MINISTRY OF EDUCATION AND SCIENCE
OF THE REPUBLIC OF KAZAKHSTAN

M.Yu. Zarubin, G.S. Ybytaeva

CRYPTOGRAPHIC SYSTEMS

КРИПТОГРАФИЧЕСКИЕ СИСТЕМЫ

TEXTBOOK
УЧЕБНОЕ ПОСОБИЕ

Almaty, 2021

УДК 004(075.8)
ББК 32.973я73
Z 32

PRecommended for publication by
Scientific Council of the Rudny Industrial Institute

Reviewers:

Tsekhovoy A.F. – Doctor of Technical Sciences, Professor, General Director of the International Academy of Informatization

Baymukhamedov M.F. – Doctor of Technical Sciences, Professor, Vice-Rector for Scientific Work of the KSTU

Baganov N.A. – Ph.D., Acting professors, vice-rector for strategic development, science and innovations of KINEU named after M. Dulatova

Oleinik A.I. – Doctor of Technical Sciences, Professor of the RII

Zarubin M.Yu., Ybytaeva G.S.
Z 32 **Cryptographic Systems – Криптографические системы:** textbook – учебное пособие / M.Yu. Zarubin, G.S. Ybytaeva. – Almaty: «Bastau», 2021. – 320 page.

ISBN 978-601-7660-08-6

Almost for the first time in the textbook, in the English and Russian languages, a systematic presentation of the scientific foundations of cryptography is given: from basic concepts and the simplest examples to methods for constructing modern cryptographic systems.

The book summarizes the latest advances in cryptography. Special attention is paid to two areas: cryptosystems with symmetric and public (public) keys and key management methods. Also considered are cryptographic protocols, which opened a new milestone in the use of information protection methods in various computer networks, and a review of cryptanalysis methods is made. The book is intended for information security specialists and students of IT specialties, but it will also be useful and interesting to the general reader who thinks about the security of their own information.

В учебном пособии практически впервые на английском и русском языках дается систематическое изложение научных основ криптографии: от основных понятий и простейших примеров до методов построения современных криптографических систем.

В книге обобщаются последние достижения в области криптографии. Особое внимание уделено двум направлениям: криптосистемам с симметричным и открытым (публичным) ключами и методам управления ключами. Также рассмотрены криптографические протоколы, которые открыли новую веху в использовании методов защиты информации в различных компьютерных сетях, и произведен обзор методов криптоанализа.

Книга рассчитана на специалистов по информационной безопасности и на студентов IT-специальностей, но также будет полезна и интересна массовому читателю, задумывающемуся о вопросах безопасности собственной информации.

УДК 004(075.8)
ББК 32.973я73

ISBN 978-601-7660-08-6

TABLE OF CONTENTS
СОДЕРЖАНИЕ

Introduction....8
Введение 156

CHAPTER 1. HISTORY OF CRYPTOGRAPHY 10
ГЛАВА 1. ИСТОРИЯ КРИПТОГРАФИИ....158
1.1 Introduction to the History of Cryptography 10
1.1 Введение в историю криптографии 158
1.2 Naive Cryptography....13
1.2 Наивная криптография 161
1.3 Formal Cryptography....17
1.3 Формальная криптография....166
1.4 Scientific Cryptography 27
1.4 Научная криптография 176
1.5 Computer Cryptography 28
1.5 Компьютерная криптография 178
1.6 Unknown Cryptography 30
1.6 Неизвестная криптография 178

CHAPTER 2. SYMMETRICAL CRYPTOSYSTEMS 36
ГЛАВА 2. СИММЕТРИЧНЫЕ КРИПТОСИСТЕМЫ....187
2.1 Main classes of symmetrical cryptosystems 36
2.1 Основные классы симметричных криптосистем 187
2.2 Substitutions, Permutations, and Gamming....39
2.2 Подстановки, перестановки и гаммирование 191
2.2.1 Encryption Using Permutations 39
2.2.1 Шифрование с помощью перестановок....191
2.2.2 Encryption Using Substitutions 47
2.2.2 Шифрование с помощью подстановок 199
2.2.3 Gumming....67
2.2.3 Гаммирование 220
2.3 Key Generators 68
2.3 Генераторы ключей 221
2.4 Block Ciphers....72
2.4 Блочные шифры....226
2.5 Stream Ciphers....97
2.5 Потоковые шифры 252

CHAPTER 3. OPEN (PUBLIC) KEY CRYPTOGRAPHY SYSTEMS 105
ГЛАВА 3. СИСТЕМЫ КРИПТОГРАФИИ С ОТКРЫТЫМ (ПУБЛИЧНЫМ) КЛЮЧОМ 261
3.1 One-Way Functions and Hook Functions 105
3.1 Односторонние функции и функции-ловушки 261
3.2 Cryptosystem RSA 106
3.2 Криптосистема RSA 263
3.3 Cryptosystems Based on Elliptic Curves 110
3.3 Криптосистемы, основанные на эллиптических кривых 266
3.4 Hash Functions and Hashing 116
3.4 Хэш-функции и хэширование 273
3.5 Digital Signature 122
3.5 Электронно-цифровая подпись 279
3.6 Prospects of Development and Problems of Application of Cryptosystems with Open (public) Key 127
3.6 Перспективы развития и проблемы применения криптосистем с открытым (публичным) ключом 285

CHAPTER 4. CRYPTOGRAPHIC KEY MANAGEMENT 131
ГЛАВА 4. УПРАВЛЕНИЕ КРИПТОГРАФИЧЕСКИМИ КЛЮЧАМИ 289
4.1 Key Management in Systems with Open (Public) Key 131
4.1 Управление ключами в системах с открытым (публичным) ключом 289
4.2 Secret Key Exchange Protocol 134
4.2 Протокол обмена секретным ключом 293
4.3 Safety Certifications 137
4.3 Сертификаты безопасности 296
4.4 Anonymous Key Distribution 139
4.4 Анонимное распределение ключей 298

CHAPTER 5. BRIEF INFORMATION ABOUT CRYPTO ANALYSIS 141
ГЛАВА 5. КРАТКИЕ СВЕДЕНИЯ О КРИПТОАНАЛИЗЕ 301
5.1 Brief History of Cryptanalysis 141
5.1 Краткая история возникновения криптоанализа 301
5.2 Methods of Cryptanalysis 142
5.2 Методы криптоанализа 302

Conclusion 148
Заключение 309

List of Recommended Sources 150
Список рекомендованных источников 311

CRYPTOGRAPHIC SYSTEMS

INTRODUCTION

The issues of safe transfer of classified information have become vital for the mankind probably since the time of the appearance of the first tribal communities and states. The development and massive use of information transmission means in the XVIII-XIX centuries in relation to the state and business only exacerbated this problem of ensuring the secrecy of the transmission of important information. The emergence in the twentieth century of social networks, e-business mechanisms and the territorial distribution of storages and consumers of information made this problem extremely urgent for a modern digital society – now not only government structures and businesses but also individuals are forced to solve the problems of ensuring security of their own information and protecting their data from unauthorized use.

The textbook «Cryptographic Systems» is able to help to the reader who is dealing with the problem of ensuring the security of information transmission and storage in both building basic modern cryptographic systems and using cryptographic protocols when building their own information security systems. The manual proposes to consider two main approaches in building crypto-protected systems: symmetric encryption and encryption with a public key. For these approaches, the main mechanisms and algorithms for implementation are reviewed, an analysis of strengths and weaknesses is provided. We also review the key management issues to ensure a systematic approach to security and cryptographic protocols which opened a new milestone in the use of information protection methods in various computer networks.

