occurrence of characters in ciphertext, order them in descending order, and compare them with the variational series of probabilities of the open text. text, then These Two Sequences will be close.

Most likely, the gap will be in the first place, followed by the following letters O, E, A, I.

Of course, if the text is not very long, then there is not necessarily a complete coincidence. O may be in second place, and E in third, but in any case, in the first and second rows, the same letters will be located not far from each other, and the closer to the beginning (the higher the probability of signs), the smaller the distance between the characters will be.

A similar picture is observed for pairs of adjacent letters (bigrams) of open text (the most common bigram of Russian plaintext is ST). However, in order to obtain a stable picture, the length of the analyzed consistency must be sufficiently large. On relatively small segments of plaintext, this picture is somehow blurred. A more stable characteristic of bigrams is the absence of meaningful text of some bigrams, as they say, the presence of forbidden bigrams,

probability equal to almost 0. Have you ever seen the bigram $\mathbb{B}$ in open text, or bigrams of the form: "vowel" $\mathbb{B}$; "space" $\mathbb{B}$? Knowledge and use of these features of plaintext greatly facilitates the decipherment of the cipher, permutation and substitution.

Permutation cipher. Suppose X is a set of open (content) texts in the alphabet I . The lengths of all possible plaintexts are multiples of the value T . The key set is a symmetric group of substitutions ST of degree T , for $g \in ST$ we define the encryption function fg by positing X for $(i_1, i_2, \dots, i_T) \in$.

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

Let's define fg on the remaining elements of X according to the rule: the text $x \in X$ is divided into segments of length T and each segment of length T is encrypted on the key g according to the above encryption law. The sequence made up of the letters of the images of the ciphered words is a ciphered text corresponding to the plaintext x and the key g . To encrypt a text of a length not a multiple of T , it is supplemented with letters up to a multiple of T .

Decrypt the permutation cipher. The ciphertext is recorded in a table with T columns. To restore the plaintext of the permutation cipher, we need to rearrange the columns in such a way that in strings
appeared Meaningful text.

Let's consider an example of decryption of an eight-column permutation cipher (Textbook "Principles and Methods of Information Protection" by G.V. Proskurin). Let the ciphertext look like this:

1	2	3	4	5	6	7	8
п	а	я		в		н	м
о	ч	ш	г		у		е
е	б	ж	л		е		о
м		ч		о	т	о	я
	е	г	е		у	с	ш
	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		

Let's compare the 8*8 table with the permutation of the columns, and put one at the intersection of the i-th row and the j-th column, if the j-th collar should follow the i-th after the reverse permutation. Our task is to restore the table that corresponds to the correct permutation of the columns.

Now let's attach one column to the other in pairs. If forbidden bigrams appear in some rows, then the columns cannot follow each other in the plaintext and the corresponding cell is crossed out. In our example, the 6th column cannot follow the 4th, because otherwise there will be two consecutive probes in the text in the first row.

For example, let's look at the 6th line. If the 4th column followed the 1st column, then there would be words beginning with B.

After looking at all the rows, we get the following table:

	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

If the text were longer and there were more lines, then in each line and in each column there would be exactly one uncrossed out cell and the permutation would be restored.

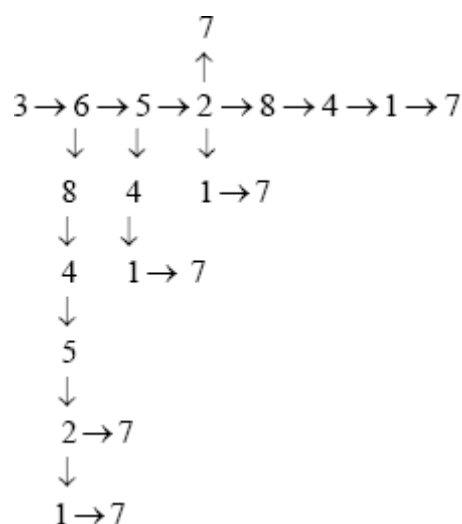
In our table, we can only state that the 6th column follows the 3rd column (let's denote this event as $3 \rightarrow 6$), if the 6th column is not the last one. For the 6th column, there can be two continuation options

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

We need to consider both and try to weed out the false variant. If it is not possible to weed out the false variant, then we must continue both variants

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

As a result, we get a tree of possible column sequences in the open text.



Each branch of the tree corresponds to a certain permutation of columns. Next, we check each variant for meaningfulness and get the correct variant.

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

Note that it was not necessary to build the tree to the end. For example, the branch $3 \rightarrow 6 \rightarrow 8 \rightarrow 4 \rightarrow 5$ could be eliminated immediately. Is it possible to recognize the following fragment of the text as meaningful?

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

Such a branch clipping procedure would be necessary if there were fewer rows, and the tree would be correspondingly much more branched.

The proposed procedure is easy to automate and make suitable for implementation on a computer. The decryption algorithm should consist of the following stages.

1. Preliminary work. Analyzing a fairly representative volume of open texts, build a lot of forbidden bigrams.

2. Preliminary work. Compile a dictionary of all possible v -grams for $v=2,3,\dots,d$ that can occur in plaintext. The number d is chosen based on the capabilities of computer technology.

Construct a table 8×8 . In this case, all the forbidden bigrams are sorted out sequentially and for each tested bigram - sequentially all rows. If in at least one row the first character of the bigram occurs in the i - column, and the second in the j -th column, then the $i \times j$ cell of the table is crossed out.

3. Select a column as the starting column.

4. Start the tree building procedure by attaching all column variations to the source column.

5. For each resulting variant, add one more of the remaining columns. If at least one of the rows in the table contains 3-grams, which is not in the dictionary of the placed 3-grams, then the variant is eliminated.

6. For each of the non-eliminated variants, add another column and filter out false variants according to the dictionary of allowed 4-grams.

If the dictionary was built only for $d \leq 3$, then the screening is carried out by checking the admissibility of the 3 grams found in the last 3 columns of each row. Continue this process until a complete permutation is obtained.

The table below shows the text restored for our example.

	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	ш	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	н		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	н	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					

To carry out the mixing of plaintext characters, individual text characters are stored in the encoder memory and their transmission is delayed in discrete time. Let us introduce the parameters n_1 and n_2 , so that $n = n_1 + n_2$. In these ciphers, the i -th character of the transmitted message is moved in the cipher message to the j -th place, where $i - n_1 \leq j \leq i + n_2$. The control sequence of the delay time is chosen so that the delay time of each symbol is a random variable with an equal distribution, that is, the probability of each fixed value of the delay time should be close to $1/n$.

Sliding Permutation Encrypting Machine

Let us consider the scheme of an encrypting automaton that makes it possible to implement an arbitrary shea of a sliding permutation with a suitable control sequence, Fig. 7.1.

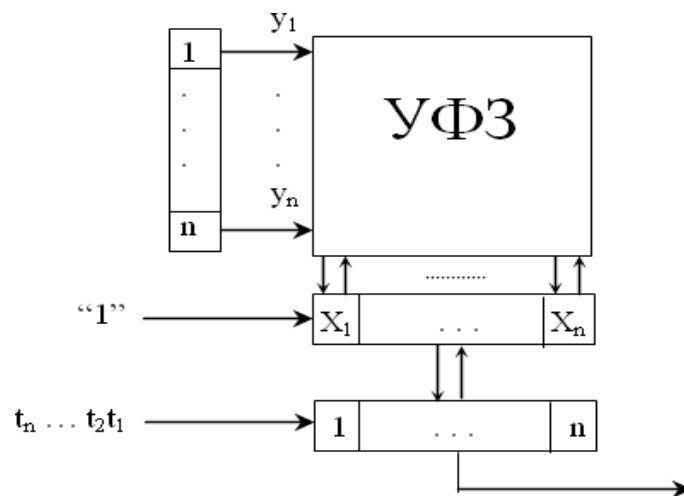


Fig. 7.1. Scheme of implementation of the encrypting sliding permutation machine

On the input of the delay generating unit (UVZ) in each clock t Vector is fed

The delay generation unit is the final automaton $m(F^n, F^n, \{1, \dots, n\}, \delta, \lambda)$, the set of Status which is A multitude of all possible binary vectors - fillings $x(t) = (x_1^t, \dots, x_n^t)$ lower commander.

In the tick t , the natural number is produced – the value of the delay.

$$\gamma_t = \max_{j \in \{1, \dots, n\}} \{x_j^t \mid y_i^t \in \{0, 1\}, i = \overline{2, n-1}, y_1^t = y_n^t = 1\}.$$

In this case, the circuit breaker switches to the following state:

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

где

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

$$x_i^{t+1} = \begin{cases} 0, & i = \gamma_t + 1 \\ x_{i-1}^t, & i \in \{2, \dots, \gamma_t, \gamma_t + 2, \dots, n\} \end{cases}$$

The plaintext characters are written to the lower storage device. The *plaintext character written in the cell with the number is sent* to the line in the t -th cycle.

The states of the $x(t)$ circuit breaker are indicators that show which of the plaintext characters have not yet been read from the lower drive.

To encrypt the sequence $tN, \dots, t2, t1$ proceed as follows. First, write the first $n1$ characters of the plaintext to the lower storage

$$(tn1, \dots, t1, 0, \dots, 0)$$

At the same time, the circuit breaker is set to its initial state

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

After that, the UFZ circuit breaker begins to operate according to the law described above until the last *plaintext character* tN is received in the storage device. With the cessation of the supply of plaintext characters to the storage device, the supply of units to the indicator storage device is stopped. In the remaining $n1$ cycles, the information recorded in the storage device is read.

When decrypting, the UVZ works according to the same scheme, only instead of reading, it is necessary to organize the recording of information in a second storage device.

Особенности работы УФЗ

В каждом такте t (за исключением последних n_1 тактов) в накопитель-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 $\{1, \dots, n\}$. В частном случае, когда $n=1$, либо $n_1=n$ УФЗ вырабатывает постоянно значения задержки $\gamma_i=1$ и $\gamma_i=n$ соответственно. Легко видеть, что результирующее преобразование открытого текста действительно будет шифром скользящей перестановки. Условие $y_1^t=1, t=1, 2, \dots$ обеспечивает постоянное считывание во всех тактах, а условие $y_2^t=1$ ограничивает величину задержки $\gamma_i \leq n$.

Example Explaining the Operation of a Sliding Permutation Encrypting Machine

Let's take $n=7, n_1=3, n_2=4$ as an example

A vector is fed to the input of the UFZ at each step t of the operation of the encrypting circuit breaker

$\vec{y} = (y_1^t, \dots, y_n^t)$ in the linear shift register (LRS), Fig. 7.2.

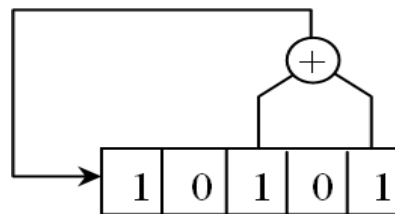


Fig.7.2. Linear shift register

We hope that the reader will be able to write the sequence of states of this linear shift register, with the help of which the control sequence of the cipher automaton is formed.

At each step of the encryption automaton, we will designate the sequence $Y_1, Y_2, \dots, Y_n$ (in our case, $Y_1, Y_2, \dots, Y_7$) coming from the LRS as

"Y", and the sequence of units $X_1, X_2, \dots, X_n$ (in our case, $X_1, X_2, \dots, X_7$) as "1". In the bottom line, we will write the plaintext characters located at this step in the lower storage device of the encryption machine $t_1, t_2, \dots, t_n$ (in our case, $t_1, t_2, \dots, t_7$). Let's consider the step-by-step operation of the encoder under specific conditions.

1-й шаг

"Y"	1	1	0	1	0	1	1
"1"	1	1	1	0	0	0	0
	t_3	t_2	t_1	0	0	0	0

2-й шаг

"Y"	1	0	1	0	1	0	1
"1"	1	1	0	1	0	0	0
	t_4	t_3	t_2	t_1	0	0	0

3-й шаг

"Y"	1	0	0	1	0	1	1
"1"	1	0	1	0	1	0	0
	t_5	t_4	t_3	t_2	t_1	0	0

4-й шаг

"Y"	1	0	0	0	1	0	1
"1"	1	0	0	1	0	1	0
	t_6	t_5	t_4	t_3	t_2	t_1	0

5-й шаг

"Y"	1	0	0	0	0	1	1
"1"	1	0	0	0	1	0	1
	t_7	t_6	t_5	t_4	t_3	t_2	t_1

6-й шаг

"Y"	1	1	0	0	0	0	1
"1"	1	1	0	0	0	1	0
	t_8	t_7	t_6	t_5	t_4	t_3	t_2

7-й шаг

"Y"	1	0	1	0	0	0	1
"1"	0	1	0	0	0	0	1
	0	t_8	t_7	t_6	t_5	t_4	t_3

8-й шаг

"Y"	1	0	0	1	0	0	1
"1"	0	0	1	0	0	0	0
	0	0	t_8	t_7	t_6	t_5	t_4

At each step, we searched starting from the left edge and going to the right for the first match in the rows "Y" and "1" (1 in both rows) and circled this column for convenience. The plaintext element that appeared in the selected column goes into the line. Thus, in our example, the sequence that went into the line looks like this:
 $t_2, t_4, t_5, t_6, t_1, t_7, t_3, t_8$

Description of the software implementation

To perform laboratory work on a computer, you need to install the XY-Mover software module.

1) Menu bar In this program, the menu consists of three items: Encryption, View, Help. You can select the desired menu item by clicking on it with the left mouse button, or using the keyboard buttons "right",

"left" by pressing the "F10" function key beforehand. After the user has selected the menu item he needs, a drop-down submenu will open. Let's take a closer look at the menu items.

- *Encryption.* The drop-down menu items can be selected either with the left mouse button or with the "up" and "down" keyboard buttons. Let's take a closer look at the submenu items.

- Parameter editing – this menu function allows you to set the necessary parameters in the "Encoder Settings" field of the program window;

- Encrypt – selecting this menu item starts the encryption process;

- Decryption – selecting this menu item starts the decryption process;

- The exit is the end of the program.

- *View.* This menu item allows you to choose the appearance of the program.

The following submenu items are listed:

- Options—Selecting this menu item allows you to use the

"Encoder Options" described below.

- Encoder diagram – selecting this menu item allows you to observe the structural diagram of the encrypting machine.

- Status Bar – Selecting this menu item allows you to see the status line described below.

2) Toolbar

It may be more convenient for the user to use the toolbar (the so-called "buttons") instead of working with the menu. "Buttons" duplicate some menu items, but it is much more convenient to select a "button" than a sub-menu item. To do this, simply click on the selected button with the left mouse button. Let's consider the "buttons" from left to right:

- "A", selecting this button allows you to get a result similar to the menu item:

View Options;

- "Encoder Scheme" – selecting this button allows you to get a result similar to the menu item View – Encoder Scheme;

- "Status bar" – selecting this button allows you to get a result similar to the menu item: View – Status bar;

- "Edit parameters" – selecting this button allows you to get a result similar to the menu item: Encryption – Edit parameters;

- "Encrypt" – selecting this button allows you to get a result similar to the menu item Encryption – Encrypt;

- "Decrypt" – selecting this button allows you to get a result similar to the Encryption – Decrypt menu item;

- "Interrupt" – selecting this button allows you to interrupt the data processing (encryption or decryption).

3) Status bar

The status bar is located at the bottom of the program window. It displays information about the status of the program:

- "Sliding permutation encryption" – This message indicates that the program is ready to work.

- "Completed ...%" – an indication of the amount of work done when encrypting or decrypting.

4) Description of the fields of the program window

- Encryptor parameters

- $n1$ is the number of characters that are written to the lower storage device first;
- $n2$ – other signs (there are 127 of them in total);
- register taps – LRS removal points;
- initial filling – the initial filling of the LRS, which can be changed by the user.

- Input Stream

- A field that displays the name of the text file whose content needs to be encrypted/decrypted.

- "Open" button – when you click on this button, the standard *Windows* window "Open File" opens;

- Text – the contents of the file opened for encryption/decryption;
- bitstream – bitwise representation of the characters of the selected file.

- Output Flow

- A field that displays the name of the text file, the content of which needs to be decrypted/encrypted.

- "Open" button – when you click on this button, the standard *Windows* window "Open File" opens;

- Text – the contents of the file opened for decryption/encryption;
- bitstream – bitwise representation of the characters of the selected file.

4. Equipment and materials

To perform the work, you will need:

- class of personal computers not lower than Pentium-4, blackboard, chalk;
- Operating Room System Microsoft Windows, Software Remedy
labWork6.exe

5. Safety Instructions

Safety requirements before starting work:

- DO NOT: change clothes, use fire, smoke, eat in the laboratory;
- make sure that electrical outlets and connectors are intact. It is forbidden to work with a faulty power tool, break it and repair it yourself;
- in the laboratory, you must wear a change of shoes;
- turn on the computer only after obtaining permission for the work performed and permission from the teacher or laboratory assistant.

Safety requirements during work:

- when performing laboratory work, students are required to use only computer equipment, peripheral equipment, connecting cables, measuring equipment and diskettes directly related to this laboratory lesson;
- Connection and disconnection of the components of the computing complex shall be carried out only when the voltage is completely removed from all components of the computing complex.

Safety requirements at the end of work:

- at the command of the teacher, students are obliged to turn off the computer and the components of the computer complex;
- put the workplace in order.

6. Quests

1. To perform laboratory work on a computer, you need to install the XY-Mover software module.
2. Perform the initial encoder settings according to the example.
3. Download the file to be encrypted.
4. Encrypt the information using a sliding permutation cipher, save the ciphertext in the file.

5. Describe in the report the process of encrypting and decrypting data using the XY-Mover emulator. Analyze the data obtained.

6. Include in the laboratory report the answers to the test questions selected according to the option number specified by the teacher.

7. Contents of the report

The report must contain:

1. Name of the work, purpose.
2. The sequence of actions in the performance of each task.
3. Conclusions on the implementation of laboratory work.

8. Control questions

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. List of references recommended for use on this topic

1. Zhuk, A.P., Zhuk, E.P., Lepeshkin, O.M., Timoshkin, A.I. Zashchita informatsii: Uchebnoe posobie dlya bakalavrov [Text] / A.P. Zhuk et al. – M.: INFRA-M, 2015. – 392 p.

2. Shangin, V.F. Zashchita komp'yuternoy informatsii [Protection of computer information]. Effective methods and means [Text] / V.F. Shangin – Moscow: DMK Press, 2012. – 544 p.

LABORATORY WORK №8

STUDY OF SOFTWARE PRODUCTS.

PGP (Pretty Good Privacy) PROGRAM

Objective: To familiarize with the general principles of building and using software tools for information security, in particular with **the PGP program.**

Mastering software system tools **PGP** designed for:

- encryption of confidential resources and access control to him;
- ensuring the integrity of information resources with the help of electronic digital signature;
- reliable destruction of residual confidential information;
- hiding the presence of confidential information in the computer system using a virtual disk.

• To perform a lab, if you do not have PGP on your computer, you must install it.

• The installation file is attached to the description of the PGPfreeware602i laboratory work. Select only the following components for installation:

- PGP 6.0.2 Program Files;
- PGP 6.0.2 User's Manual;
- Unconfigured PGP 6.0.2 Client Install;
- PGPdisk for Windows.

When asked by the installer about the existence of the keys, answer "No", and when asked about the need to reboot the system, answer "Yes".

Lab Description PGP 6.0 Features

The main tasks of the *PGP* program are to encrypt files and mail messages, certify them with electronic signatures, and completely destroy files on the disk. In addition, *PGP* provides the following features:

- storage of public keys on a remote server;
- the use of three symmetric encryption algorithms and two asymmetric ones;
- four method Launches: *E-mail plugins*, *PGP tray*, *PGP tools*, context menu;
- key sharing;
- setting the level of key validity and trust in the key owner.

PGPkeys

This is a program included in *PGP 6.0* and provides work with keys, Fig. 8.1.

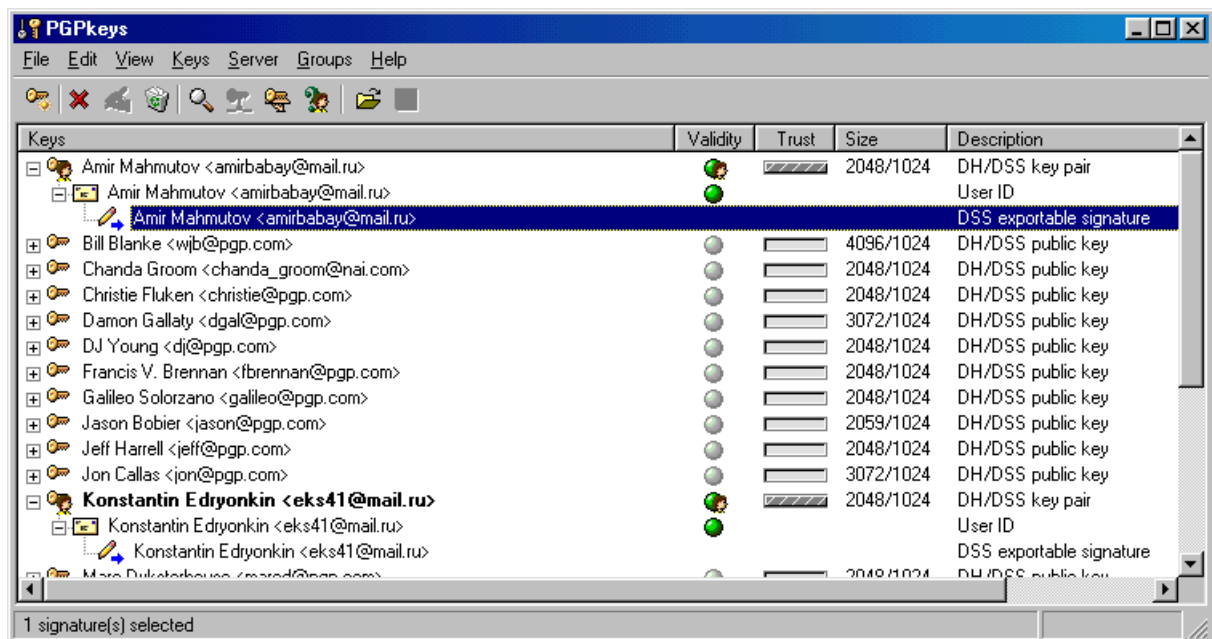


Fig. 8.1 PGPkeys window

The **File** menu contains two commands, one of which is **Exit**, the other is **Send Key Shares**, which allows you to send a local copy of the split key over the network.

In addition to the usual commands, the **Edit** menu contains the **Preferences** command, which is used to specify the program settings, Fig.8.2.

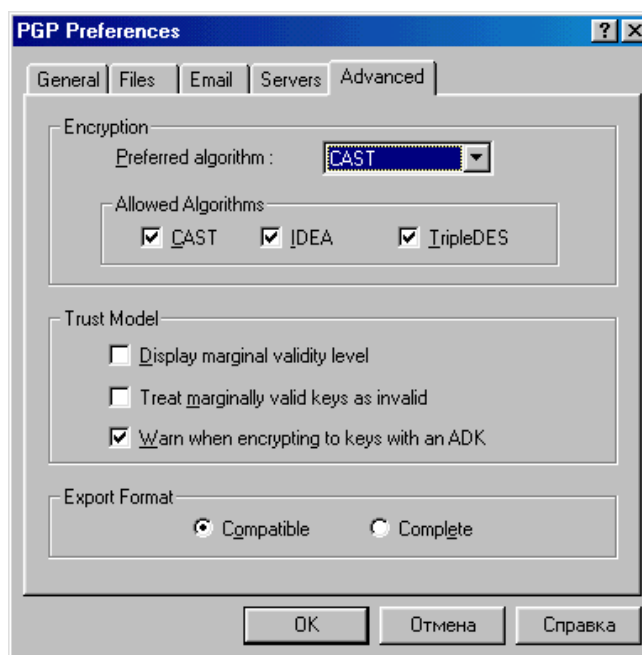


Fig. 8.2. The PGPPreferences window

Using the **View** menu, you can specify the key properties that are displayed on the screen.

Keys menu commands

Sign – allows you to sign the public keys of other users with your private key. In this way, you can show that you trust the owner of the key being used, that is, the key really belongs to the owner. At the same time, you can set modifiers: 1) *Non Exportable* – for a local set of keys; 2) *Exportable* – the certified key is sent to the server; 3) *Meta-introducer Non-Exportable* – you trust not only the owner, but also any key they trust; 4) *Trusted Introducer Exportable* – you trust the owner sign keys, Fig.8.3.

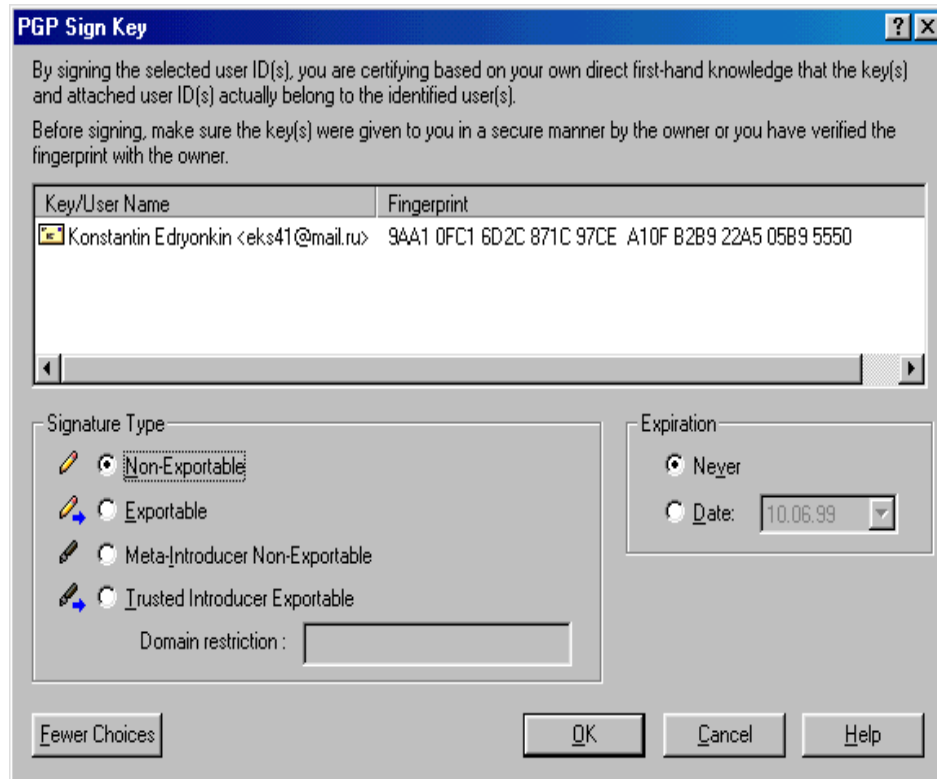


Fig.8.3. PGP SignKey window

Set as Default Key – assign this key as the default key. **Add Name/Photo/Revoker** – allows you to add the name of the owner who has the right to declare your key invalid to the key properties.

Revoke – declare the key invalid.

Reverify Signatures – check the signature (correctness) of the key.

New Key – add a new key.



Fig. 8.4. Key Generation Wizard Window

Share Split – Share the key between several owners.

Import/Export – import/export the key from/to a text file. **Key Properties** – properties (at the key), Fig. 8.4, 8.5.

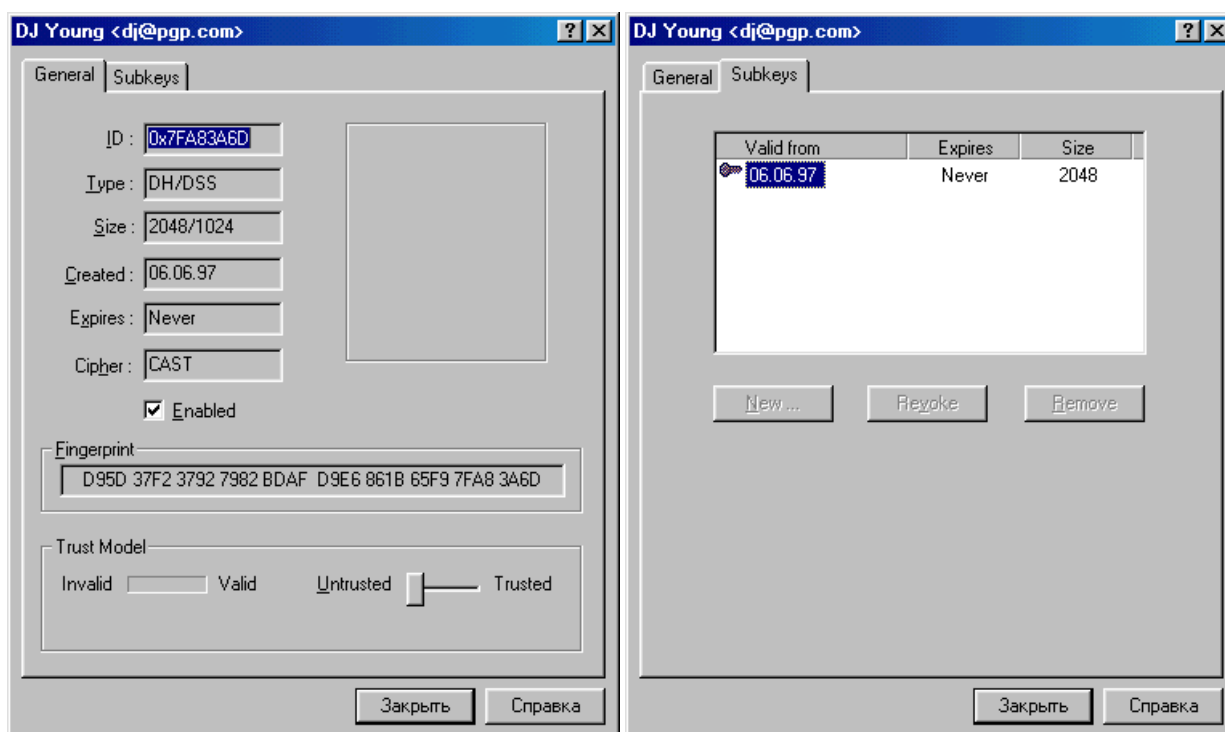


Fig. 8.5. Key properties

The **Server** menu contains commands for working with the remote key server. The Send to command allows you to send keys to the selected server, Fig.8.6.

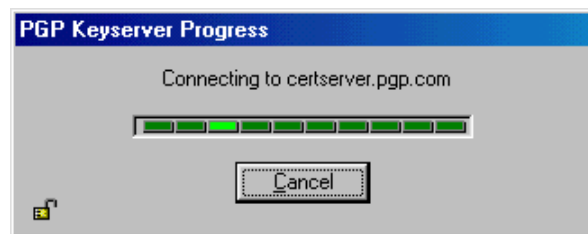


Fig.8.6. Connection to a remote server

Using the Search command, you can try to find the key on the remote server by setting the appropriate parameters, Fig.8.7.

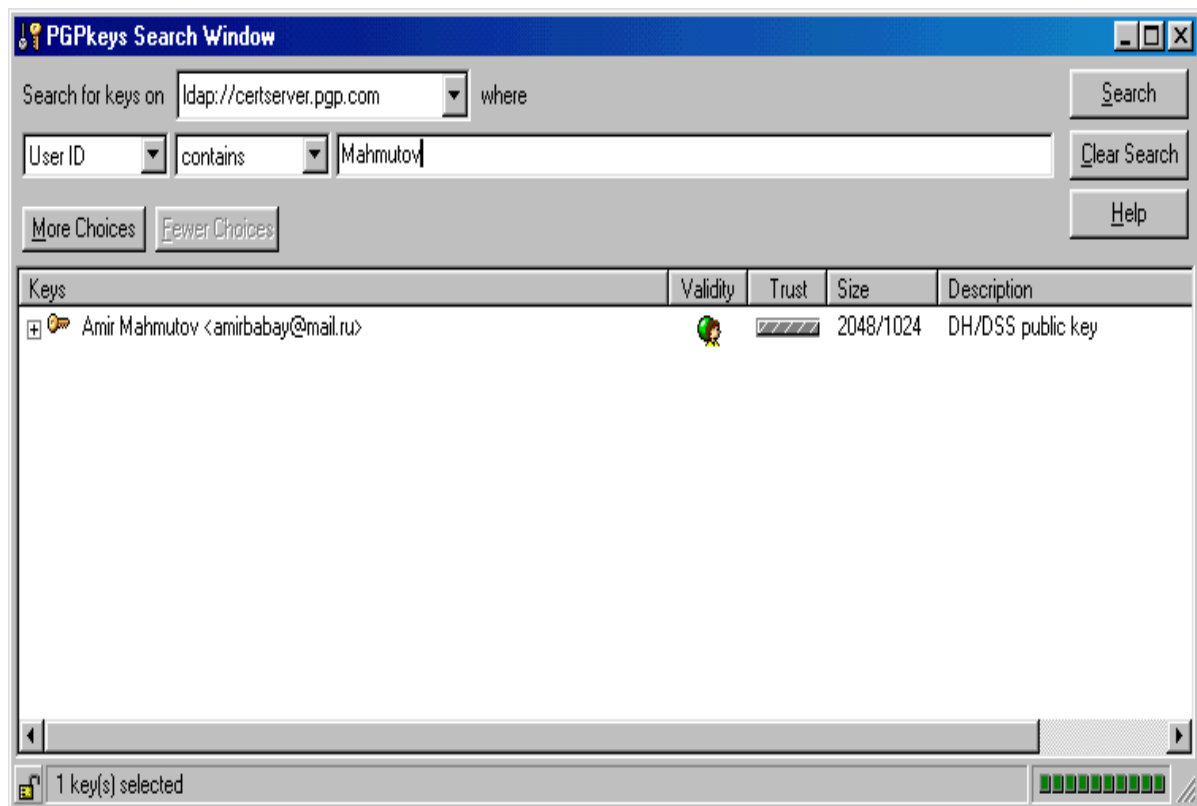


Fig. 8.7. Searching for a key on a remote server

The Update **command** allows you to update the selected key by retrieving information from the server.

The **Groups** menu contains commands for working with groups. Commands **New Group**, **Show Groups**, **Import Groups** allow you to:

respectively, add a group, show groups, import groups from a file. Combining key owners into groups makes it easy to encrypt messages to send to all members of the group.

Finally, the **Help** menu, standard for *Windows*, allows you to get help

PGPtray

It loads when *Windows* starts. To activate the menu, just click on the icon next to the clock, Fig.8.8.