The authors express their deep gratitude to the reviewers: General Director of the International Academy of Informatization, ***Professor Aleksey Filippovich Tsekhovy***, Doctor of Technical Sciences, Professor of the Kostanay Social and Technical University named after Academician Z. Aldamzhar ***Baimukhamedov Malik Fayzullovich***, Candidate of Technical Sciences, Acting Professor of Kostanay Engineering and Economic University named after M. Dulatov ***Nikolai Anatolyevich Baganov***, Head of the Department of Computer Engineering and Software of the Akhmet Baitursynov Kostanay State University, Associate Professor ***Olga Sergeevna Salykova*** and Doctor of Technical Sciences, Professor of the Rudny Industrial Institute ***Oleinik Alexander Ivanovich***.

The authors will carefully and with gratitude consider all critical comments and suggestions related to further improvement of this tutorial.

All comments and suggestions please send to the address: 111500, Republic of Kazakhstan, Rudny, st. 50 years of October, 38 or by e-mail zarubin_mu@mail.ru or gali17@mail.ru.

CHAPTER 1. HISTORY OF CRYPTOGRAPHY

Keywords: primitive cryptography, formal cryptography, scientific cryptography, computer cryptography, substitution, permutation, scital, Polybius square, Aeneas' disc and ruler, atbash, edduba, Black cabinet, perlustration, Vernam cipher, rotary cipher machine, Jefferson cylinder, Enigma, Turing Bombe, block cipher, public key encryption, Voynich manuscript, fest disc, Rohonzi Code, kipu, probabilistic encryption, quantum cryptography, lattice cryptography , honey encryption, functional encryption, homomorphic encryption, DNA encryption.

1.1 Introduction to the History of Cryptography

«Who owns the information, he owns the world!» – the famous phrase of Nathan Rothschild as never before accurately reflects the importance of information and its safety from unauthorized use. Therefore, it is not surprising that mankind is interested in methods of protecting information from strangers.

Cryptography has become one of the most dynamically and successfully developing sciences that provide protection of information from unauthorized access.

The word «cryptography» comes from a combination of ancient Greek words «Κρυπτός» («hidden») and «γράφω» («I write»).

In its present understanding, «cryptography» is the science of methods of ensuring confidentiality (the impossibility of reading information by strangers), data integrity (the impossibility of imperceptible changes to information), authentication (verification of authorship or other properties of an object), and also the impossibility of denial of authorship.

We are accustomed to using methods and means of cryptography in authentication for e-mail, chats, computer games, electronic payment systems, protecting our data during storage and transmission in the lines of computing systems, confirming our actions by means of an electronic digital signature. And, often, we do not even think about how it is all implemented.

In this section, material will be presented on how cryptographic systems appeared and developed from the ancient world to ultra-modern solutions. The known history of cryptography is about 4 thousand years old[1]:

[1] As the main criterion for the periodization of cryptography, modern historians use the technological characteristics of the encryption methods used.

– ***the first period*** (approximately from the III millennium BC). This period was also called *naive cryptography.* It is characterized by the dominance of the simplest mono-alphabetic ciphers;

– ***the second period*** (from the IX century in the Middle East and from the XV century in Europe – to the beginning of the 20th century) was marked by the introduction of polyalphabetic cipher systems into use and was called *formal cryptography.* The period is associated with the emergence of formalized and relatively resistant to manual cryptanalysis ciphers. In European countries, this happened during the Renaissance, when the development of science and trade created a demand for reliable ways to protect information. During this period, in the XIX century, the Dutchman Kerkhoffs formulated the main requirement for cryptographic systems, which remains relevant to this day: the secrecy of ciphers should be based on the secrecy of the key, not the secrecy of the algorithm;

– ***the third period*** (30s-60s of the XX century) is characterized by the use of a rigorous mathematical apparatus for constructing cryptosystems and the introduction of electromechanical devices into the work of encryptors. By the beginning of the 30s, the branches of mathematics were finally formed, which are the scientific basis of cryptology: probability theory and mathematical statistics, general algebra, number theory, the theory of algorithms, information theory, and cybernetics began to actively develop. Claude Shannon's work «Theory of communication in secret systems» which formulates the theoretical principles of cryptographic information protection, became a kind of watershed. Shannon introduced the concepts of «dispersion» and «mixing», substantiated the possibility of creating arbitrarily strong cryptosystems. In the 60s, the leading cryptographic schools approached the creation of block ciphers, even more secure in comparison with rotary cryptosystems, but allowing practical implementation only in the form of digital electronic devices;

– ***the fourth period*** began in the mid-70s of the XX century – this is the period of transition to computer cryptography. Block ciphers became the first class of crypto-systems, the practical application of which became possible due to the advent of computing means. In the 70s, the American DES encryption standard was developed. One of its authors, Horst Feistel, proposed approaches for constructing block ciphers, on the basis of which other, more stable symmetric cryptosystems were later built.

With the advent of DES, cryptanalysis also enriched; for attacks on this cryptoalgorithm, several new types of cryptanalysis (linear, differential,

etc.) were created, the practical implementation of which, again, was possible only with the advent of powerful computing systems.

In the mid-70s, there was a real breakthrough in modern cryptography – the emergence of cryptosystems based on the use of a public key. Here, the starting point is considered to be a work published by Whitfield Diffie and Martin Hellman in 1976 entitled «New Directions in Modern Cryptography.» It was the first to formulate the principles of exchanging encrypted information without exchanging a secret key. A few years later, Ron Rivest, Adi Shamir, and Leonard Adleman introduced the RSA cryptosystem to the world, based on the use of public and private keys. Asymmetric cryptography opened several new applied areas at once, in particular: electronic digital signature (EDS) systems and electronic money. The task of improving symmetric cryptosystems remains relevant for this period. In the 80s and 90s, GOST 28147-89, non-Faystel ciphers (SAFER, RC6, etc.) were developed, and in 2000, after an open international competition, a new US national encryption standard, AES, was adopted.

In recent years, completely new directions of cryptography have appeared. For example:

- probabilistic encryption of Shafi Goldwasser;
- quantum cryptography[2] by Stephen Wiesner;
- lattice cryptography formulated by Cecilia Bocini;
- homomorphic encryption, proposed in 1978 by Ronald Rivest, Leonard Aldeman and Michael Dertuzos;
- Honey Encryption presented at the Eurocrypt conference in Copenhagen in 2015 by Ari Jules and Thomas Ristenpart;
- functional encryption – formulated in the early 90s of the twentieth century by Whitfield Diffie and Martin Hellman whose concept is considered one of the most promising for public key encryption;
- DNA encryption by Leonard Adleman.

No matter how fantastic these directions sound, most likely they (and maybe completely different solutions) will be at the heart of cryptography in the XXI century. For example, in 1989, Bennett and Brassard at the IBM Research Center built the first working quantum cryptographic system. Awareness of the practical value of these scientific studies, we assume, will allow us to single out the fifth stage in the development of cryptography.

[2] Modern cryptographers are already raising questions about the creation of post-quantum cryptography.

1.2 Naive Cryptography

The history of cryptography is more than four thousand years old and appeared in parallel with the advent of writing. Probably the most ancient centers for the emergence of cryptographic transformations were Mesopotamia, India, China. A much later period is characterized by the better studied and illuminated encryption methods of Ancient Greece.

The first known application of cryptography is considered to be the beginning of the use of special hieroglyphs about 4000 years ago in Ancient Egypt. Elements of cryptography were found already in the inscriptions of the Old and Middle Kingdoms (periods III-VI and XI-XII dynasties of the pharaohs), completely cryptographic texts are known from the period of the XVIII dynasty of the Egyptian pharaohs. Hieroglyphic writing originated from pictography, it uses ideograms (a written sign or conventional image, a drawing corresponding to a certain idea of the author) and, as a result of the lack of vocalization, made it possible to create phonograms according to the principle of puzzles. The cryptography of the Egyptians was used, most likely not with the aim of making it difficult to read but more likely with the desire of scribes to surpass each other in wit and ingenuity, and also with the help of unusualness and mystery to draw attention to their texts. One of the illustrative examples of such «cryptograms» are the texts of glorification of the «chief of the East» Khnumhotep II (XIX century BC) found in the area of Beni-Khasan. In ancient Indian manuscripts more than sixty ways of writing are given, among which there are some that can be considered cryptographic. There is a description of the system of replacing vowels with consonants and vice versa.