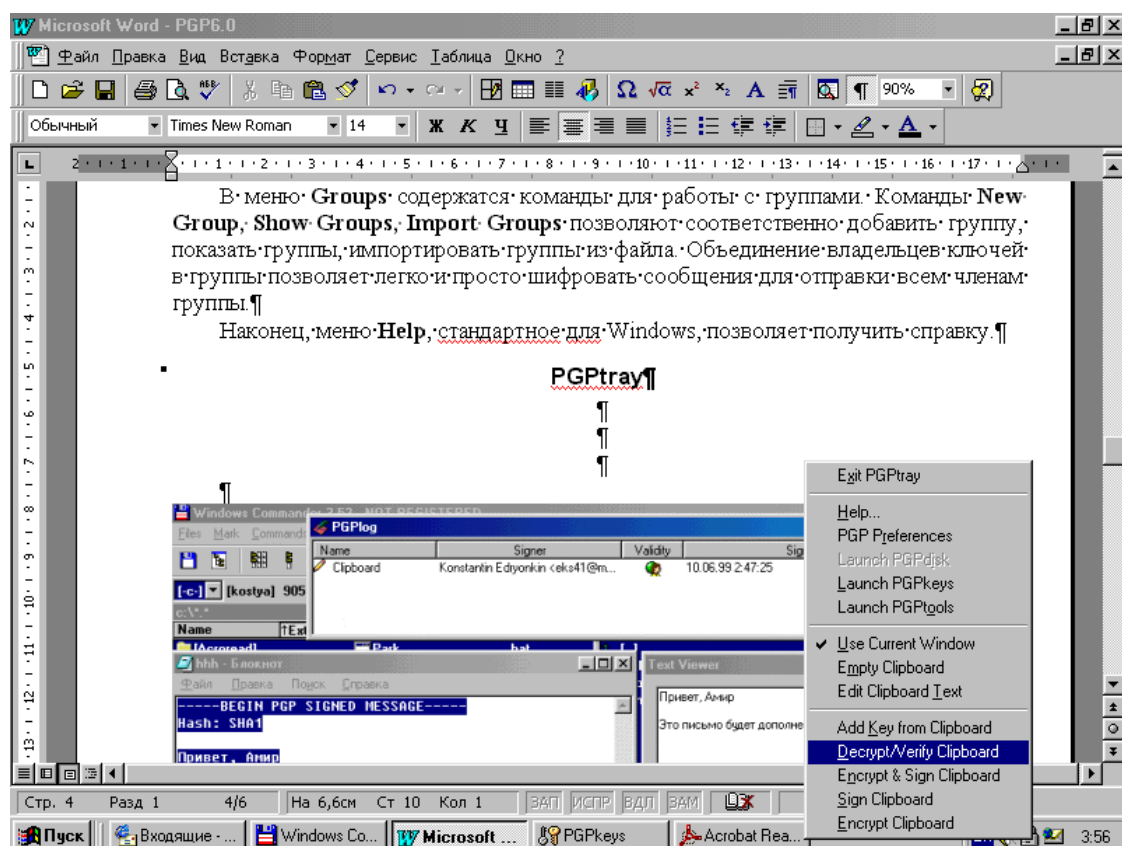


Fig. 8.8. Starting PGP Using the Taskbar Icon

The menu contains commands for exiting, launching other *PGP* components, editing the clipboard, working with the clipboard (add key, decrypt/verify signature, encrypt and sign, just signature and just encryption).

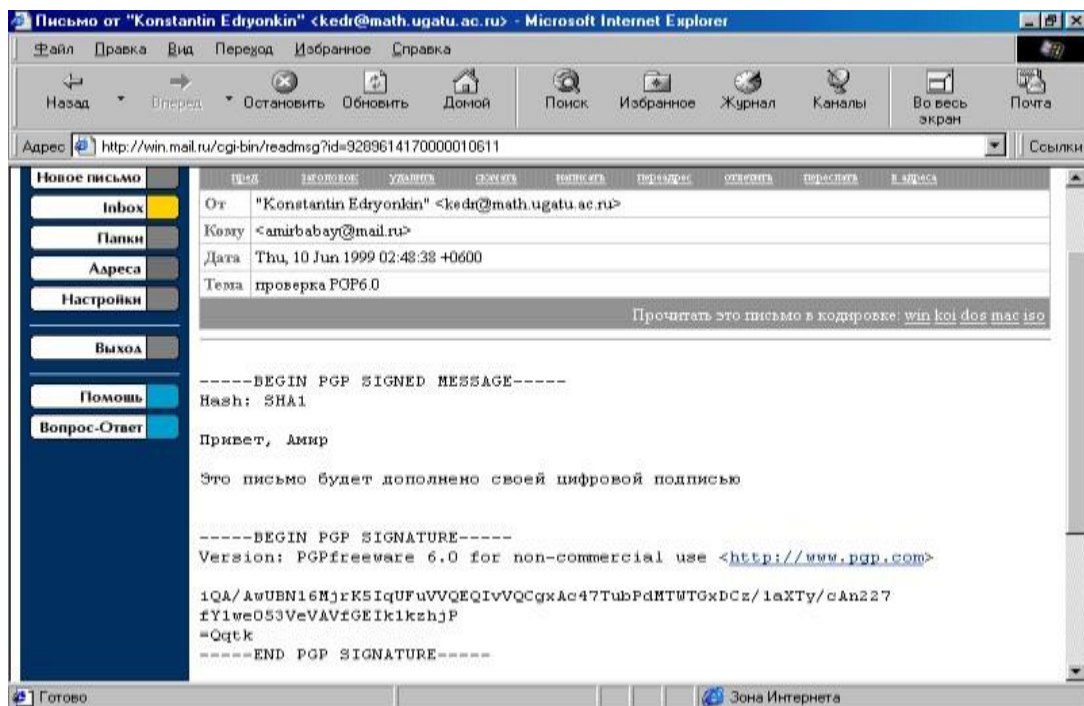


Fig. 8.11. Example of receiving a digitally signed letter

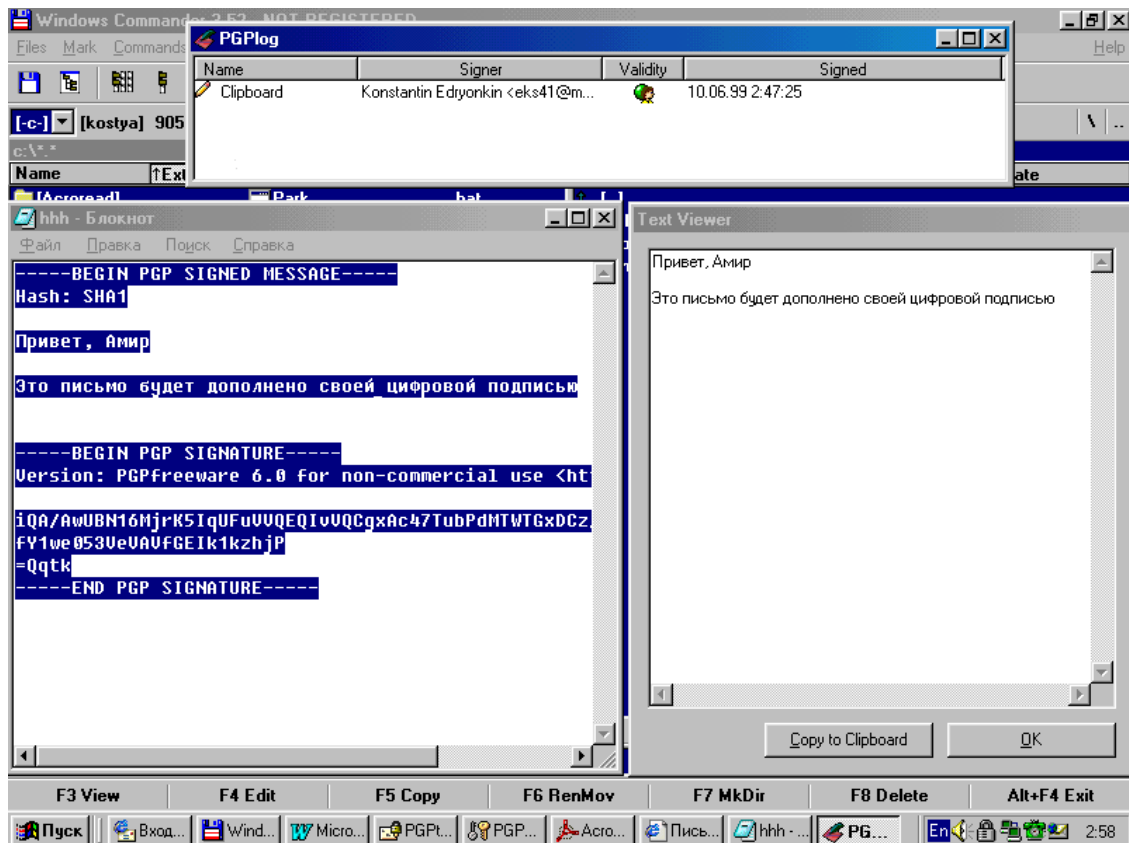


Fig. 8.12. Signature verification results

PGPtools

This component *PGP* is a small panel with buttons that allow you to perform the desired action: open *PGPkeys*, encrypt, sign, Encrypt and Sign At the same time, decrypt/verify the signature, erase the file, erase unused disk space, Fig. 8.13.

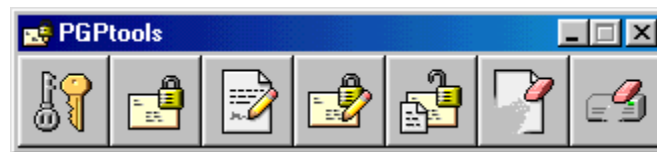


Fig. 8.13. PGPtools panel components

An alternative to *PGPtools* is the commands that are added to the *context menu* for working with files and the Windows File menu when installing *PGP*.

Task

1. See information about the PGP program (see paragraph 1).
2. Launch PGPtools (using the Start menu or the PGPtray icon on the taskbar), review and report on the software included in the *PGP system* (if necessary, use the PGP system help).
3. Create cryptographic keys using PGPkeys. Include information in the lab report about how to generate encryption keys in the *PGP system*, and include copies of the screen forms used in this process in the electronic version of the report, as well as answers to the following questions:
 - how the randomness of the selected cryptographic keys in the PGP system is ensured;
 - how and where the user's private key is stored in the PGP system;
 - how PGP can be able to recover a user's private key if it is accidentally lost.
4. To study (using the example of ordinary text files and image files)

methods of encrypting and decrypting files using the Encrypt and Decrypt functions of PGPtools. Include in the laboratory report *information about the procedure for encrypting and decrypting files in the PGP system*, including in the electronic version of the report copies of the screen forms used in this process and answers to the following questions:

- **what additional encryption parameters can be used, what is their meaning and possible application (required)**
- **Check on Example and results Checks reflected in the report);**
- **How to is generated, How to and where The key is stored symmetric file encryption in the PGP system;**
- **how the encrypted file can be accessed by other users;**
- **changes Lee and How to Size file After his encryption (give specific examples for different types of files).**

5. Explore (using the example of documents in your folder) how to obtain and verify an electronic digital signature under files using the Sign and Verify features of PGPtools. Include in the report *information on how to ensure the authenticity and integrity of electronic documents in the PGP system*, include copies of the screen forms used in this process in the electronic version of the report, and answers to the following questions:

- **what additional parameters for obtaining an electronic digital signature can be used, what is their meaning and possible application (be sure to check on an example and reflect the results of the check in the report);**
- **what is the reaction of the program to the violation of the integrity of the signed document (be sure to check on an example and reflect the results of the check in the report).**

6. To study the methods of simultaneous encryption (decryption) and

Obtain (verify) an electronic digital signature in the PGP system using the Encrypt Sign and Decrypt/Verify functions of PGPtools. Include in the report *information on how to simultaneously ensure the confidentiality, authenticity and integrity of electronic documents in this system*, as well as include in the electronic version of the report copies of the screen forms used in this process.

7. Learn how to securely delete sensitive files using PGPtools' Wipe feature. Include in report *information on the procedure for the destruction of confidential electronic documents in the PGP system* and include copies of the screen forms used in this case in the electronic version of the report.

8. Explore ways to reliably destroy residual information that may contain sensitive information using PGPtools' Freespace Wipe feature. Include information about *the purpose and use of this feature in the program*, copies of the screen forms used in the program, and answers to the following questions:

- **how to reliably destroy residual confidential information in the PGP system;**
- **whether such a method of destruction is completely reliable and, if not, how can the residual information be completely reliably destroyed**

9. Learn how to quickly perform the functions of a PGP system using PGPtray, a shortcut to the right side of the taskbar. Include information about the *purpose and use of the program in the report*, and include copies of the screen forms used in the electronic version of the report.

10. Learn how to manage PGP settings in organizations using PGPadmin (go through all the steps of the wizard dialogue up to the last one, where instead of the "Save" button

click the "Cancel" button). Include in the report *information about the capabilities and procedure for administering the PGP system*, include in the electronic version of the report copies of the screen forms used in this process and answers to the following questions:

- **what encryption and asset management functions PGPAdmin provides to the Programme Administrator;**
- **What PGPadmin features provide to the administrator to manage PGP user cryptographic keys.**
- **what PGPadmin can do to manage user-friendly PGP features and where such information is stored.**

11. Include in the laboratory report the answers to the test questions selected according to the option number indicated by the teacher.

Number Variant	Control questions
1,5,7	How do I choose the length of a cryptographic key in a PGP system?
2,4,6,8	What are the advantages and disadvantages of cryptographic methods of information protection?
11,13,15, 17,19	What computer systems are called secure?
12,14,16	What are the main requirements for the security of computer systems?
20,22,24, 30	To meet what requirements for the security of computer systems can cryptographic protection methods be used?
21,23,25	How reliable, in your opinion, are the methods of cryptographic protection of information used in the PGP program?
3,9,18, 28	What are the main information security features of PGP?

10,26,27, 29	What is the principle behind PGP's Secure Destruction of Residual Privacy feature?
-----------------	--

References

1. Agranovsky A. V., Khadi R. A. Prakticheskaya kriptografiia [Practical cryptography]. Algorithms and their programming (+ CD-ROM). St. Petersburg, SOLON-Press Publ., 2002.
2. Welshenbach M. Cryptography in C and C++ in Action (+CD-ROM). Moscow, Triumph Publ., 2004.
3. Bashly P.N., Babash A.V., Baranova E.K. Informatsionnaya bezopasnost': uchebno-prakticheskoe posobie [Information Security: Educational and Practical Manual]. Moscow: Izd. EAOI Center, 2010.
4. Coutinho S. Introduction to Number Theory. RSA Algorithm. Moscow: Postmarket, 2001.
5. Mao V. Sovremennaya cryptography: teoriya i praktika [Modern cryptography: theory and practice]. Moscow: Izdatel-house Williams, 2005.
6. Moldovyan A. A., Moldovyan N. A., Sovetov B. Y. Cryptography. Textbook for Higher Educational Institutions – Moscow: Lan, 2005.
7. Fomichev V.M. Discrete Mathematics and Cryptology. Course of Lectures. Moscow: Dialog-MEPHI, 2003.

LABORATORY WORK No 9

CORRECTIVE CODES. HAMMING CODES

Objective: Familiarization with the general principles of construction and use of correction codes to control the integrity of information distributed through telecommunication channels.

Note: To perform laboratory work on a computer, you need to install the *Hemming.exe*, which is archived in *Code ХЭММИНГА.rar*.

1. THEORETICAL INFORMATION

1.1. Ways to control the correctness of data transmission

1.1.1. Parity-checked code

Control of the integrity of information during transmission from the source to the receiver can be carried out using correction codes.

The simplest correction code is *a code with a parity check*, which is formed by adding one redundant one to the group of information bits, the value of which is chosen in such a way that the sum of the units in the code combination (that is, the weight of the code combination) is always even.

Example. Consider the parity-checking code generated by adding a check bit to a simple binary code.

	<i>Information</i>	<i>Control</i>
0	000	0
1	001	1
2	010	1
3	011	0
4	100	1
5	101	0
6	110	0
7	111	1

Such a code detects all single errors and group errors of odd multiplicity, since the parity of the number of units in this case will also be violated.

It should be noted that when coding, it is advisable to make the number of units in the code combination odd, and to check for oddness, in this case, any combination, including the one depicting zero, will have at least one unit, which makes it possible to distinguish the complete absence of information from the transfer of zero.

1.1.2. Hamming Codes

N - the digit Hamming code has m - informational Discharges and
 k - control bits. The number of check bits must meet the ratio:

$$k \geq \log_2 (n + 1),$$

From
$$m \leq n - \log_2 (n + 1).$$

The Hamming code is constructed in such a way that the number of check bits calculated according to the above formula is added to the existing information bits of the code combination, which are formed by calculating the parity of the sum of units for certain groups of information bits. When such a code combination is received from the obtained information and check bits by means of similar parity calculations, a correction number is made, which is equal to zero, in the absence of an error, or indicates the number of the erroneous digit.

Let's take a closer look at the coding process.

Suppose the first check digit has an odd sequence number, set it during coding in such a way that the sum of the units of all rarefactions with odd sequence numbers is equal to 0. Such an operation can be written as a ratio:

$$E1 = a1 \oplus a3 \oplus a5 \oplus a7 \dots = 0,$$

where a_1, a_3, a_5, a_7 are binary characters placed in digits numbered 1, 3, 5, 7,...

The appearance of one in the second digit (right) of the correction number means an error in those bits of the code combination, the ordinal numbers of which (2, 3, 6, 7, ...) in the binary image have one in the second digit from the right. Therefore, the second coding operation, which allows you to find the second check bit, looks like this:

$$E_2 = a_1 \oplus a_3 \oplus a_6 \oplus a_7 \dots = 0,$$

Reasoning in a similar way:

$$E_3 = a_4 \oplus a_5 \oplus a_6 \oplus a_7 \dots = 0, E_4 = a_8 \oplus a_9 \oplus a_{10} \oplus a_{11} \dots = 0,$$

After receiving the code combination, together with the formed control bits, the same counting operations are performed as described above, and the resulting number:

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

is considered corrective, and at $E_k E_{k-1} \dots E_2 E_1 = 0$ there are no errors, and if there are errors, those amounts E_i , in the formation of which the erroneous digit participated, are not equal to zero.

The correction number will be equal to the ordinal number of this digit.

The choice of place for check bits in each of the code combinations is determined in such a way that the check bits participate in only one parity calculation operation. This simplifies the coding process, such positions are integer powers of two: 1, 2, 4, 8, 16,...

Example. Let's make a six-digit Hamming code

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

fra	Qi	Simple Binary code	Hamming Code				
			6 2	5 1	4	3	
0		000	0	0	0	0	0
1		001	0	0	0	1	1
2		010	0	1	1	0	0
3		011	0	1	1	1	1
4		100	1	0	1	0	1
5		101	1	0	1	1	0
6		110	1	1	0	0	1
7		111	1	1	0	1	0

Error correction options

Code accepted: 111100 corrected 110100 - error on the correction number in the 4 digit;

111010 corrected 101010 - error on the correction number in the 5 digit;

100000 fixed 000000 - error on the correction number in bit 6

1.1.3. Cyclic codes

Cyclic codes are a type of systematic codes and therefore have all their properties. A characteristic feature of a cyclic code, which determines its name, is that if *the n-digit code combination* $a_0 a_1 a_2 \dots a_{n-1}$ *belongs to a given code, then the combination* $a_n a_0 a_1 a_2 \dots a_{n-1}$ *obtained by cyclic permutation of signs, also belongs to this code.*

The idea of constructing cyclic codes is based on the use of *irreducible polynomials*. An irreducible polynomial is a polynomial that cannot be represented as a product of polynomials of lower degrees, that is, a polynomial that is divisible only by itself or by 1.

In the construction of cyclic codes, irreducible polynomials play the role of so-called constitutive polynomials, the type of which, in fact, determines the main characteristics of the resulting code: redundancy and corrective ability. Table 1.1 shows irreducible polynomials with powers $k = 1, 2, 3, 4$.

Table 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

The basic principles of coding in a cyclic code are as follows. A binary-encoded n -digit number is represented by a polynomial $(n-1)$ of the degree of a variable x , and the coefficients of the polynomial are the binary signs of the corresponding digits. The code combinations in the cyclic code are written, read and transmitted starting from the most significant digit. In accordance with this rule, in the future the numbers themselves and the corresponding polynomials will be written so that the most significant digit is on the right.

0 1 2 3 4 5

Example. The number 110101 (the numbering of digits according to the above rule, carried from left to right from 0 to 5) will be represented by a polynomial of the fifth degree: $1 + x + x^3 + x^5$.

It should be noted that the cyclic permutation of bits in the binary representation of the number corresponds to the multiplication of the polynomial by x , in which x^n is replaced by 1 and passes into the beginning of the polynomial.

Example. Let's multiply the polynomial obtained in the previous example by x . The new polynomial $x + x^2 + x^4 + x^6$, transform it by replacing x^6 with 1. Finally, we get $1 + x + x^2 + x^4$, which corresponds to the number 111010.

The cyclic code of an n -digit number, like any systematic code, consists of m information and k control signs, the latter occupying k of the lowest digits. Since, as we have already pointed out, the sequential transmission of code compounds is carried out starting from the most significant digits, the control signs are transmitted at the end of the code.

As already mentioned, the formation of a code is carried out with the help of the so-called generative polynomial, $P(x)$, degree k , the type of which determines the main properties of the code – redundancy and corrective ability.

A code polynomial, $F(x)$, is a polynomial of degree less than $(m+k)$ if it is evenly divisible into the generating polynomial $P(x)$. After the message is transmitted, decoding consists of dividing the $H(x)$ polynomial corresponding to the received code by $P(x)$. In the absence of errors, $H(x)=F(x)$, and the division is performed without remainder. The presence of a non-zero remainder indicates that information has been distorted during transmission or storage.

To obtain a systematic cyclic code, the following ratio is used:

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

where $G(x)$ is a polynomial representing information symbols
(information polynomial);

$R(x)$ is the remainder of $x^k G(x)$ divided by $P(x)$.

Example. Let us consider the encoding of an 8-digit number 10110111. Suppose the encoding is given a generative polynomial of the 3rd degree $P(x) = 1 + x + x^3$.

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

$$\begin{array}{r} x^{10} + x^9 + x^8 + x^6 + x^5 + x^3 \quad x^{10} + x^8 + x^7 \\ \hline x^9 + x^7 + x^6 + x^5 + x^3 \quad x^9 + x^7 + x^6 \\ \hline x^5 + x^3 \\ x^5 + x^3 + x^2 \quad R(x) = x^2 \end{array} \quad \begin{array}{r} 1 + x + x^3 \\ \hline x^7 + x^6 + x^2 \end{array}$$

Using the ratio for to obtain a systematic cyclic code, we find $F(x)$:

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

Thus, In this way, Finally Code the combination corresponding to $F(x)$ has the form

$$\begin{array}{r} 00010110111 \\ 001 \\ \hline \end{array} \quad \begin{array}{l} \text{контрольные разряды} \longrightarrow \boxed{001} 10110111 \end{array}$$

In practice, the applied coding procedure is even simpler, since we are interested only in the remainder, and the quotient is not used in the final result, it is possible to perform sequential subtraction modulo 2 of the divisor from the divisor and the resulting differences until the difference has a lower degree than the divisor. This difference is the remainder. Such an algorithm can be implemented in hardware using a k-bit shifting register that has inverse connections. Obviously, the cyclic code obtained in this way will be systematic.

However, there is also a second variant of obtaining a cyclic code, when the next code combination is obtained by multiplying the code combination $C(x)$ of a simple n-digit code by the polynomial $P(x)$.

In the second method of cyclic code formation, the information and control symbols in the cyclic code combinations are not separated from each other, which complicates the decoding process. Therefore, this method of encoding is used less often than the first one. Let's consider an example of encoding using the second option, and when performing operations, we will use directly the records of the original code combinations in binary form.

Example. A generative polynomial of the form $P(1,0)=1101$ is given. It is necessary to construct a cyclic code from a simple four-digit code in the second way. As Example for Constructions Use original combination $C(1,0)=0011$. The operation of multiplying this combination by the generating polynomial will be written as follows:

$$\begin{array}{r}
 0011 \\
 1101 \\
 \hline
 0011 \\
 \oplus 0011 \\
 0011 \\
 \hline
 0010111
 \end{array}$$

— это и есть циклический код для 0011.

The remaining code combinations of the non-system cyclic code shown in Table 1.2 are obtained in the same way.

Table 1.2

<i>Simple character code $S(x)$</i>	<i>four- bit code $C(x)$</i>	<i>Cyclic (7,4) – code $C(x)$ $P(x)$, where $P(1,0)=1101$</i>
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

The problem of detecting various types of errors with the help of cyclic code, as already mentioned, comes down to finding the necessary generative polynomial. It is not our task to consider this issue, the basic principles of selecting generative polynomials are set out in the textbook in sufficient detail.

Efficient Information Coding

1.1.4. General Provisions

Let us recall that *encoding* is the presentation of messages in a form convenient for transmission through a given channel, and *decoding* is the restoration of information from the received signal. One of the most important issues of encoding is to increase its efficiency, that is, by eliminating redundant

reduction in the average number of characters required for the letter of the message. Such encoding is called *effective coding*. It follows from the above that effective coding solves the problem of *maximum compression of information*.

Taking into account the statistical properties of the source of information, it is possible to minimize the average number of binary characters required to express one letter of the message, which, in the absence of interference, reduces the time of transmission of the message and its volume.

Effective coding is based on *Shannon's basic theorem for a no-interference channel*⁴, which is as follows.

³ Werner M. Fundamentals of coding. Textbook for universities. – Moscow: Technosphere, 2006. – 288 p.

⁴ Shannon K. Mathematical Theory of Communication // K.Shannon. Works on Information Theory and Cybernetics: Translated from English; edited by R.L. Dobrushin and O.B. Lupanov. – Moscow: IL, 1963. – 512 p.

The essence of Shannon's theorem

Messages composed of letters of an alphabet can be encoded so that the average number of binary characters per letter is arbitrarily close to the entropy of the source of these messages, but not less than this value.

The presence of redundancy in messages makes it possible to raise the question of data compression, that is, the transfer of the same amount of information using the sequences of characters of a shorter length - *lossless compression methods*. For this purpose, special compression algorithms are used to reduce redundancy. The compression effect is estimated by *the compression ratio*

$$K = n/q,$$

where n – the number of minimum necessary symbols to convey the

;

q is the number of characters in the message compressed by this method.

For example, with effective binary coding, n is equal to the entropy of the source information.

Along with compression methods that do not reduce the amount of information in the message, compression methods based on the loss of insignificant information are used – *lossy compression methods*. It should be noted that the use of lossy compression methods is unacceptable for some types of information, for example, text or numerical information.



Fig. 1.1. Information compression methods

The simplest way to compress the numerical information represented in *ASCII* code is to use an abbreviated code with four bits per character instead of eight, since in *ASCII* encoding, the digits, as well as *the dot, comma, and space* characters have zero as the first code character, which can be discarded.

Among the compression algorithms widely used due to its simplicity, the most famous is the *RLE (Run Length Encoding) algorithm*, where instead of transmitting a chain of identical characters, a symbol and a value of the length of the chain are transmitted. This method is effective for transmitting raster images, especially monochrome images, where there are a large number of chains of 1 and 0, but is of little use for transmitting text.

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.

The main area of use of lossy compression methods is graphic, audio and video files, where it is possible, partially sacrificing the quality characteristics of information, to achieve a significant reduction in its volume.

The most common methods of information compression are statistical compression algorithms, which take into account the probability of the appearance of individual symbols in the information flow. First of all, these are the Shannon-Fano and Huffman algorithms.

1.1.5. Shannon-Fano algorithm

Effective Shannon-Fano coding is based on Shannon's fundamental theorem for an interference-free channel.

Shannon-Fano Efficient Coding Algorithm

- The letters of the alphabet of messages are written out in the table in descending order of probabilities.

- They are then divided into two groups so that the sums of probabilities in each group are as similar as possible.
- All the letters of the upper half are assigned 1 as the first character, and all the lower ones are assigned 0.
- Each of the resulting groups is in turn divided into two sub-groups with the same total probabilities, and so on, the process is repeated until there is only one letter left in each sub-group.

Example. Let's consider an alphabet of 7 letters, ascribing to each letter the probability of its appearance in the message p_i (there are special tables of the probability of the appearance of letters of the Russian or Latin alphabets in messages, for example, see Appendix 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

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

It is clear that in conventional binary encoding, without taking into account statistical characteristics, three characters are required to represent each letter. The greatest compression effect is obtained when the probabilities of the letters are integer negative powers of two. The average number of characters per letter in this case is exactly equal to entropy.

Average characters per letter

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

where p_i is the probability of the appearance of the i -th character of the alphabet; n_i is the number of characters in the code combination of the i -th character of the alphabet. For the example considered in the table

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

The difference in values $(L - H)$ is the *redundancy of the code*, and the value of the $(L - H)/L$ – *relative redundancy*.

Redundancy can be interpreted as a measure of uselessly made alternative choices. Since in practical cases individual characters never occur equally often, encoding with a constant length of code words is in most cases redundant. Despite this, such encoding is used quite often, guided by technical considerations.

1.1.6. Huffman's algorithm

Huffman's Efficient Coding Algorithm

The letters of the message alphabet are written out in the main column of the table in descending order of probabilities.

- The last two letters are combined into one auxiliary letter, to which the total probability is attributed.
- The probabilities of the letters that did not participate in the union and the resulting total probability are again arranged in descending order of probabilities, and the last two are combined, and so on, until they get a single auxiliary letter with a probability of 1.
- Next, a binary tree is used to build the code, at the root of which

If a letter is located with probability 1, when branching a branch, the code 1 is more likely to be assigned, and the code 0 is less likely (perhaps the left one is 1, and the right one is 0).

Example. Consider a conventional alphabet of eight letters, each of which is assigned a corresponding probability of its appearance in the message.

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

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

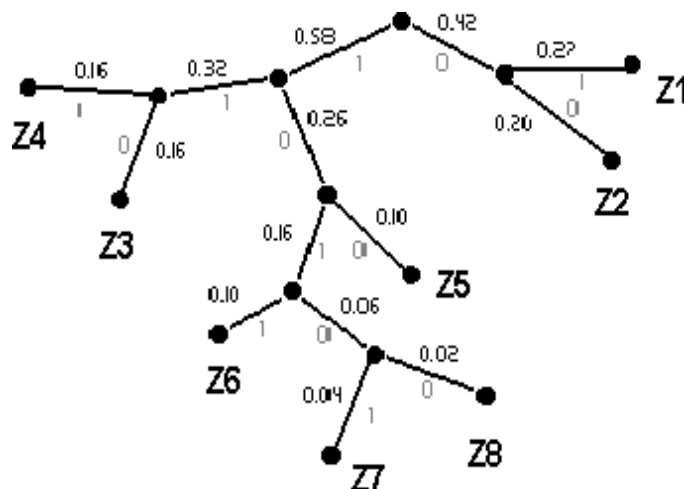
$$H = 2,76;$$

$$L - H = 0,04.$$

For

comparison, in the Shannon code with the same probability distribution, $L - H = 0.08$.

Building a binary tree



Appendix 1.2.2 provides tasks for independent work on information coding using the Shannon-Fano and Huffman algorithms.

Annex 1.2.1

Probabilities of individual letters in the Russian language

Letter	Probability	Letter	Probability
—	0,175	I	0,018
A	0,090	S	0,016
E	0,072	Z	0,016
A	0,062	Ъ,ь	0,014
AND	0,062	B	0,014
T	0,053	G	0,013
T	0,053	H	0,012
S	0,045	Y	0,010
P	0,040	X	0,009
In	0,038	F	0,007
L	0,035	Yu	0,006
K	0,028	Sh	0,006
M	0,026	C	0,004
D	0,025	Sch	0,003
P	0,023	E	0,003
U	0,021	F	0,002

Annex 1.2.2

Tasks for independent work

1. Write a procedure that ensures efficient coding: option a) Shannon code; option b) Huffman code, and calculating the average amount of information per H-sign; redundancy (L-H); and relative redundancy of the resulting code (L-H)/L.

To test the program, use two control options:

Option 1

Letters	Probabilities	Shannon's Code
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 L = 2,84 L – H = 0,08

Option 2

Letters	Probabilities	Shannon's Code
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 = 1 63/64 L - H = 0

2. The source of messages generates signs *A, B, C* with probabilities of 0.7; 0.2; 0.1.

Write a procedure that allows:

- determine the entropy of the source;
- Specify the optimal binary coding for these three characters:

variant a) Shannon code; Variant

b) Huffman code,

and determine the average length of code combinations;

- encode all pairs *AA, AB, ...*;
- Build an optimal binary code for these nine pairs:

variant a) Shannon code; Variant b)

Huffman code.

• increase the blockiness of the code to three-character combinations, and build an optimal binary code:

variant a) Shannon code;

Variant b) Huffman code.

Conclude that code redundancy changes with increasing blocking. How does this affect the efficiency of the code?

Letter	Probability	Code	Sign	Probability	Pair code
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

$$H = 1,1571;$$

$$L = 1,3;$$

$$L_{digit} = 2.33;$$

$$L_{symbol} = L_{Sign} / 2 = 1,165.$$

Conclusion: when encoding in large blocks, it is possible to reduce the redundancy of the code, theoretically, a minimum of redundancy can be achieved when encoding blocks that include an infinite number of letters:

1.2.3. LZW – compression

One of the most widely used lossless compression methods today is the Lempel-Ziv-Welch algorithm .

The compression method was proposed in 1977, and an improved version (*Terry Welch*) was published in 1984, and has since been modified several times. This algorithm is used, in particular, by the standard *UNIX Compress procedure*. The algorithms of the *LZ group* (*LZ77*, *LZ78*, *LZSS*) are the basis of almost all modern software and hardware information compression tools. *PKZIP*, *LHA*, *Stacker*, *SuperStor*, *Dblspace*, and many other archivers use some modification of the *LZ algorithm*.

The idea of compression proposed by Lempel and Ziv is as follows: if a sequence of two previously occurring symbols appears in the text of the message, then this sequence is declared a new symbol, a code is assigned to it, which, under certain conditions, can be much shorter than the original sequence. Later, in the compressed message, the assigned code is written instead of the original sequence.

During decoding, similar actions are repeated and therefore the sequences of symbols for each code become known.

Coding Process

1. Each character of the original alphabet is assigned a certain code (here the code is a serial number, starting with 0).
2. The first character of the message is selected and replaced with its code.
3. The next two characters are selected and replaced with their own codes. At the same time, this two-character combination is assigned its own code (usually it is a serial number equal to the number of codes already used). So, if the alpha includes 8 characters with codes from 000 to 111, then the first two-character combination will receive the code 1000, the next - the code 1001, and so on.
4. The next 2, 3, ... N-character combinations are selected from the source text until a combination that has not yet been encountered is formed. When this combination is assigned a new code, and since the set of the first N-1 characters has already occurred, it has its own code, which is written instead of these N-1 characters. Each act of introducing a new code is called a coding step.
5. The process continues until the source code is exhausted.