Figure 1.1 – Mesopotamian clay tablet of Eddub and tablet with Egyptian hieroglyphs

One of the ancient Indian documents on cryptography is ... Kamasutra. Compiled in the fourth century BC, it contains a description of 64 arts (yogas) that every woman must master. Among them are such familiar skills as cooking and drinking, the art of choosing an outfit, preparing aromas, and the skill of doing massage. But in its Chapter 3 the special art of «Mlecchita vikalpa» is indicated under Number 44 which is described as «the art of understanding writing in cipher and writing words in a special way.»

In Mesopotamia[3] the unknown author of the tablet with the recipe for the manufacture of glaze for pottery used rare designations, omitted letters, and replaced names with numbers to hide what was written.

Similar encryption algorithms were used in ancient China.

However, cuneiform, drawing and hieroglyphic writing in itself was extremely difficult and required lengthy training, so the question of encrypting messages was often simply not raised, since the number of literate people was minimal. It is impossible to judge the breadth of distribution of various cryptographic systems and secret writing of that period, since the number of artifacts and records that have come down to us is very small.

With the advent of phonetic writing in the II-I millennium BC, writing became much simpler, which made it more accessible. Accordingly, the importance of cryptography has also increased.

The states of ancient Greece and Rome are considered one of such centers. In these states, some of the most famous cryptographic transformations and devices were used: the scital, the disc and ruler of Aeneas, the square of Polybius and, a little later, the code of Julius Caesar.

One of the most ancient cryptographic devices that have come down to us is the scital[4] (from the Greek σκυτάλη «wand»). A scitala is a wand (cylinder) and a narrow strip of papyrus or parchment (leather) wrapped around it in a spiral. After that, the text of the message was applied to it. When the strip was unwound from the wand, the letters of the message lost their order – the simplest rearrangement took place. The recipient of the message for decryption must have a rod of the same diameter.

---

[3] The states in the interfluve of the Tigris and Euphrates (Mesopotamia) are better known to us as the Sumerians, Babylon and Assyria (the period from 3200 to 100 BC). The first schools for training scribes in Mesopotamia were called «houses of tablets» (in Sumerian «edubba»). They were named after the clay tablets on which the cuneiform was applied.

[4] A permutation cipher, implemented by means of a scital, is also called a Spartan cipher.

Figure 1.2 – Scytala and a fragment of a fresco of its use by the Spartans

Apparently, initially the Greeks used the script for the convenience of writing (because in the early mentions they wrote poetry, among other things), and somewhere from the 4th century BC they began to use it as a tool for secret writing.

The Athenians, or rather Aristotle, are credited with inventing a method for decrypting texts written using scital. It was enough to wrap a strip of intercepted parchment with a secret message around a rather long cone at its base, and then gradually move to the top of the cone. Where the diameter of the cone coincided with the diameter of the scital, the letters on the parchment were combined into syllables and words.

The name of Aeneas Tacticus, the commander of the IV century BC. E., is linked to several techniques of encryption and cryptography. These are two devices – Aeneas's disc and ruler – and Aeneas's book cipher.

Aeneas's disk was a wooden or copper disk 10-15 centimeters in diameter with holes according to the number of letters of the alphabet. Each hole was assigned a specific letter. In the center of the disk was a coil with a cord wound on it. To record the message, the string was pulled through the holes in the disc corresponding to the letters of the message. When reading, the recipient pulled the string, and received the letters, however, in the reverse order. Although the ill-wisher could read the message if he intercepted the disk, Aeneas also provided a way to quickly destroy the message – for this it was enough to pull the thread.

The first truly cryptographic tool can be called Aeneas' line, which implements the replacement cipher. Instead of a disc, a ruler with holes according to the number of letters of the alphabet, a coil and a slot was used. For encryption, the thread was pulled through the slot and hole, after which another knot was tied on the thread. For decryption, it was necessary to have the thread itself and a ruler with a similar arrangement of holes.

Thus, even knowing the encryption algorithm, but without a key (ruler), it was impossible to read the message.

Also associated with the name of Aeneas Tactics is the use of subtle marks in the text of the document (for example, needle punctures, placed next to the letter). Such markings make it possible to isolate meaningful characters from the general text – the hidden text. This concealment of information was named the book cipher of Aeneas[5].

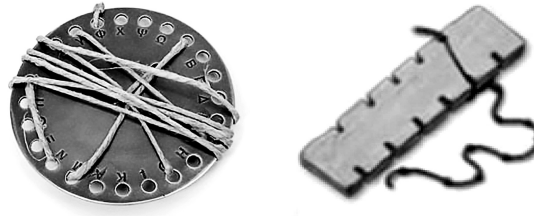


Figure 1.3 – Aeneas's disk and ruler

One of the ancient Greek substitution ciphers that have come down to us was an algorithm proposed by the philosopher and commander Polybius, who lived in the second century BC, called the «Polybius square». The algorithm key was a 5x5 square, in which letters of the Greek alphabet were written in random order. In a cipher message, usually transmitted by a heliograph, line numbers and column numbers of letters were transmitted sequentially. When receiving a message, to decrypt it, exactly the same square was required – a key.

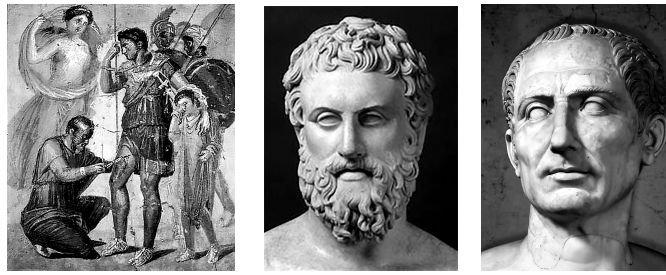


Figure 1.4 – Aeneas the Tactician (IV century BC), Polybius (206-124 BC), Gaius Julius Caesar (100-44 BC)

[5] The first mention of the use of a book cipher is found in the work of Aeneas the Tactician, «On the transfer of the siege,» where he proposes this method for secret writing. Much later, a similar code was used by German spies in the First World War.

Probably the most famous is the cryptoalgorithm used by the Roman patrician and great pontiff Guy Julius Caesar to encrypt his correspondence with his generals and friends. Julius Caesar, according to Suetonius' Life of the Twelve Caesars, used it with a shift

3. Although Caesar was the first recorded person to use this scheme, other substitution ciphers are known to have been used in the past. There is also evidence that Caesar used more complex ciphers.

The Atbash cipher is also worth mentioning. The cipher is believed to have been invented by the Essenes, a Jewish rebel sect, in order to protect their followers from disclosure by the authorities and avoid executions. The knowledge of these codes and ciphers was then taken over by the Order of the Templar. Thus, the atbash cipher was used for many hundreds of years (from about 500 BC to 1300 AD).

Even then, encrypted correspondence was used not only by statesmen and military leaders, but also by the church and scientists. The priests ciphered the texts of the soothsayers, and the scientists – their works. For example, E.Shure in his book "The Great Initiates" has a phrase that "with great difficulty and great price Plato obtained one of the manuscripts of Pythagoras, who never wrote down his teachings other than secret signs and under various symbols".

Unfortunately, the writing of most American and African peoples of that time did not reach our days, therefore, the use of cryptographic methods in these countries can now be attributed to the mysteries of history.

Summing up, we can say that all naive cryptography algorithms (up to the beginning of the XVI century) are characterized by the use of any (usually primitive) methods of confusing the enemy regarding the content of encrypted texts. At the initial stage, encryption and steganography methods were used to protect information, while most of the encryption algorithms used were reduced to permutation or mono-alphabet substitution (see Chapter 2).

1.3 Formal Cryptography

Formal cryptography is largely associated with the history of Renaissance Europe[6]. During this period, both public and private correspondence was actively developing.

[6] Before the Renaissance in Christian Europe, cryptography was considered a «dark» art and mixed with Kabbalah.

Naturally, this also gives rise to the rapid development of all kinds of cryptographic methods for protecting this correspondence from prying eyes.

The most characteristic for this period are again monoalphabetic substitution ciphers and permutation ciphers. During this period, the first scientific works on cryptography appeared.

The first currently known European book describing the use of cryptography is Roger Bacon's The Message of the Monk Roger Bacon on the Secret Actions of Art and Nature and the Insignificance of Magic, which describes, among other things, the use of 7 methods of hiding text. He is also credited with the authorship of the mysterious Voynich manuscript. In the XIV century, Cicco Simoneti, an employee of the secret office of the papal curia, wrote a book on cryptography systems, and in the 15th century, Pope Clement's secretary

XII Gabrielle de Levinda, completing his Treatise on Ciphers.

Another well-known result belongs to the pen of the German abbot Johann Trithemius (or Tritemus), who is considered by many historians to be the second father of modern cryptology. He becomes the author of the first printed book on cryptography. In the fifth book of the Polygraphia series, published in 1518, he described a cipher in which each successive letter is encrypted with its own shift cipher. His approach was refined by Giovan Battista Bellaso. In addition, Trithemius was the first to notice that it is possible to encrypt two letters at a time – with bigrams (see Chapter 2).

The next step in the development of cryptography was taken by Giovanni Porta, a renowned Italian naturalist. In 1563 he wrote a book

«On Secret Correspondence», which provides a description of all known cipher systems. It also provides a description of the bigram cipher in which pairs of letters are replaced. Porta anticipated what is called the «probable word method» and provides examples of lists of probable words from various fields.

In the same period, the first organization appears to devote itself entirely to cryptography. It was created in Venice in 1452. Three secretaries of this organization were engaged in breaking and creating ciphers on the instructions of the government.

In 1626, during the siege of the city of Realmont during the Huguenot uprising in France, and later in 1628 during the siege of La Rochelle, Antoine Rossignol (1600-1682) deciphered the intercepted messages and thereby helped the king to defeat. After the victory, the French government several times involved him in decrypting ciphers. After the death of Rossignol, his son, Bonaventure, and later his grandson, Antoine-Bonaventure Rossignol,

continued his work which later resulted in the creation of a special service for perlustration and cryptanalysis in France - the so-called black cabinets[7]. England also had its own «black cabinet». In his work in the 17th century. a prominent place was occupied by John Wallis, known as the greatest English mathematician before Isaac Newton. In Germany, the head of the first decryption department was Count Gronsfeld, who created one of the options for improving the Vigenere cipher. In the Russian Empire at this time, a digital chamber was created for the same purposes.

Figure 1.5 – Roger Bacon (1220-1292), Antoine Rossignol (1600-1682), John Wallis (1616-1703)

More developed and known for the period of formal cryptography is Arabic cryptography. The level of development of mathematics and other sciences during this period was significantly ahead of the knowledge of the peoples of Europe. Therefore, the period of development and the complexity of the ciphers of the Arab world up to the X-XII centuries was significantly ahead of Europe. Even the word cipher is of Arabic origin.

The most ancient is the scientific work of the Arab scientist Abu Bakr Ahmed ben-Ali ben-Wakhshiyya al-Nabati dated 855. It mentions various encryption systems based on the symbols of ancient peoples. These ciphers were used until the beginning of the 19th century to encrypt secret correspondence, reports of spies, and treatises on black magic. The knowledge of the Arabs in the field of cryptography was presented by Shehabeh Kalkashandi, the author of an encyclopedia written in 1412. He included a whole section on the use of encryption systems based on both

[7] In 1911, the Britannica Encyclopedia wrote that the «black offices» no longer exist, but in fact, in one form or another, services for perlustration and decryption of correspondence existed at that moment and later, despite the existing laws on the secrecy of correspondence.

permutations and substitution (including plural) characters. In addition, great attention was paid to the opening of encrypted messages[8].

Figure 1.6 – Cabinet of perlustration at the post office in the Russian Empire (XIX century), an analogue of the French «black cabinets»

In Europe and Asia, ciphers called *nomenclators* became widespread during this period. They combine simple replacement ciphers and encoding. In the simplest nomenclators, the code consisted of several dozen words or phrases with the corresponding two-letter code designations.

Mono-alphabetic ciphers are also being replaced by polygram and polyalphabetic substitution ciphers (see Chapter 2) which were first suggested by Leon Battista Alberti. Alberti also proposed a device of two discs fastened in the center each of which had an alphabet written on the edge and could rotate relative to the other disc. While the disks do not move they allow encryption using the Caesar cipher but after a few words, the disks are rotated and the shift key changes.

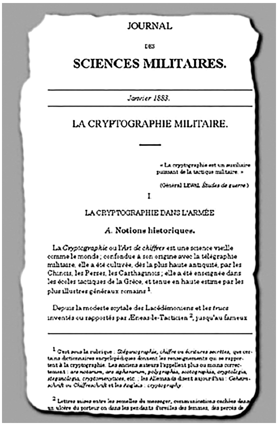
In 1883, cryptology received new ideas outlined in a work called «Military Cryptography» by Auguste Kerkhoffs. Based on his knowledge in the field of linguistics and mathematics, Kergoffs conducts a comparative analysis of ciphers, on the basis of which he formulates the requirements for ciphers and concludes that only those ciphers are of practical interest that remain strong even with intensive correspondence.

[8] The proposed methods were based on the statistical and linguistic properties of the language. Based on the text of the Quran, the encyclopedia provided statistics of all symbols of the Arabic language and an example of opening someone else's message.

Kerkhoffs formulated the principle that became the basis of modern cryptology – the strength of a cryptographic system should depend not on the secrecy of the encryption algorithm, but on the cryptographic strength and secrecy of the key used. This principle has not lost its relevance today.

Equally valuable is Kerkhoffs' idea that the reliability of a cipher should be evaluated by decryptors. Of course, this was suspected before him, but after the closure of the «black offices» they somehow forgot. In any case, the inventors of new ciphers, instead of submitting them to the trial of cryptanalysts, sought to evaluate their strength on their own, counting the number of centuries required for sequential enumeration of all possible keys, or tried to prove the impossibility of «breaking through» any of the cipher elements.

Kerkhoffs wrote: «I am amazed that our scientists and professors teach and recommend systems for use in wartime, the keys to which will undoubtedly be revealed in less than an hour by the most inexperienced cryptanalyst ... One can also assume that the absence of serious work on the art of reading secret writing contributed to the spread of the most erroneous ideas about the strength of our cipher systems».



JOURNAL

DES

SCIENCES MILITAIRES.

Janvier 1883.

LA CRYPTOGRAPHIE MILITAIRE.

« La cryptographie est un auxiliaire puissant de la tactique militaire. »

(Général LEWAL, *Études de guerre*)

I

LA CRYPTOGRAPHIE DANS L'ARMÉE

A. Notions historiques.

La *Cryptographie* ou l'*Art de chiffrer* est une science vieille comme le monde ; confondue à son origine avec la télégraphie militaire, elle a été cultivée, dès la plus haute antiquité, par les Chinois, les Perses, les Carthaginois ; elle a été enseignée dans les écoles tactiques de la Grèce, et tenue en haute estime par les plus illustres généraux romains [1].