Decoding Process

During *decoding*, the code of the first character, and then the second and third characters are replaced by the symbols of the alphabet. In this case, the code of the combination of the second and third characters becomes known. The next position can contain only the codes of already known symbols and their combinations.

Example.

Let the source text be a binary code (the first line of Table 2.3), i.e. the source alphabet $A=\{1,0\}$. The codes of these characters are also 0 and 1, respectively.

The Lempel-Ziv *LZ* code is shown in the second line of Table 2.3. The third line shows the coding steps, after which the transition to the representation of codes A with an increased number of digits r takes place. In general, $r=k$ after step $2k-1$.

In the above example, the *LZ* code turned out to be even longer than the source code, since usually short texts with a small number of repeated symbols do not produce a compression effect. The compression effect is manifested in fairly long texts and is especially noticeable, for example, in graphic files.

Table 1.5

Source text	0.00.000. 01. 11. 111.1111. 110. 0000.00000. 1101. 1110.
<i>LZ</i> code	0.00.100.001.0011.1011.1101. 1010.00110.10010.10001.10110.
r	2 3 4
Codes to be entered	- 10 11 100 101 110 111 1000 1001 1010 1011 1100

An improved algorithm for compression and decompression according to the Lempel-Ziv-Welch method is shown in Fig. 1.2. Fig. 1.3 shows an example of a software implementation of the *LZW algorithm*.