Depuis la modeste scytale des Lacédémoniens et les *trucs* inventés ou rapportés par Æneas-le-Tacticien [2], jusqu'au fameux

Figure 1.7 – Jean Wilhelm Hubert Victor Francois Alexander Auguste Kerckhoffs von Niuvengoff (1835-1903) and Military Cryptology

Thanks to the work of Kergoffs, in all the leading countries of the world already in the 80s of the XIX century, cryptography is recognized as a science and without fail begin to teach in military academies.

The end of the 19th and the beginning of the 20th centuries was characterized by the massive appearance, first of the telegraph, and then

of the radio for transmitting information. Naturally, the possibilities of intercepting information have also increased. Therefore, this period of development is also characterized by the emergence of ciphers and systems oriented to the transmission of information through electrical and radio communications.

To encrypt telegraph messages, a whole galaxy of codes and cipher algorithms is being developed, the apogee of which is the emergence of an absolutely reliable cipher – the Vernam cipher.

In addition, to encrypt teletype messages, Vernam proposed to prepare in advance a «gamma» – a punched tape with random characters - and then electromechanically add its pulses with pulses of plain text characters. The amount received was a ciphertext. At the receiving end, the pulses received via the communication channel were added to the pulses of the same «gamma», as a result of which the original message pulses were restored. And if the message was intercepted, then it was impossible to decipher it without the «scale» the enemy saw only a meaningless sequence of «pluses» and «minuses».

During the First World War, codes were the main (and often the only) means of encryption. Despite the fact that all participants in the hostilities constantly developed new codes and improved old ones, it was far from always possible to ensure their safety, so the opponents were often fully aware of everything that was contained in someone else's secret correspondence. A number of tragic events are associated with the use of ciphers, of which we will only mention the defeat of two Russian armies – Generals Rannenkamp and Samsonov in East Prussia in August 1914. The reason for the defeat was, among other things, the poor organization of closed communications, as a result of which negotiations on the radio were conducted without any encryption at all.

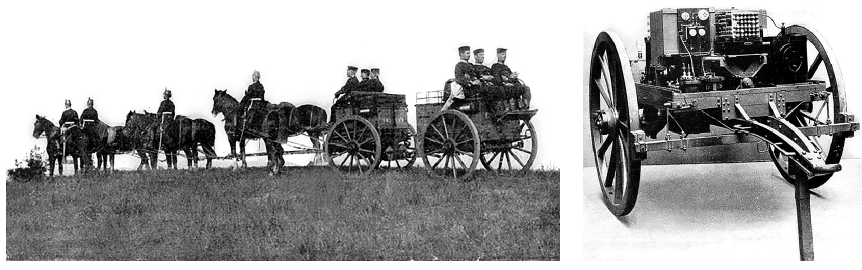


Figure 1.8 – Horse small (field) military radio station with a combat crew of the early XX century

The Second World War further raised the requirements for transmission speed and secrecy of information. Manual conversion algorithms and codes still somehow solved the problems of illegal intelligence, but were absolutely inapplicable for the belligerent armies and navies. It was necessary to increase the cryptographic strength of ciphers and to automate (more precisely, mechanization) encryption processes.

One of the first such systems was invented back in 1790 by Thomas Jefferson, the future president of the United States, a mechanical machine – the Jefferson cylinder.

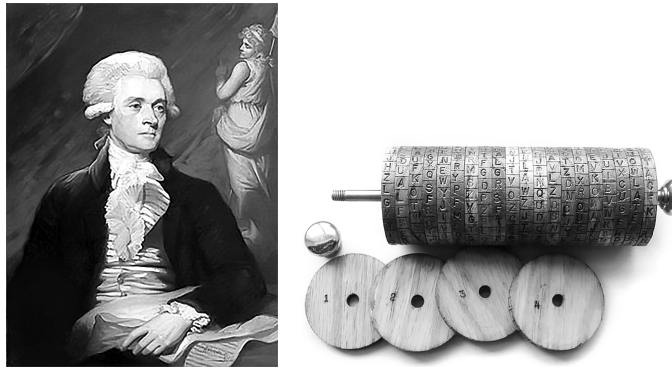


Figure 1.9 – Thomas Jefferson (1743-1826) and Jefferson Cylinder

Jefferson called his encryption system «disk cipher». However, he himself was not sure of the reliability of his invention, so he treated it with caution and, as President of the United States, did not use it, but continued to use traditional codes and ciphers.

When using the Jefferson cylinder, it is enough to rotate the disks so that the desired source text appears on the edges of the disks in the line, and write the line from any other edge. After receiving the cipher message and typing it on the Jefferson cylinder, the original text appeared on one of the faces. Naturally, the cylinders of the sender and the recipient must be identical.

However, mechanical encoders received practical distribution only at the beginning of the XX century. One of the first machines in practice was the rotary machine, developed in 1917 by Edward Hebern. And on February 23, 1918, the German engineer Arthur Scherbius was granted a patent for the Enigma encryption machine, which became the legend of rotary encryption machines.

«Enigma» initially consisted of four drums rotating on the same axis, which provided more than a million variants of the cipher of simple replacement. On each side of the drum, there were 25 electrical contacts in a circle (by the number of letters). The contacts on both sides of the drum were connected in pairs in a random manner by 25 wires, forming a substitution of symbols. The wheels were folded together and their contacts, touching each other, ensured the passage of electrical impulses through the entire set of wheels. Before starting work, the reels were rotated so that a given code word was set. When a key was pressed and the next character was encoded, the right drum rotated one step. After the drum made a full turn, the next drum was turned one step. Thus, the resulting key was obviously much longer than the message text.

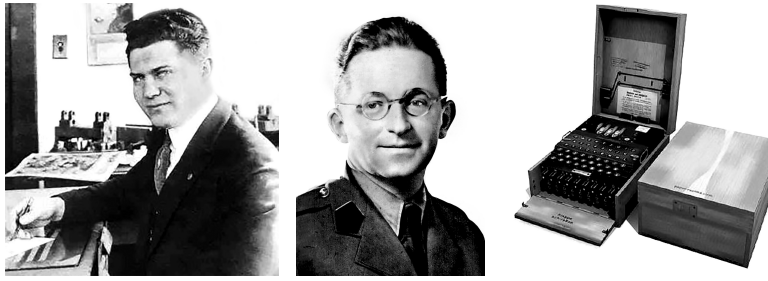


Figure 1.10 – Edward Hoog Hebern (1869-1952), Arthur Schrebius (1878-1929) and the Enigma rotary cipher machine

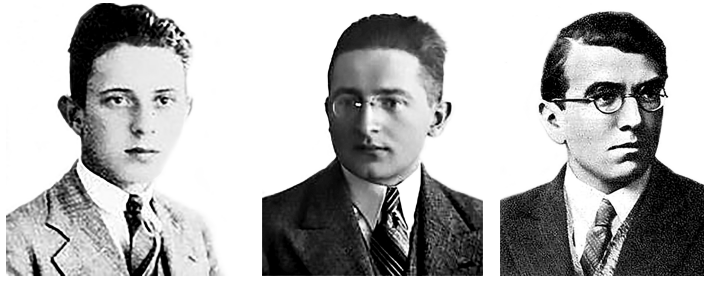


Figure 1.11 – Jerzy Ruzycki (1909-1942), Marian Adam Rejewski (1905-1980), Henrik Zygalski (1908-1978)

The Enigma cipher was deciphered for the first time by the Polish Bureau of Ciphers in December 1932. Marian Rejewski, Jerzy Ruzicki, Heinrich Zygalski and Marian Rejewski, with the help of French intelligence data,

mathematical theory and reverse engineering methods, were able to develop a special device for decrypting encoded messages, which was called a cryptological bomb. After that, German engineers complicated the Enigma device and in 1938 released an updated version, which required building more complex mechanisms to decipher.