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
```

Fig.1.2. *LZW* compression and decompression algorithm

Введите текст:

///wWRRRTTET

Код

/ #256 W #258 R #260 T T E T

№	Вход новый код	Старый код	Строка выход	Символ	Новый код таблицы
1	/	/	/		
2	256	/	//	В	//
3	W	256	W	W	//W
4	258	W	WW	!	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

Fig.1.3. An example of a software implementation of the algorithm

1. Description of the laboratory work

The program is designed to encode symbols according to the Hamming algorithm. The main window of the program is shown in Fig.9.1.

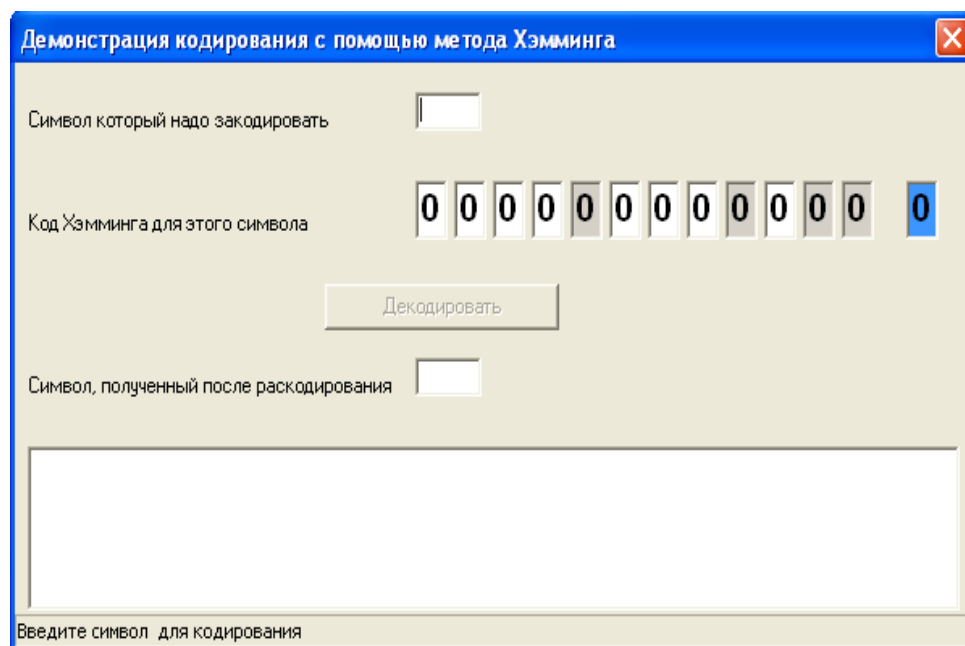


Fig. 9. 1. Main window of the program

Any ASCII characters are encoded using a thirteen-character Hamming code containing 8 information bits, 4 control bits and a common parity digit.

The processes of encoding and decoding are presented in Fig.9.2

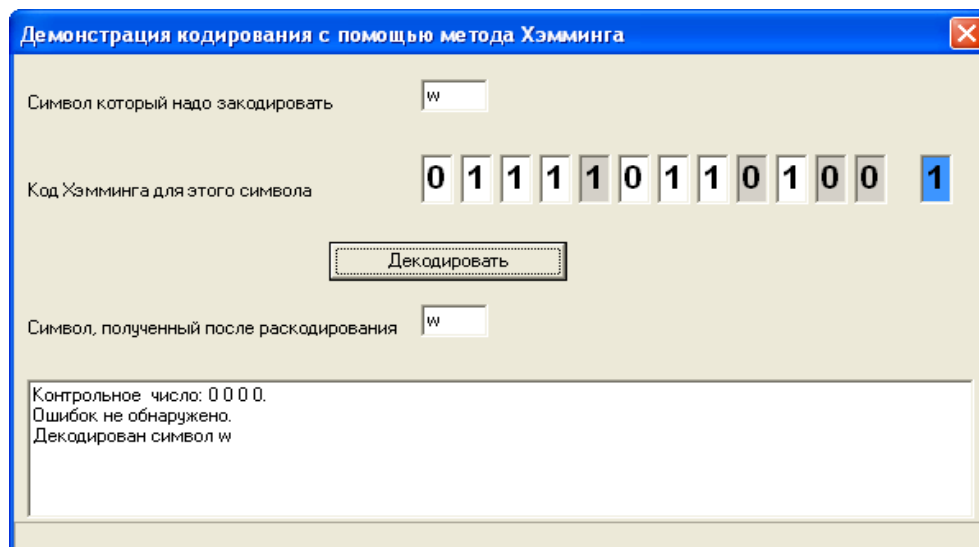


Fig. 9. 2. Encoding and Decoding Process

The check bits are marked in gray, and the common parity bit is marked in blue. Errors can be entered into the resulting code. Single errors can be corrected, double errors can be detected without correction. If the error has been corrected, then the bit in which it was made is indicated. Then the symbol is decoded. The process of correcting a single error is shown in Fig. 9.3.

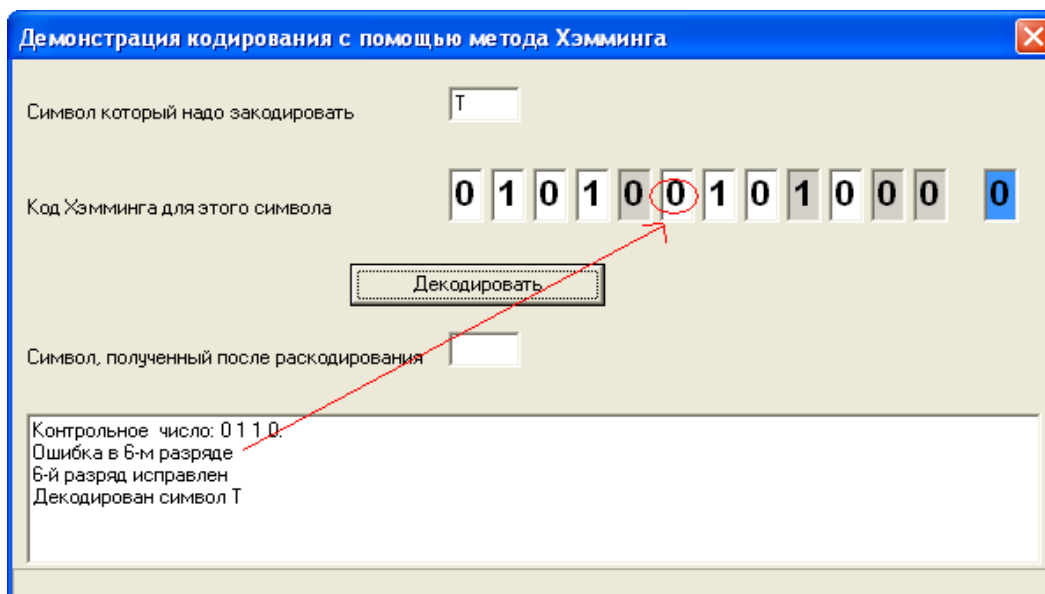


Fig. 9. 3. Single Error Correction Process

If a double or more multiplicity error occurs, a message is displayed that it is impossible to correct the code.

2. Task

1. Use the Hamming code to encode the proposed alphabet (options are shown in Table 9.1).

2. Enter a single error in each row of the table with encoded information, record the decoding result in the code table.

3. Make double errors in the last two lines of the table with encoded information, and record the decoding result in the code table.

4. To analyze the results obtained and formulate authoritative conclusions.

Describe the received Hamming code:

- the number of control and information digits and their numbers;
- code redundancy;
- relative redundancy;
- minimum code distance;
- rate corrective Ability of the received code.

5. Compose a word of at least 5 characters from the proposed alphabet and encode it using the obtained Hamming code.

Calculate the length of the source text (*ASCII* encoding) and the encoded text (Hamming code).

6. To present a brief theoretical material in which to describe the methods of encoding and decoding information using the Hamming code.

7. To present the results of the coding experiments carried out using the Hemming.exe program .

8. Evaluate the results of detecting and correcting single errors and detecting double errors. Draw conclusions about the corrective ability of the

code.

9. Give answers to control questions in the report, in accordance with the number of the option specified by the teacher.

Variants of tasks are presented in Table 9.1 Table

9.1

Rooms tov	Varian-	Original alphabet
1,5,9,13,17		Cyrillic A... M
2,6,10,14,18,22		Cyrillic O.. I
3,7,11,15,19,23,27		Latin alphabet A.. N
4,8,12,16,20,24,28		Latin O.. Z
21,25,29,26,30		Decimal digits 0..9

Variant Number	Control questions
1,5,7, 3,9,18,28	What is a Hamming code and how is it used?
2,4,6,8, 20,22,24, 26,30	How is the Hamming code compiled?
11,13,15, 10,17,19, 27	What errors can be corrected with a Hamming code? What properties of the code determine its corrective ability?
12,14,16 21,23,25, 29	What errors can occur when composing the Hamming code?

LABORATORY WORK No 10

CORRECTION CODES. CYCLIC CODES

The purpose of the work: Familiarization with the general principles of construction and use of correction codes to control the integrity of information distributed through telecommunication channels. Study of the principles of construction of cyclic codes.

Note: To perform laboratory work on a computer, you need to install the *CyclicCode74.exe*, which is in the *Cyclic kod.rar* archive.

1. Description of the laboratory work

1. Study the information about the program for encoding information using cyclic code (*Documentation* folder). Start the encoding program (*folder Executable Module*, file *CyclicCode74.exe*), Fig. 10.1.



Fig. 10.1. Running *CyclicCode74*

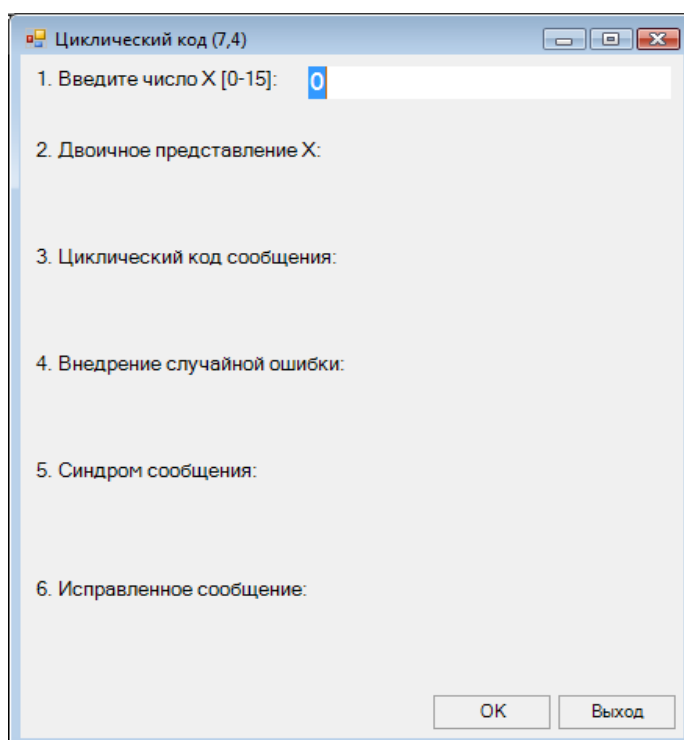


Fig.10. 2. Main program window

2. Encode the proposed variant using a cyclic code (*see task options*). To do this, enter the specified number in the line "1. Enter the number X [0-15]". Save the code table and move it to your personal folder (to do this, press Ctrl+Alt+PrtScSysRq at the same time, then Ctrl+V).

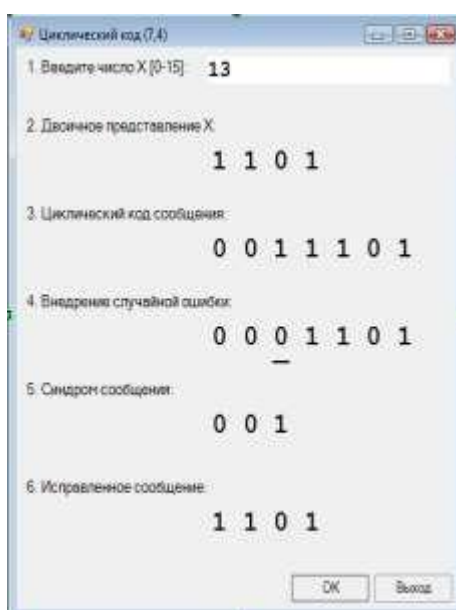


Fig. 10. 3. Example of filling in the fields of the main window of the program

3. To analyze the results obtained and formulate authoritative conclusions.

2. Task

1. Obtain a systematic cyclic code using the generative polynomials given in Table 10.1, according to the option indicated by the teacher:

Table 10.1 Resource requirements by component

<i>Variant</i>	<i>Generating Degree 3 Polynomial</i>
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

Example: Consider encoding an eight-digit number 00011111. Suppose the encoding is given a generative polynomial of degree 3

$$P(1,0) = 1101, P(x) = 1 + x + x^3$$

We divide $x^3 G(x)$ by $P(x)$, where the information polynomial $G(x) = x^3 + x^4 + x^5 + x^6 +$

$$x^7 \quad x^3 G(x) = x^6 + x^7 + x^8 + x^9 +$$

$$x^{10} \quad x^{10} + x^9 + x^8 + x^7 + x^6 \quad x^{10} +$$

$$x^8 + x^7$$

$$\underline{x^9 + x^6}$$

$$\underline{x^9 + x^7 + x^6}$$

$$x^7$$

$$\underline{x^7 + x^5 + x^4}$$

$$x^5 + x^4$$

$$\underline{x^5 + x^3 + x^2}$$

$$\underline{x^4 + x^3 + x^2} \quad x^4 + x^2 + x$$

$$x^3 + x$$

$$\begin{array}{r} x^3 + x + 1 \\ \hline x^7 + x^6 + x^4 + x^2 + x + 1 \end{array}$$

$$\frac{x^3 + x + 1}{R(x) = 1}$$

Find the value of the code polynomial $F(x)$:

$$F(x) = (x^6 + x^7 + x^8 + x^9 + x^{10}) \square 1$$

Thus, the final code combination

$F(x)$ has the form of:

Check Digits

	0	1	2	3	4	5	6	7	8	9	10
							1	0	0	0	0
							1	1	1	1	1
							1	0	0		

10000011111

2. Present in the report a brief theoretical material that describes the method of encoding and decoding information using a cyclic code.
3. To present the results of the coding experiments carried out using the CyclicCode74.exe program .
4. Evaluate the results of detecting and correcting single errors. Draw conclusions about the corrective ability of the code under study.
5. Give answers to control questions in the report, in accordance with the number of the option specified by the teacher.

Variants of tasks are given in Table 10.2.

Table 10.2 Resource requirements by component

<i>Variant</i>	<i>Number</i>
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

Number variant	Control questions
1,5,7, 3,9,18,28	Specify Differences concepts: <i>Coding information, information encryption.</i>
2,4,6,8, 20,22,24, 26,30	Define the following concepts: <i>uniform codes, non-uniform codes, prefix coding, separable coding, system codes, non-systematic codes.</i>
11,13,15, 10,17,19, 27	What characteristics of the code determine its corrective ability?
12,14,16 21,23,25, 29	Compare Hamming codes and cyclic codes in terms of the effectiveness of information integrity control.

LABORATORY WORK № 11

SHANNON AND HUFFMAN COMPRESSION METHODS

Objective: To familiarize with the statistical principles of information compression using the methods of Shannon-Fano and Huffman.

Note. To perform laboratory work on a computer, you need to install the *Shannon-Huffman.exe* file, which is located in the *Shannon and Хаффмена.rar Compression Method*.

1. Description of the laboratory work

The work is performed on a personal computer using the *Shannon-Huffman.exe* program. The program is designed to demonstrate methods of information compression using the Shannon-Fano and Huffman algorithms (Fig. 11.1).

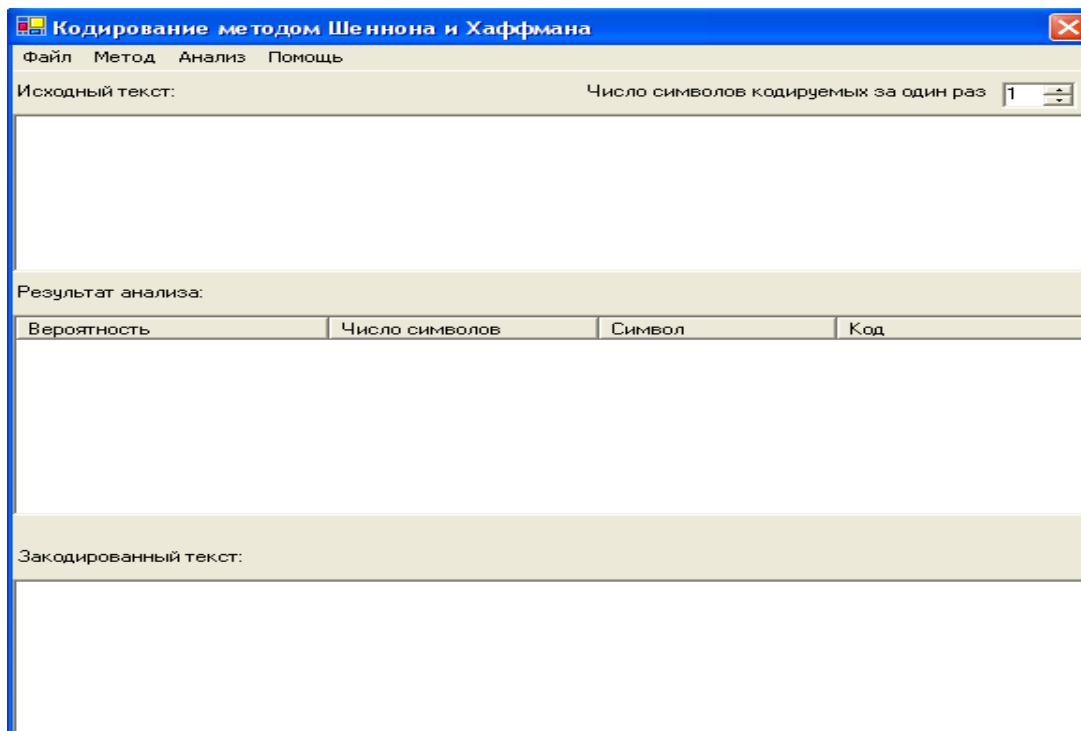


Fig. 11. 1. Main window of the program

To work with the program, the user selects the required compression method, Fig. 11.2.

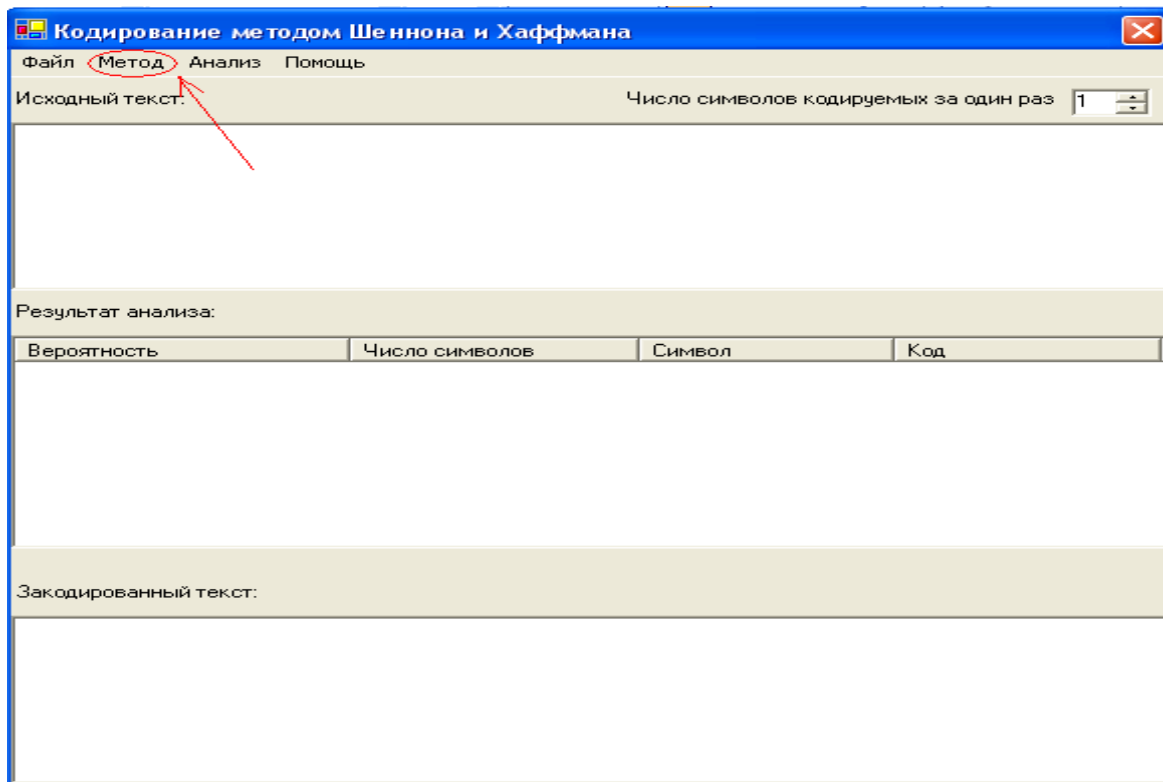
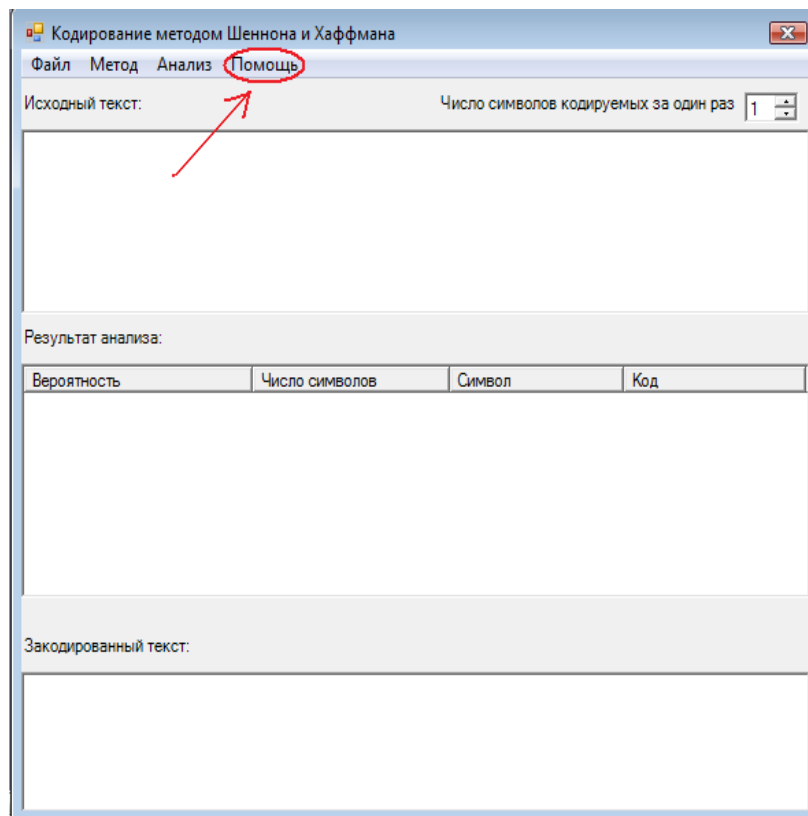


Fig. 11. 2. Choosing a compression method

Next, a message of arbitrary length is written in the *Source text* window (or a message is loaded from a pre-prepared file in *.txt format*). After that, you need to specify the number of symbols to be encoded at a time.

To calculate the probability of letters appearing in the entered message P , to determine the entropy of the source of messages H , to determine the average number of symbols when encoding one letter of the message L , you need to select the Analysis overlay.



To determine the coding efficiency, code redundancy ($L-H$) is calculated.

To exit this program, just select the *File* and then *Exit*.

Example 1.

Original message *abbwvvggggdddddeeeeeee*, consisting of six-letter characters $A = \{a, b, c, d, e, e\}$, encode using the Huffman method. Determine the probability of the symbol appearing in the original message

$$P = n/N,$$

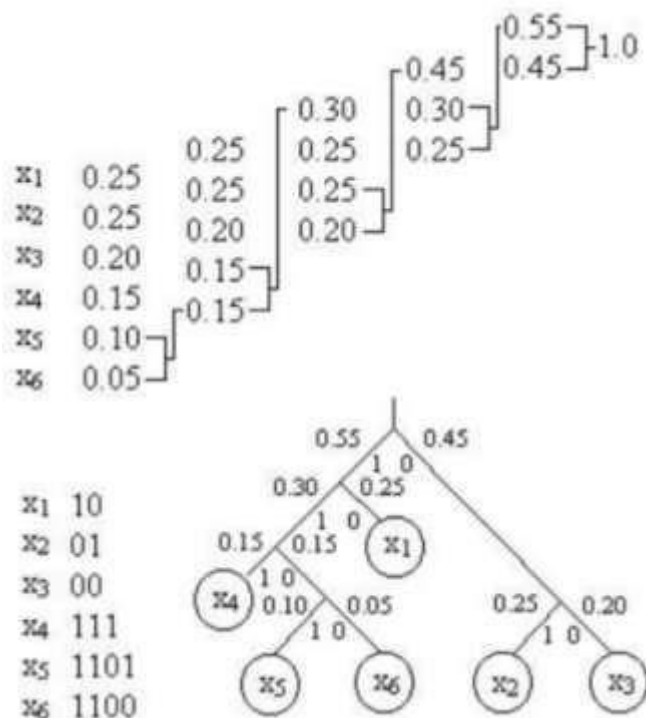
where n is the number of character repeats in the message, N is the length of the message.

Write down all the symbols of the alphabet in the column in descending order of the probability of their appearance in the text (Table 11.1).

Table 11.1 Resource requirements by component

Symbol	Number repetitions In this case, The Communist Party	Probability
<i>d</i>	5	0,25
<i>e</i>	5	0,25
<i>g</i>	4	0,20
<i>in</i>	3	0,15
<i>b</i>	2	0,10
<i>a</i>	1	0,05

By sequentially combining the two symbols with the lowest probabilities of appearing into a new compound symbol, the probability of which is assumed to be equal to the sum of the probabilities of its constituent symbols, we will eventually construct a tree, each node of which has the summative probability of all nodes below it. We trace the path to each leaf of the tree, marking the direction to each node (for example, to the right - 0, to the left - 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 $(L-H) = 0.025$.

Compare the data obtained with the results of the program (Fig. 11.4)

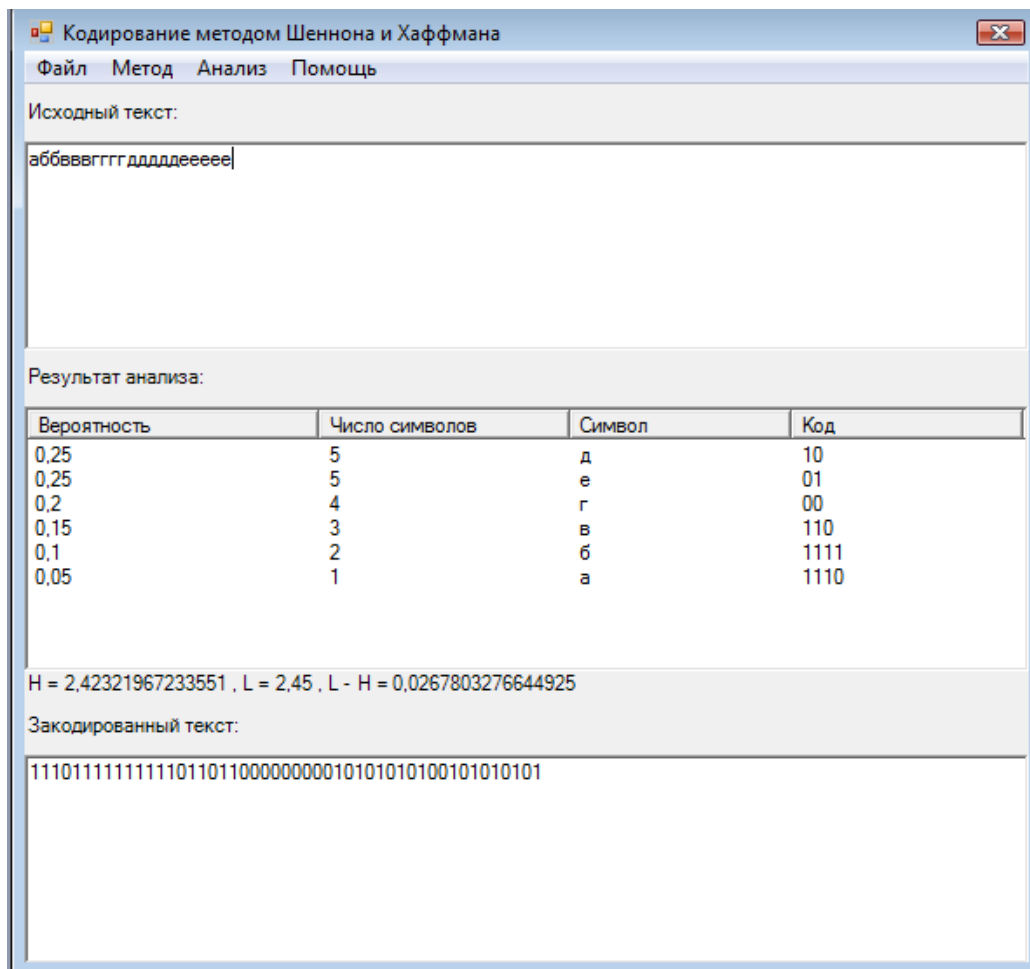


Fig. 11. 4. Coding example

Example 2.

Encode the original message (see *Example 1*), but using the Shannon method. Write down all the symbols in the column in descending order of the probability of their appearance in the text;

Divide the set of symbols sequentially into two subsets so that the sum of the probabilities of the appearance of the symbols of one subset is approximately equal to the sum of the probabilities of the appearance of the symbols of the other. For the lower subset, we assign "0" to each symbol, and "1" for the upper subset. Further splits are repeated until all the subsets consist of one element (Table 11.2).

Table 11.2 Resource requirements by component

Sim- vol	Number of repetitions in this message	Probability	Shannon's Code		
<i>d</i>	5	0,25	1	1	
<i>e</i>	5	0,25	1	0	
<i>g</i>	4	0,20	0	1	1
<i>in</i>	3	0,15	0	1	0
<i>b</i>	2	0,10	0	0	1
<i>a</i>	1	0,05	0	0	0

The average length of the received code will be

Average number of characters per letter of the message:

$$L = \sum_{i=1}^n n(i) * P(i) = 0.25 * 2 + 0.25 * 2 + 0.2 * 3 + 0.15 * 3 + 0.1 * 3 + 0.25 * 3 = 2.5$$

Entropy:

$$H = - \sum_{i=1}^n P(i) \log \frac{1}{P(i)} = 2.425$$

Compare Obtained Data c Results Jobs programs
(fig.11.5)

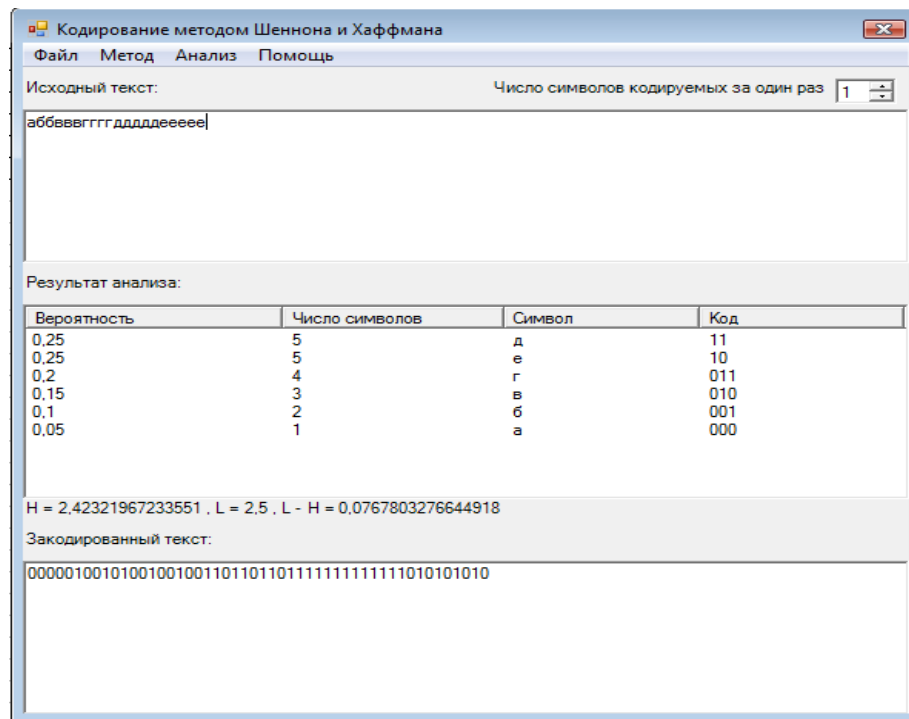


Fig. 11. 5. Coding example

2. Task

Familiarize yourself with information about the Shannon-Huffman.exe program, and consider examples of effective information coding.

1. Run a Shannon-Huffman.exe program designed to demonstrate information compression techniques using the Shannon-Fano and Huffman algorithms.

2. Complete the proposed tasks:

1). Number of characters of the alphabet $k = m$ (m is the number of the variant).

Compose such an initial message that

a) the characters of the alphabet occur in the message with equal probabilities;

(b) The characters of the alphabet occurred in the communication with different probabilities.

2). Enter any coherent text in Russian.

It can be a proverb, a poem or an arbitrary text. Using the results of the program, it is necessary to analyze the alphabet of the entered message: count the number of characters of the alphabet, the value of entropy H , the average number of characters per sign L in integer coding.

3. Save screen forms in the report that demonstrate the process of information compression.

4. The report on the laboratory work should contain the results of the analysis of the Shannon-Huffman.exe program during coding; the algorithm for building the Huffman and Shannon code using knowledge on the topic; the results of comparison of different coding methods; conclusions about the effectiveness of the compression methods used.

5. Give answers to control questions in the report, in accordance with the number of the option specified by the teacher.

Include completed tasks 1, 2,
3.

Variants of tasks

Task 1.

The message consists of a sequence of two letters, A and B, the probabilities of each of which do not depend on which one was transmitted earlier, and are equal to 0.8 and 0.2, respectively. Encode according to the Shannon method: a) individual letters; b) blocks consisting of two-letter combinations; c) blocks consisting of three-letter combinations. Compare the resulting codes according to their effectiveness.

Task 2.

Compose a text that would correspond to the data given in Table 11.3.

Using the Shannon-Huffman.exe program, encode the text of the Huffman method.

Table 11.3 Resource requirements by component

Variant's Room	Probability	Symbol	Number volov Sim
1,5,7, 3,9,12,14,18,28	0,33333333	about	2
	0,16666667	g	1
	0,16666667	p	1
	0,16666667	d	1
	0,16666667	a	1
2,4,6,8,16,21 20,22,24, 30	0,25	e	2
	0,25	t	2
	0,125	about	1
	0,125	P	1
	0,125	p	1
	0,125	a	1
11,13,15, 10,17,19,23,25	0,25	p	2
	0,25	a	2
	0,25	c	2
	0,125	t	1
	0,125	e	1

Task 3.

For variants A), B), C) (see Figure 11.6 and Table 4), make a Houghman code. Calculate the average number of characters per character; redundancy ($L - H$); and relative redundancy of the resulting code $(L - H) / L$. Compare the obtained values with L , H , $(L-H)$ for the Shannon code, draw conclusions.

Δ)

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

В)

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

С)

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

Fig. 11. 6. Variants of tasks

Table 11.4 Resource requirements by component

Variant Number	Task
1,5,7, 3,9, 12,14,18, 28	Task 3. Option A).

2,4,6,8, 20,22,23, 24,25 30	<i>Task 3. Option B).</i>
11,13,15, 16 21,10,17, 19	<i>Task 3. Option C).</i>

Number variant	Control questions
1,5,7, 3,9,18,28	Which codes allow for unambiguous decoding Even without the use of a separating character? Give examples of such codes.
2,4,6,8, 20,22,24, 26,30	What are the conditions for building optimal codes?
11,13,15, 10,17,19, 27	What is the purpose of effective codes, and which ones do you know?
12,14,16 21,23,25, 29	List the main methods of lossless information compression.

MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN
FEDERATION

FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION OF HIGHER
EDUCATION
"NORTH CAUCASUS FEDERAL UNIVERSITY"

SUGGESTIONS FOR TEACHING

on the organization and conduct of independent work on the discipline

"Principles and Techniques of Information Security"

for students of the field of study

11.03.02 Infocommunication Technologies and Communication Systems
"Intelligent Data Processing in Infocommunication Systems and Networks"

Stavropol 2026

Methodical instructions for the organization of students' independent work are compiled in accordance with the requirements of the program of the discipline "Analysis of random signals" for students of the direction of training 11.03.02 Infocommunication Technologies and Communication Systems. The textbook is devoted to the planned educational work of students, carried out in extracurricular (classroom) time on the assignment and under the methodological guidance of the teacher, but without his direct participation (with the partial direct participation of the teacher).

CONTENTS

1. Independent work as the most important form of the educational process	4
2. Goals and main objectives of the ACS	5
3. Types of independent work	6
4. Organization of SRS	7
5. General recommendations for the organization of independent work	8
6. Independent work of a student is a necessary link in the formation of Researcher.....	12
7. Methodical recommendations for students on individual forms independent work.....	13
8. Independent work of students in the conditions of point-rating Training systems	20
9. Educational, methodological and information support of the discipline.....	23

1. Independent work as the most important form of the educational process

Independent work is the planned educational, research, research work of students, carried out in extracurricular (classroom) time on the assignment and under the methodological guidance of the teacher, but without his direct participation (with the partial direct participation of the teacher, leaving the leading role behind the work of students).

Independent work of students at the university is an important type of educational and scientific activity of the student. Independent work of students plays a significant role in the rating technology of education. Education at the university includes two parts that are almost the same in volume and mutual influence - the process of learning and the process of self-study. Therefore, the SRS should become an effective and purposeful work of the student.

The concept of modernization of Russian education defines the main tasks of vocational education - "training a qualified worker of the appropriate level and profile, competitive in the labor market, competent, responsible, fluent in his profession and oriented in related fields of activity, capable of effective work in the specialty at the level of world standards, ready for constant professional growth, social and professional mobility."

The solution of these problems is impossible without increasing the role of students' independent work on educational material, increasing the responsibility of teachers for the development of independent work skills, for stimulating the professional growth of students, fostering creative activity and initiative.

Society imposes a fairly wide range of requirements on a modern specialist, among which it is important that graduates have certain abilities and ability to independently obtain knowledge from various sources, systematize the information received, and assess a specific financial situation. The formation of such a skill occurs during the entire period of study through the participation of students in practical classes, the performance of control tasks and tests, the writing of term papers and graduation qualification papers. At the same time, the independent work of students plays a decisive role in the entire educational process.

The forms of independent work of students are diverse. The discipline "Teletraffic of Multiservice Networks" provides the following:

Use of the material of the educational and methodological complex Disciplines

At the first stage, it is necessary to familiarize yourself with the mandatory minimum content of the basic educational program in the discipline being studied. Next, it is necessary to study the work program of the discipline, which considers the content of the topics of the lecture course, the relationship of lecture topics from laboratory classes, topics and types of independent work. For each type of independent work, certain forms of reporting are provided.

Work with literature

For the successful development of the discipline, it is necessary to independently study the presented topics in detail according to the recommended sources of information:

No p/n	Types of independent work	Recommended sources of information (No source)			
		Main	Additional Telnaya	Methodical	Internet Resources
1	Independent study of literature	1, 2	1, 2	1	1, 2, 3

Independent work introduces students to scientific creativity, the search for and solution of urgent modern problems.

2. Goals and main objectives of the ACS

The leading goal of the organization and implementation of SRS should coincide with the goal of teaching a student - the preparation of a bachelor with higher education. In the organization of SRS, an important and necessary condition is the formation of the ability to work independently to acquire knowledge, skills and the ability to organize educational and scientific activities.

The purpose of students' independent work is to master fundamental knowledge, professional skills and abilities in the profile, experience in creative and research activities. Independent work of students contributes to the development of independence, responsibility and organization, a creative approach to solving problems of the educational and professional level.

The objectives of the SRS are:

- systematization and consolidation of the theoretical knowledge and practical skills of students;

- deepening and expanding theoretical knowledge;
- formation of skills to use regulatory, legal, reference documentation and special literature;
- development of cognitive abilities and activity of students: creative initiative, independence, responsibility and organization;
- formation of independent thinking, abilities for self-development, self-improvement and self-realization;
- development of research skills;
- the use of material collected and obtained in the course of independent studies at seminars, practical and laboratory classes, when writing term papers and final qualification papers, for effective preparation for final tests and exams.

3. Types of independent work

In the educational process of a higher professional educational institution, there are two types of independent work

– classroom, under the guidance of a teacher, and extracurricular. The close relationship between these types of work provides for the differentiation and effectiveness of the results of its implementation and depends on the organization, content, logic of the educational process (interdisciplinary connections, promising knowledge, etc.):

Classroom independent work on the discipline is carried out in classes under the direct guidance of the teacher and on his assignment.

Extracurricular independent work is performed by the student on the instructions of the teacher, but without his direct participation.

The main types of independent work of students without the participation of a teacher in this discipline are:

- formation and assimilation of the content of lecture notes on the basis of educational literature recommended by the lecturer, including information educational resources (electronic textbooks, electronic libraries, etc.);
- preparation for laboratory work, their design.

The main types of independent work of students with the participation of teachers are:

- ongoing consultations;
- acceptance and defense of laboratory work (during the L/R).

4. Organization of SRS

The methodology for organizing students' independent work depends on the structure, nature and features of the discipline "Teletraffic of Multiservice Networks", the number of hours for its study, the type of tasks for students' independent work, the individual qualities of students and the conditions of educational activity.

The process of organizing students' independent work includes the following stages:

- preparatory (definition of goals, drawing up a program, preparation of methodological support, preparation of equipment);
- the main one (implementation of the program, the use of methods of information search, assimilation, processing, application, transfer of knowledge, recording of results, self-organization of the work process);
- final (assessment of the significance and analysis of the results, their systematization, assessment of the effectiveness of the program and methods of work, conclusions about the directions of labor optimization).

Students' activities for the formation and development of skills of educational independent work.

In the process of independent work, the student acquires the skills of self-organization, self-control, self-management, self-reflection and becomes an active independent subject of educational activity.

Performing independent work under the supervision of a teacher, the student must:

master the minimum content submitted for independent work of students and proposed by the teacher in accordance with the educational standards of higher education;

plan independent work in accordance with the schedule of independent work proposed by the teacher;

– the student must carry out independent work in the organizational forms provided for by the curriculum and the teacher's work program;

perform independent work and report on its results in accordance with the schedule for submitting results, types and terms of reporting on students' independent work.

A student can:

in addition to the one proposed by the teacher (with justification and agreement with him) and the minimum mandatory content determined by the program for this discipline:

independently determine the level (depth) of the study of the content of the material;

Suggest topics and questions for self-study;

– within the framework of the general schedule for the performance of independent work, offer a reasonable individual schedule for the implementation and reporting on the results of independent work;

– offer their own options for organizational forms of independent work;

use teaching aids, manuals, developments in addition to the list proposed by the teacher for independent work;

use not only control, but also self-control of the results of independent work in accordance with the methods of self-control proposed by the teacher or chosen independently.

Independent work of students should have an important impact on the formation of the personality of the future bachelor, it is planned by the student independently. Each student independently determines the mode of his work and the degree of work spent on mastering the educational content in each discipline. He performs extracurricular work according to a personal individual plan, depending on his preparation, time and other conditions.

5. General recommendations for the organization of independent work

The main form of independent work of the student is the study of lecture notes, their supplementation, recommended literature, active participation in laboratory classes. But for successful educational activity, its intensification, it is necessary to take into account the following subjective factors:

1. Knowledge of school program material, the presence of a solid system of gaps necessary for mastering the main university courses. This is especially important for mathematical disciplines. It is necessary to distinguish gaps in knowledge that make it difficult to assimilate new material from small abilities. By spending effort to overcome these gaps, the student will ensure normal academic performance and believe in his abilities.

2. Availability of skills, skills of mental work:

a) the ability to take notes at lectures and when working with a book;

b) mastery of logical operations: comparison, analysis, synthesis, generalization, definition of concepts, rules of systematization and classification.

3. The specifics of cognitive mental processes: attention, memory, speech, observation, intelligence and thinking. Poor development of each of them becomes a serious obstacle in learning.

4. Good working capacity, which is provided by normal physical condition. After all, serious learning is a large, versatile and diverse work. The result of learning is evaluated not by the amount of information communicated, but by the quality of its assimilation, the ability to use it and the development of one's ability for further independent education.

5. Correspondence of the chosen activity, profession to individual abilities. It is necessary to develop the ability to self-regulate one's emotional state and eliminate circumstances that disturb the business mood, interfere with the planned work.

6. Mastering the optimal style of work that ensures success in activities. Alternating work and pauses in work, rest periods, individually justified sleep duration, preference for evening or morning classes, stress resistance at exams and features of preparation for them,

7. The level of requirements for oneself, determined by the established self-esteem.

An adequate assessment of knowledge, advantages, disadvantages is an important component of a person's self-organization, without it it is impossible to successfully manage one's behavior and activity.

One of the main features of education in higher school is that constant external control is replaced by self-control, the active role in learning belongs not so much to the teacher as to the student.

Knowing the basic methods of scientific organization of mental work, it is possible to achieve the best results with the least expenditure of time, money and labor efforts.

The effectiveness of assimilation of incoming information depends on the working capacity of a person at a particular moment of his activity.

Working capacity is the ability of a person to work with a high degree of intensity for a certain time. There are internal and external factors of working capacity.

Internal factors of working capacity include intellectual characteristics, will, and state of health.

External:

- organization of the workplace, work and rest schedule;
- Level Organizations labor - Trait Get it Reference and use information;
- the amount of mental load.

Outstanding Russian Physiologist N. E. Vvedensky identified the following conditions for the productivity of

mental activity:

- every work must be entered into gradually;
- dimensionality and rhythm of work. Different people have a more or less different pace of work;
- the usual sequence and systematicity of activity;
- correct alternation of work and rest.

Rest does not imply obligatory complete inaction on the part of a person, it can be achieved by a simple change of business. During the day, the working capacity changes. The most fruitful is the *morning time (from 8 to 14 hours)*, and the maximum working capacity falls on the period from 10 to 13 hours, then *the afternoon (from 16 to 19 hours)* and *the evening (from 20 to 24 hours)*. It is better to study the material that is very difficult to understand at the beginning of each period of time (preferably in the morning) after a good rest. After 1-1.5 hours, breaks of 10 - 15 minutes are needed, after 3 - 4 hours of work, the rest should be long - about an hour.

An integral part of the scientific organization of mental labor is the mastery of the technique of mental labor.

The time that the student has to complete the curriculum consists of two components: one of them is classroom work at the university according to the schedule of classes, the other is extracurricular independent work. Tasks and materials for independent work are given during classes according to the schedule, in the same classes the teacher controls independent work, and also assists students in the correct organization of work.

Self-study will require intensive mental work, which must not only be properly organized, but also stimulated. At the same time, it is very important to be able to maintain a steady attention to the material being studied. The development of attention requires significant volitional efforts. That is why, if a student notices that he is often distracted during self-study, he needs to force himself to concentrate. Such a procedure must be done constantly, since this

is a training of attention. Sustained attention appears when a person treats a matter with interest.

It is necessary to properly organize your classes in time: 50 minutes - work, 5-10 minutes - break; after 3 hours of work, a break - 20-25 minutes. Otherwise, increasing fatigue will lead to instability of attention. A very significant factor affecting the increase in mental performance is systematic physical education. The organization of active recreation provides for the alternation of mental and physical activity, which completely restores a person's working capacity.

6. Independent work of a student is a necessary link in the formation of a researcher

The progress of science and technology, information technology leads to a significant increase in scientific information, which imposes higher requirements not only on the moral and ethical qualities of a person, but also, in particular, constantly increasing requirements in the field of education - renewal, modernization of general and professional knowledge, skills of a specialist.

Any education should act as a dynamic process inherent in a person and continuing throughout his life. Mastering scientific thought and the language of science is a necessary component in the self-organization of a future research specialist. This is understood not so much as the accumulation of knowledge, but the mastery of scientifically based ways of acquiring it. This, generally speaking, is the main task of the university.

The specificity of the university educational process, in the organization of which more and more space is given to the independent work of the student, is that it seems to be the last and most adequate link for the implementation of this task. For during the study at the university, the development of the style and skills of educational (cognitive) activity takes place, the rational nature of which will contribute to the constant renewal of the knowledge of a highly qualified university graduate.

However, there are certain difficulties before this path, in particular, the transition of the student from the synthetic process of learning in secondary school to the analytical one in higher school. This is due both to the new content of education (expansion of general education and deepening of professional training) and to new, hitherto unknown forms of training (lectures, seminars, laboratory classes, etc.). The student receives not only the knowledge provided by the program and teaching aids, but he must also become familiar with the ways of acquiring knowledge in such a way as to be able to assess what we know, how we know it and how we have achieved this knowledge. All this is achieved through one's own independent work.

This is also because self-acquired knowledge is more operational, it becomes personal property, as well as a motive for behavior, develops intellectual traits, attention, observation, criticality, and the ability to evaluate. The role of the teacher is mainly to guide the accumulation of knowledge (in relation to first-year students), and in the subsequent years of study, in the senior years, in the joint identification of problems and care for independent search

student, as well as control over their activities. It should be noted that it is not possible to limit oneself only to the acquisition of knowledge provided for by the program of the discipline studied, it is necessary to constantly deepen the knowledge gained, concentrating it on some narrow specific area that corresponds to the interests of the student. In-depth study of all subjects provided for by the program is possible in practice, and good organization of work allows you to save time, which creates conditions for a deep, systematic, interested study of the topic independently chosen by the student.

Of course, all the advice, examples, recommendations in this area given by the teacher, or certain publications, or other sources, do not guarantee any success without the manifestation of one's own activity in this matter, i.e. they do not give ready-made recipes, but should contribute to the analysis of one's own work, its goals, organization in accordance with individual characteristics. Taking into account personal capabilities, existing living and working conditions, skills, on the basis of these recommendations, it is possible to develop an individually justified set of methods, ways, find one's own style or improve it, in order to study certain material, have time to assess its significance, suitability and possibilities of its application, in order to ultimately ensure the success of one's studies from future professional activity

7. Methodical recommendations for students on certain forms of independent work

The system of university education implies a much greater independence of students in planning and organizing their activities. It can be very difficult for yesterday's schoolchildren to do this: if at school the daily control by the teacher forced them to constantly and systematically prepare for classes, then at the university the question of the level of knowledge arises closely before the student only during the exam period. For some, this situation turns into a temptation to devote the whole semester to free time ("when necessary, I will learn!"), and when the time comes for exams, there is so much material to be mastered that no memory is able to cope with it in the remaining period of time.

Work with literature.

When working with a book, it is necessary to select literature, learn how to read it correctly, and keep notes. For the selection of literature in the library, alphabetical and systematic catalogs are used.

It is important to remember that rational skills of working with a book are always a great saving of time and effort.

The correct selection of textbooks is recommended by the teacher who reads the lecture course. The necessary literature can also be indicated in the methodological developments for this course.

Studying the material from the textbook, you should move on to the next question only after the correct understanding of the previous one, describing on paper all the calculations and calculations (including those that are omitted in the textbook or given in the lecture for independent conclusion).

In the study of any discipline, independent individual work plays a large and important role.

Particular attention should be paid to the definition of the basic concepts of the course. The student should analyze in detail the examples that explain such definitions, and be able to build similar examples independently. It is necessary to achieve an accurate idea of what you are studying. It is useful to make reference notes. When studying the material from the textbook, it is useful to supplement the lecture notes in the notebook (in specially designated margins). There you should also note the questions selected by the student for consultation with the teacher.

It is recommended to highlight the conclusions obtained as a result of the study in the notes so that they are better remembered when rereading the notes.

Experience shows that many students are helped by compiling a list of reference signals containing the most important and most frequently used formulas and concepts. Such a sheet helps to memorize formulas, the main points of the lecture, and can also serve as a permanent reference for the student.

There are two types of reading; primary and secondary. *Primary reading* is attentive, unhurried reading, in which you can dwell on difficult places. After it, not a single incomprehensible tin should remain. The content may not always be understood after the primary reading.

The task of *secondary* reading is the complete assimilation of the meaning of the whole (this reading may not be the second, but the third or fourth).

Rules for independent work with literature.

As already noted, independent work with textbooks and books (as well as independent theoretical research of the problems identified by the teacher at lectures) is the most important condition for the formation of a scientific way of cognition. The main tips here can be summarized as follows:

- Make a list of books that you should get acquainted with;

"Do not try to remember everything that you will not need in the near future," G. Selye advises the student and young scientist, "just remember where you can find it" (Selye, 1987, p. 325).

- Such a list itself should be systematized (what is necessary for seminars, what is necessary for exams, what is useful for writing term papers and diploma papers, and what interests you outside the framework of official educational activities, that is, what can expand your general culture...).