During the Second World War in England to decrypt Enigma messages, a machine codenamed «Turing Bombe» was created which provided significant assistance to the anti-Hitler coalition.

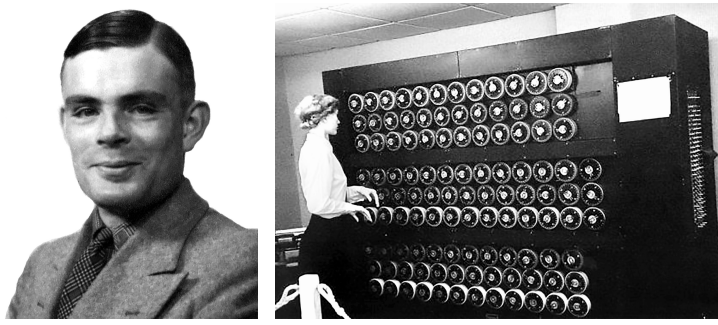


Figure 1.12 – Alan Turing (1912-1954) and the electronic computer «Turings Bomb»

Rotary machines were actively used during the Second World War and other states. In addition to the German vehicle, the Sigaba (USA), Tureh (UK), 91-shiki ohbun-injiki (Japan) and many others were also used. Rotary systems became the pinnacle of formal cryptography, since they were relatively easy to implement sufficiently cryptographically strong ciphers.

Figure 1.13 – American rotary encryption machines SIGABA (ECM MARK II) and its successor KL-7[9]

[9] The KL-7 remained in service until the 1970s. In some countries, KL-7s served as backup devices for many years until they were finally withdrawn from service in 1983.

With foreign machines that worked on the principle of mechanically programmable disk encoders, the technology of the USSR had little in common. The cipher machine, developed in 1934, the M-100 «Spectrum» and its descendants worked on the principle of imposing a gamma on a plain text. It allowed encryption at up to 300 characters per minute. Only the K-37 «Crystal» encryption machine, developed in the USSR in 1939, was built by analogy with other rotary machines.

At the same time, no cases of decryption of messages encrypted by M-100 and M-101 were recorded[10]. The messages encrypted by the K-37, the United States on a regular basis «read» from April 1946 to 1947, automating this process using a machine analogue Sauterne Mark I.

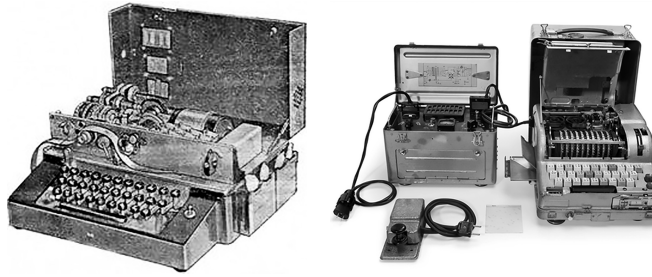


Figure 1.14 – Soviet encryption machine M-100 «Spectrum» and its descendant – coding machine «Fialka-125»

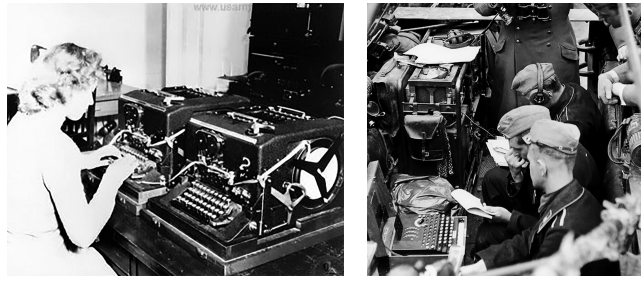


Figure 1.15 – The work of the encryptor in the United States on the rotary Sigaba machine and the Wehrmacht encryptors on the Enigma machine in the trenches.

[10] It is known that in the period from 1941 to 1947, on the basis of the Ufa GSPEI 56 and a number of other factories, a total of 2,024 speech encoders were produced. During the war years, the eighth (encryption) directorate of the CCCH General Staff sent out about 3.2 million cipher suites. A total of over one and a half million encrypted telegrams and codograms were transmitted. As the author of several books on cryptography Dmitry Larin writes, «the load on communication channels sometimes reached 1500 telegrams per day».

1.4 Scientific Cryptography

After the First World War, the governments of almost all the leading countries classified all work in the field of cryptography. By the early 1930s, the branches of mathematics were finally formed, which are the basis for future science: general algebra, number theory, probability theory and mathematical statistics. By the end of the 1940s, the first programmable calculating machines were built, the foundations of the theory of algorithms and cybernetics were laid. Nevertheless, in the period after the First World War and until the end of the 1940s, a minimum of works and monographs were published in the open press, but even they did not reflect the most current state of affairs. The greatest advances in cryptography was made in the military.

A key milestone in the development of scientific cryptography is Claude Shannon's fundamental work Communication Theory of Secrecy Systems, a secret report presented by the author in 1945 and published by him in the Bell System Technical Journal. in 1949. In this work, in the opinion of many modern cryptographers, the approach to cryptography in general as a mathematical science was first shown.

Claude Shannon introduced the concepts of «dispersion» and «mixing», substantiated the possibility of creating almost arbitrarily strong cryptosystems.

Shannon proved that the encryption method proposed by Vernam in 1917 is an absolutely strong encryption system. Naturally, provided that the key length is equal to or greater than the message length.

Figure 1.16 – Electronic encryption machine Aroflex (USA) and portable electronic cipher machine HG-530/535 of the CRIPTOMATIC 500 family of Boris Hagelin (Switzerland)

This period is also characterized by the gradual decommissioning of electromechanical rotary encryption machines by government and army

structures and the transition to electronic encryption machines such as KW-26, KW-37, KL-51 (RACE) and Aroflex in the USA and NATO countries.

In the 1960s, various block ciphers began to appear that were more cryptographic than the result of rotary machines. However, they assumed the mandatory use of digital electronic devices – manual or semi-mechanical encryption methods were no longer used.

1.5 Computer Cryptography

Almost all modern cryptosystems used are reasonably strong, that is, the strength of this cryptosystem today is estimated by the amount of computation required to open it. It is believed that an encryption key is strong enough if all known methods of finding it are so complex that they take more time than a simple search of all possible keys. And the period of its finding exceeds the lifetime of the protected information (or the costs are greater than the cost of obtaining this information in other ways).

Block ciphers became the first class of computer cryptosystems, the practical application of which became possible with the advent of compact computing facilities.

In the 70s of the twentieth century, a lot of work was done on the standardization of ciphers. Probably one of the first in this direction were US cryptographers. INTEL employees developed an algorithm that later became the American DES encryption standard. One of its authors, Horst Feistel, described a model of block ciphers, on the basis of which other, more secure symmetric cryptosystems were later built, including the Soviet and Russian encryption standard GOST 28147–89 and the modern AES standard.

With the advent of DES, cryptanalysis was also enriched; for attacks on the American algorithm, several new types of cryptanalysis (linear, differential, etc.) were created, the practical implementation of which, again, was possible only with the advent of powerful computing systems.

It should also be especially noted that in the mid-70s of the XX century, there was a real breakthrough in modern cryptography – the emergence of cryptosystems with two keys – secret and public. With their appearance, the problem of distribution of encryption keys has become less significant. Such systems are also called *asymmetric cryptosystems*.

The starting point for asymmetric cryptography is considered to be the work published by Whitfield Diffie and Martin Hellman in 1976 entitled

«New Directions in Modern Cryptography.» It was the first to formulate the principles of exchanging encrypted information without exchanging a secret key. Ralph Merkley approached the idea of asymmetric cryptosystems independently.