- Be sure to write out all the imprints for each book (when writing term papers and diploma papers, this will save a lot of time).

- Figure out for yourself which books (or which chapters of books) should be read more carefully, and which should just be looked through.

- When compiling lists of literature, you should consult with teachers and supervisors (or even with more prepared and erudite fellow students), who will help you better orient yourself on what you should pay more attention to, and what is not worth wasting time on at all...

- Naturally, all books, textbooks and articles read should be taken notes, but this does not mean that it is necessary to take notes of "everything": you can briefly write down the main ideas of the author and sometimes give the most vivid and indicative quotes (with pages indicated).

- If the book is your own, then it is allowed to make brief notes in the margins of the book or at the end of the book, on blank pages, just make your own "subject index", where the most interesting thoughts for you are noted and the pages in the author's text are necessarily indicated (this is a very good tip that allows you to save time and quickly find "selected" passages in a variety of books).

- If you have not worked much with scientific literature before, then you should develop the ability to "perceive" complex texts; for this, the best technique is to learn to "read slowly", when you understand every word you read (and if the word is unfamiliar, then either with the help of a dictionary or with the help of a teacher, it is necessary to recognize it), and this may take

a considerable time (for some - up to several weeks and even months); experience shows that after that, by some "miracle", the student begins to literally swallow books and almost see "through the cover" whether this work is worthwhile or not...

- "Either read or flip through the material, but do not try to read quickly... If I am interested in the text, then reading, thinking and even fantasizing about it merge into a single process, while forced speed reading not only does not contribute to the quality of reading, but also does not bring the feeling of satisfaction that we get when thinking about what we have read," G. Selye advises (Selye, 1987. – P. 325-326).

- There is another effective way to optimize acquaintance with scientific literature – you should be carried away by some idea and look through all the books from the point of view of this idea. In this case, the student (or young scientist) will look for arguments "for" or "against" the idea he is interested in, and at the same time he will communicate with the authors of these books about his ideas and reflections... The only problem is how to find "your" idea...

Reading a scientific text is a part of cognitive activity. Its purpose is to extract the necessary information from the text. The effectiveness of the action largely depends on how conscious the reader is of his own inner attitude when referring to the printed word (to find the necessary information, to assimilate the information fully or partially, to critically analyze the material, etc.).

There are *four main attitudes in reading a scientific text*:

1. information retrieval (the task is to find, highlight the information sought)
2. assimilative (the reader's efforts are aimed at understanding and remembering as fully as possible both the information presented by the author and the entire logic of his reasoning)
3. analytical-critical (the reader seeks to critically comprehend the material, analyzing it, determining his attitude to it)
4. creative (creates in the reader a readiness in one form or another – as a starting point for his reasoning, as an image for action by analogy, etc. – to use the author's judgments, the course of his thoughts, the result of observation, the developed methodology, to supplement them, to subject them to a new check).

The existence of several types of reading *is associated with the presence of different attitudes to addressing a scientific text*:

1. bibliographic – viewing catalog cards, recommendation lists, summary lists of journals and articles for the year, etc.;

2. browsing – used to search for materials containing the necessary information, it is usually resorted to immediately after working with references and catalogs, as a result of such a review, the reader establishes which of the sources will be used in further work;

3. introductory – implies a continuous, sufficiently detailed reading of the selected articles, chapters, individual pages, the purpose is to get acquainted with the nature of the information, to find out what issues are submitted by the author for consideration, to sort the material;

4. studying – involves a thorough mastery of the material; in the course of such reading, the reader's trust in the author, readiness to accept the information presented, the attitude to the most complete understanding of the material is realized;

5. Analytical-critical and creative reading are two types of reading that are close to each other in that they participate in solving research problems. The first of them involves a directed critical analysis of both the information itself and the ways of its receipt and presentation by the author; the second is the search for those judgments, facts, on which or in connection with which, the reader considers it necessary to express his own thoughts.

Of all the types of reading considered, the main one for students is the learning one - it allows them to accumulate knowledge in various fields when working with educational literature. That is why this type of reading should be mastered in the first place. In addition, when mastering this type of reading, the main techniques are formed that increase the effectiveness of working with a scientific text.

The main types of systematized recording of what has been read:

1. Annotation is an extremely brief coherent description of the book (article) viewed or read, its contents, sources, nature and purpose;

2. Planning is a brief logical organization of the text, revealing the content and structure of the studied material;

3. Thesis is a concise reproduction of the author's main statements without involving factual material;

4. Quotation is a verbatim excerpt from the text of excerpts, extracts that most significantly reflect one or another thought of the author;

5. Note-taking is a brief and consistent presentation of the content of what has been read.

A synopsis is a complex way of presenting the content of a book or article in a logical sequence. A synopsis accumulates the previous types of notes, allows you to comprehensively cover the content of a book or article. Therefore, the ability to make a plan, theses, make extracts and other notes determines the technology of compiling a synopsis.

Methodical recommendations for compiling a synopsis:

1. Read the text carefully. Clarify the incomprehensible words in the reference literature. When writing down, do not forget to put the reference data in the margins of the note.

2. Highlight the main thing, make a plan.

3. Briefly formulate the main provisions of the text, note the author's argumentation.

4. Take notes of the material, clearly following the points of the plan. When taking notes, try to express the idea in your own words. Notes should be kept clearly.

5. Write down quotes correctly. When quoting, take into account the conciseness and significance of the thought.

In the text of the synopsis, it is desirable to give not only thesis statements, but also their proofs. When drawing up a synopsis, it is necessary to strive for the capacity of each sentence. The thoughts of the author of the book should be stated briefly, taking care of the style and expressiveness of what is written. The number of additional elements of the synopsis should be logically justified, the notes should be distributed in a certain sequence that corresponds to the logical structure of the work. Margins should be left for clarification and addition.

Mastering the skills of taking notes requires purposefulness and daily independent work from the student.

Laboratory classes.

It should be emphasized that only after mastering the lecture material from a certain point of view (namely, from the point from which it is presented at the lectures), it will be consolidated in practical classes both as a result of discussion and analysis of the lecture material, and with the help of performing tasks of laboratory work, tasks. With these

The student will not only learn the material well, but also learn how to apply it in practice, and will also receive an additional incentive (and this is very important) to actively study the lecture.

When performing tasks independently, it is necessary to justify each stage of the solution, based on the theoretical provisions of the course. If the student sees several ways to solve a problem (task), then it is necessary to compare them and choose the most rational one. It is useful to draw up a brief plan for solving the problem (task) before starting calculations. The solution of problems or examples should be presented in detail, calculations should be arranged in a strict order, separating auxiliary calculations from the main ones. Solutions, if necessary, should be accompanied by comments, diagrams, drawings and drawings.

It should be remembered that the performance of each educational task should be brought to the final logical answer required by the condition, and, if possible, with a conclusion. The answer obtained should be checked in ways that follow from the essence of the given task. It is also useful (if possible) to solve in several ways and compare the results obtained. For example, tasks are performed using NetCracker Professional 4.1 tools, then analytically and the results are compared to substantiate the result of the task. Laboratory work. Tasks of this type should be continued until you acquire solid skills in solving them.

Self-check.

After studying a certain topic from the notes in the notes and textbook, as well as solving a sufficient number of relevant tasks in laboratory classes and independently, the student is recommended, using a sheet of reference materials, to reproduce from memory definitions, conclusions of formulas, formulations of the main provisions and evidence.

If necessary, you need to carefully understand the material again.

Sometimes the insufficiency of assimilation of a particular issue becomes clear only when studying further material. In this case, it is necessary to go back and repeat the poorly mastered material. An important criterion for mastering theoretical material is the ability to solve problems or pass a test on the material covered. However, it should be remembered that the correct solution to the problem can be obtained as a result of the use of mechanically memorized formulas without understanding the essence of theoretical provisions.

Consulting

If in the process of independent work on the study of theoretical material or when performing tasks, the student has questions that cannot be solved independently, it is necessary to contact the teacher to obtain explanations or instructions from him. In his questions, the student should clearly express what he is experiencing difficulties, the nature of this difficulty. Advice should also be sought if there are doubts about the correctness of the answers to the self-test questions.

8. Independent Work of Students in the Conditions of the Point-Rating System of Education

The rating system of education involves multi-point assessment of students, but this is not a simple transition from a five-point scale, but an opportunity to objectively reflect in points the expansion of the range of assessment of individual abilities of students, their efforts spent on performing this or that type of independent work. There is a lot of room for creating a block of differentiated individual tasks, each of which has its own "price". A properly organized technology of rating education allows you to get away from the five-point assessment system from the very beginning and come to it only when summing up, when the points earned by students are converted into the usual grades (excellent, good, satisfactory, unsatisfactory). In addition, the rating system includes additional incentive points for originality, novelty of approaches to performing tasks for independent work or solving scientific problems. A student has the opportunity to increase his educational rating by participating in extracurricular work (participation in Olympiads, conferences; performing individual creative tasks, essays; participating in the work of a scientific circle, etc.). At the same time, students who are in no hurry to submit their work on time can also receive negative points. At the same time, individual students are encouraged to complete the program faster. For example, if a student is ready to take a test or write an independent work before the group, you can add additional points to him.

The rating system is a regular monitoring of the quality of knowledge and skills assimilation in the educational process, the implementation of the planned amount of independent work. Maintaining a multi-point assessment system allows:

On the one hand, to reflect the individual characteristics of students in the point range, and on the other hand, to objectively evaluate in points the efforts of students spent on the performance of certain types of work. Thus, each type of educational activity acquires its own "price".

The "cost" of the work performed by the student flawlessly is a quantitative measure of the quality of his learning in the totality of the educational material studied by him, which was necessary for the successful completion of the task.

The maximum possible score for the entire current control is set at 55. The current control activity is considered passed if the student received at least 60% of the maximum score established for this control. The rating score given to the student for the current control measure passed by the student within the time limits established by the schedule of control activities is determined as follows:

Level task	Performance of the control	Rated Score (in % from the maximum score for the control task)
	Excellent	100
	Good	80
	Satisfactory	60
	Unsatisfactory	0

The final grade for a discipline studied in one semester is determined by the sum of points scored for the work during the semester

Scale for converting the rating score in the discipline into a grade according to a 5-point system

Rating score in the discipline	Score on a 5-point system
88 – 100	Excellent
72 – 87	Good
53 – 71	Satisfactory
< 53	Unsatisfactory

The developed scale for converting the rating in the discipline into a final five-point grade is available and easily calculated by both the teacher and the student.

When using the rating system:

- the main emphasis is placed on the organization of active types of educational activities, the activity of students goes to the creative comprehension of the proposed tasks;
- In the relationship between the teacher and students, there is cooperation and co-creation, there is psychological and practical

the teacher's readiness for the fact of the individual originality of the "self-concept" of each student;

- a variety of stimulating, emotionally regulating, directing and organizing methods of intervention (if necessary) of the teacher in the independent work of students is assumed;
- the teacher acts as a teacher-manager and director of training, ready to offer students the minimum necessary set of teaching tools, and not only transmits educational information; the student acts as a subject of activity along with the teacher, and the development of his individuality acts as one of the main educational goals;
- Educational information is used as a means of organizing educational activity, and not as a goal of learning.

The rating system of education provides the greatest informational, procedural and creative productivity of independent cognitive activity of students, provided that it is implemented through the technologies of personality-oriented learning (problem, dialogue, discussion, heuristic, game and other educational technologies).

Most students have a positive attitude to such a system for tracking the results of their training, noting that the rating system of training contributes to the even distribution of their efforts during the semester, improves the assimilation of educational information, and provides systematic work without "rushes" during the session. A large number of various tasks offered for independent study, and different scales of their assessment allow the student to monitor his progress, and if desired, he always has the opportunity to improve his rating (by performing additional types of independent work) without waiting for the exam. The organization of the learning process within the framework of the rating system of education using various types of independent work allows you to get better results in teaching students in comparison with the traditional university system of education.

The use of the rating system allows you to achieve a more rhythmic work of the student during the semester, as well as activates the cognitive activity of students by stimulating their creative activity. The use of tests directly in the learning process, in independent work, is very effective

students. In this case, the student tests his knowledge himself. Without immediately answering the test task, the student receives a hint explaining the logic of the task and performs it a second time.

It should also be noted that automated training and teaching-control systems are increasingly penetrating into the educational process, which allow the student to independently study a particular discipline and at the same time control the level of mastering the material.

9. Educational, methodological and information support of the discipline

On this discipline, the teacher recommends the following literature:

The following are considered as the main literature:

1. Iversen V.B. Development of Teletraffic and Planning of Networks Elektronnyi resurs: uchebnoe posobie [Development of Teletraffic and Planning of Networks]. - Development of Teletraffic and Planning of Networks, 2019-12-20. Moscow: Internet University of Information Technologies (INTUIT), 2016. - 625 p. - The book is in the basic version of the EBS IPRbooks.

2. Reshetnikova I.V. Teoriya teletraffika: Elektronnyi resurs [Theory of teletraffic: Elektronnyi resurs] / Reshetnikova I.V. : textbook. - Rostov-on-Don: RGUPS, 2019. - 80 p. - ISBN 978-5-88814-923-2.

As additional literature:

1. Fundamentals of Infocommunication Technologies: Theory of Teletraffic textbook / E.D. Bychkov, V.A. Maistrenko, O.N. Kovalenko, D.N. Kovalenko; ed. by V.A. Maistrenko; Ministry of Education and Science of Russia; Omsk State Technical University. - Omsk: OmSTU Publishing House, 2017. - 156 p. : graph., scheme. - <http://biblioclub.ru/>. - References: p. 130-131. - ISBN 978-5-8149-2433-9.

2. Bratchenko N.Y. Theory of Teletraffic Elektronnyi resurs : uchebnoe posobie [Theory of Teletraffic: Electronic Resource]. - Stavropol: North-Caucasian Federal University, 2014. - 177 p. - The book is in the basic version of the EBS IPRbooks.

Students can also use methodological literature:

Theory of Teletraffic : Textbook : Direction of training 210700.62 (11.03.02)
– Infocommunication Technologies and

Communication Systems. Profile of training "Communication Networks and Switching

Systems" / author. Bratchenko; FGAOU VPO Sev.-Kav. Federal. University. - Stavropol: NCFU, 2014. - 55 p. : ill.

The following sources are offered as Internet resources:

1. <http://biblioclub.ru>
2. <http://www.iprbookshop.ru>
3. <http://catalog.ncfu.ru/catalog/ncfu>

For independent development theoretical and

The following software will be required:

1. The operating system of the Microsoft Windows family.
2. MathCad.
3. MS Office 2010.
4. MatLab.
5. Net Cracker 4.1.
6. MS Power Point.