A few years later, Ron Rivest, Adi Shamir, and Leonard Adleman developed the RSA algorithm, the first practical asymmetric cryptosystem whose robustness was based on the problem of factorizing large primes. Asymmetric cryptography opened several new applied areas at once, in particular, *electronic digital signature (EDS) systems and electronic money.*

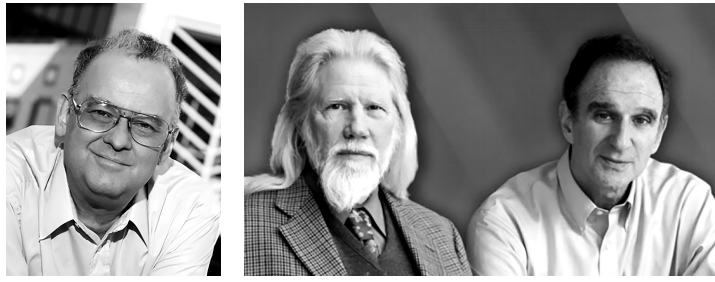


Figure 1.17 – Ralph Merkley and Wilfrid Diffie with Martin Hellman after the 2015 Turing Award for Fundamental Contribution to Cryptography

Figure 1.18 – RSA Algorithm Developers Ron Rivest, Adi Shamir, and Leonard Adelman[11]

[11] Interesting fact: the RSA cryptographic algorithm, created by Ronald Rivest, Adi Shamir and Leonard Adleman, as already noted, is based on the work of Whitfield Diffie and Martin Hellman «New Directions in Cryptography». At the same time, Rivest, Shamir and Aldeman received the Turing Prize for the RSA algorithm in 2002, and Diffie and Hellman only in 2015.

The urgent task of this period is the task of improving symmetric cryptosystems. In the same period, non-Faystel ciphers were already developed (SAFER, RC6, etc.), and in 2000, after an open international competition, a new US national encryption standard – AES was adopted.

1.6 Unknown Cryptography

Despite the fantastic advances in cryptography and cryptanalysis based on the use of computers and artificial intelligence systems, mankind knows a number of artifacts that break all axioms and seemingly successfully proven hypotheses.

In our opinion, the story of the achievement of cryptography and cryptanalysis would be incomplete without information about the mysterious mysteries of the past and unencrypted cryptograms.

Probably the most famous of these mysteries is the Voynich Manuscript, an illustrated manuscript dated to the 15th century and named after the Polish-Lithuanian bibliophile and antiquarian Mikhail Leonardovich Voynich. He bought the unusual 240-page book at the Villa Mondragone near Rome in 1912 during a secret sale of the Jesuit college library archives.

The book is written in an unknown language with unknown characters (the alphabet contains more than 50 unique characters) and contains diagrams and drawings of unknown animals and plants. Radiocarbon analysis at the University of Arizona suggests that the manuscript was written on parchment between 1404 and 1438, while analysis of its ink by the McCrone Research Institute in Chicago confirmed the date. Semantic and frequency analysis allows us to say that the language in which the manuscript is written has similar characteristics to Latin and European languages. Repeatedly conducted research suggests that the book contains well-thought information.

The general impression that the pages of the manuscript create allows researchers to assume that it was intended to serve as a pharmacological or medical reference book or encyclopedia. However, the confusing details of the illustrations feed many theories about the book's origins, the content of the text, and the purpose for which it was written.

It is safe to say that the first part of the book is about herbs, but attempts to compare them with real samples of herbs and with stylized drawings of herbs of that time have generally failed[12]. The remaining sections are

[12] By 2014, scientists were able to presumably identify 37 of the 303 plants depicted in the manuscript.

conventionally called astronomical, biological, cosmological, pharmaceutical and prescription.

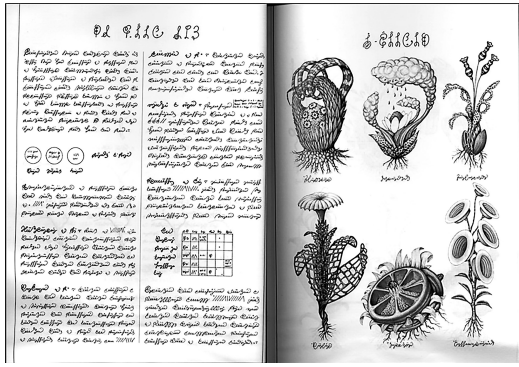


Figure 1.19 – The Voynich manuscript page and Wilfred Voynich himself (1865-1930)

There are many hypotheses about the origin of the manuscript – from the fact that the book was written in Ukrainian as a reference to alchemy, to the fact that the book is a copy of a document from another world.

Almost the entire twentieth century, scientists of various directions (from linguists and historians to physicists and cryptanalysts) tried to solve the riddle of the manuscript. Most researchers assume that it is some kind of encryption, and today dozens of scientific groups are trying to decipher it using both artificial intelligence systems and methods of historical sciences[13].

Since 1969, the manuscript has been kept in the Beinecke Rare Book Library at Yale University. The book is completely digitalized, so anyone can try to decipher the mysterious graphics and letters.

Less known than the Voynich manuscript, but no less mysterious are documents such as the Rohontsi Codex and the Magical Tablets from Kassel. The Rohoni Code is a «pocket-sized» book containing 448 pages, covered with some symbols. The number of unique characters used in the Codex is about ten times more than in any known alphabet. In some places

[13] As of January 2018, scientists from the University of Alberta in Canada, using artificial neural networks, managed to decipher one of the phrases in the book. The researchers found that the author of the manuscript changed the order of the letters in each word for his cipher and dropped the vowels. The phrase sounds like “She gave advice to the priest, the owner of the house, me and people,” although this translation is now disputed.

on the pages there are illustrations containing not only religious, but also quite everyday subjects.

Examination of the Rohonzi Codex paper showed that it was most likely made in Venice in the early 16th century. So far, no one has been able to decipher the Code, there are only some versions.

Figure 1.20 – A couple of pages of the Rohontsi Code

The next class of unsolved ancient cipher programs is the Phaistos disc, linear writing and rongo-rongo.

In 1908, the Italian archaeologist Luigi Pernier discovered a small clay disc while excavating at the site of the ancient Cretan city of Festus. The disc contains 242 characters. Experts were able to distinguish 45 types of symbols, but only a few of them were identified as hieroglyphs that were used in the pre-palace period of the ancient history of Crete.

At the moment, despite many attempts, no one has managed to unravel the mystery of the Phaistos disc[14].

Figure 1.21 – Phaistos disc and linear writing

[14] Some scholars believe that the Phaistos disc is an astronomical calendar, others believe that it hails from the legendary sunken city of Atlantis.

Linear[15] was also found in Crete and named after the British archaeologist Arthur Evans. In 1952, Michael Ventris partially decoded the Linear script that was used to encrypt the Mycenaean language.

Rongo-rongo is a system of mysterious records that was discovered on Easter Island in the XIX century. Rongo-rongo is believed to represent a lost prototype system. Numerous attempts at deciphering Rongo-rongo have been unsuccessful. Perhaps this would give an answer to the main mystery of the island – the purpose of the giant statues of Easter Island.

The nodular letter of the Incas is also mysterious and only fragmentarily deciphered[16].

There are a number of legends about the huge gold treasures of the «sons of the Sun», which the Incas hid from the Spanish conquistadors[17], and information about their whereabouts was preserved in the intercepted pile of the Incas.

Also, the inscription Shagboro can be attributed to the ancient undeciphered cryptograms. Excites cryptographers and linguists, intercepted from space radio signal, dubbed «Wow». Criminal cases have not yet been disclosed, where cryptograms were found with the bodies of people who died by violent death – the «Taman Shud» case and Ricky McCormick's notes.

---

[15] Used in Ancient Greece in the XIX - XV centuries BC.

16 In ancient times, in many regions of the Earth, the so-called nodular letter flourished, which, according to legend, was brought to Earth by the white Gods. Nodular writing was widespread among different peoples: in the Inca and Maya states, in China, Australia, Tibet, California, West and Central Africa, on the Ryukyu Islands, Palau, Hainan.

Chinese nodular writing is mentioned in the treatise Tao Te Ching («Book of the Way and Dignity»), written by the ancient Chinese philosopher Lao Tzu in the 6th-5th centuries. BC. Tied cords act as a carrier of information, and the information itself is carried by the knots and colors of the laces. Also, the nodular letter or Nodular Elm is a reflection of the Slavic alphabet «Glagolitic».

[17] For example, in 1533 a detachment of conquistadors led by Franco Pizarro invaded the empire of the Great Inca. Capturing one of the heirs of the Great Inca - Atahualpu - Pizarro demanded a ransom - to fill the huge room with gold to the level of a raised hand. Gold was brought from all over the Inca empire, but did not manage to reach the agreed height before the end of the established period. The Inca asked to wait a little longer, but on the evening of August 26, 1553, Atahualpa was hanged in a square in Cuzco. Before his death, Inca gave the kippah to the faithful people. Thirteen knots were tied on it, besides them, Atahualpa tied a bar of gold to a cord. It is believed that it was in this message that the order was encrypted to hide the treasures of the Incas in some secret place. On the same day, all the treasures disappeared from the temples, including a 350-paced gold chain with links as thick as an arm, weighing so much that only 200 people could lift it. Also missing were eleven thousand lamas, laden with gold and on their way to the capital of the Inca empire with ransom for Atahualpa.

Unfortunately, the format of the tutorial does not even cover the most famous of these cryptographic mysteries. History leaves future cryptanalysts with many interesting tasks, the answers to which will reveal the mystery of the mysterious events of the past.

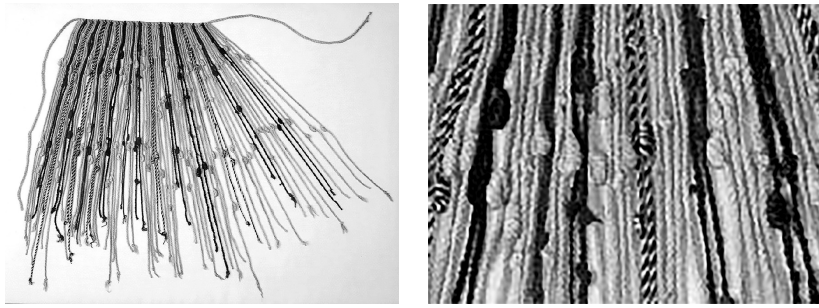


Figure 1.22 – Bale of the Incas and its enlarged fragment

Questions for Self-Control

- What historical periods can the history of cryptography be divided into?
- What device is considered to be the first European cryptography device?
- What encryption algorithm did Guy Julius Caesar suggest?
- What is «atbash»?
- Who proposed the principles of polygram substitution?
- What is a «black office»?
- On what principles is the Enigma encryption machine built?
- What was the M100 Kristall device used for?
- When were the principles of asymmetric cryptography established?
- What is the Voynich manuscript about?

Recommended Reading

1. Amirov A.Zh., Sultanova B.K., Shakhanov D.Zh. The history of the development of cryptology. Stages. – Young scientist. 2016. – No. 1.
2. Amirov A.Zh., Sultanova B.K., Shakhanov D.Zh. The history of the development of cryptology. Stages. – Young scientist. 2016.

3. Spivak S.I., Vildanov A.N., Zaripova L.I. Achievements and applications of modern informatics, mathematics and physics: materials of the III All-Russian scientific and practical correspondence conference (Neftekamsk, October 20-22, 2014). – Ufa: RIC BashGU, 2014.
4. Luciano D., Prichett G. Cryptology: From Caesar Ciphers to Public-Key Cryptosystems. – The College Mathematics Journal. – Mathematical Association of America, 1987. – Vol. 18, Iss. 1.
5. Singh S., Book of ciphers. The secret history of ciphers and their decryption. – M.: Astrel, 2007. – 448 p.
6. Soboleva T.A. The history of encryption in Russia. – M.: OLMA-PRESS Education, 2002.
7. Babash A.V., Shankin G.P. Cryptography (security aspects). - M.: SOLON-PRESS, 2007. – 512 p.
8. Chmora AL Modern applied cryptography. – M.: «Helios ARV», 2001.
9. Gardner M. A new kind of cipher that would take millions of years to break. – Mathematical Games, Scientific American, 1978. – 237(2).
10. Rivest R.L., Shamir A., Adleman L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. – Comm. ACM, 1978. – 21(2).

CHAPTER 2. SYMMETRICAL CRYPTOSYSTEMS

Keywords: symmetric cryptosystem, substitution, permutation, gamma, encryption algorithm, key, round, avalanche effect, run key, key schedule, Feistel network, cryptographic strength, block cipher, stream cipher.

2.1 Main Classes of Symmetrical Cryptosystems

Symmetric cryptosystems[1] is an encryption method in which the same cryptographic key is used for encryption and decryption. The algorithm key must be kept secret by both parties. The encryption algorithm is chosen by the parties before the start of the exchange of messages.

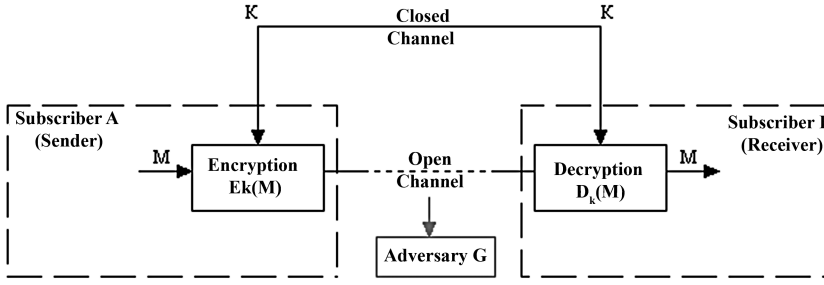


Figure 2.1 – General scheme of symmetric cryptosystems operation

Currently, modern symmetric ciphers are divided into block ciphers and stream ciphers:

– ***block ciphers*** – they process information in blocks of a certain length (usually 32, 64, 128 or 256 bits), applying a key to the block in a prescribed order, usually in several cycles of mixing and substitution, called rounds. The result of the repetition of rounds is an avalanche effect – an increasing loss of bit correspondence between blocks of open and encrypted data;

– ***stream ciphers*** – encryption is carried out over each bit or byte of the original (plain) text using gamma.

[1] Such systems can also be called «symmetric encryption», «symmetric ciphers» or in English «symmetric-key algorithm».

Symmetric cryptosystems are based on three basic operations for transforming a message or plain text:

– substitutions;
– permutations;
– gamming.

Most modern symmetric ciphers use a complex combination of many substitutions and permutations. Many such ciphers are executed in several (sometimes up to 80) passes, using a «pass key» on each pass. The set of «pass keys» for all passes is called a «key schedule». As a rule, it is created from a common key by performing certain operations on it, including permutations and substitutions.

Typical methods for constructing symmetric encryption algorithms are a substitution-permutation network[2] and a Feistel network.

A substitution-permutation network-based cipher receives a block and a key as input and performs several alternating rounds, consisting of alternating substitution stages and permutation stages.

One S-box is sufficient to achieve security, but such a box will require a lot of memory. Therefore, small S-boxes mixed with P-boxes are used.

The non-linear substitution stage mixes the key bits with the plaintext bits, creating Shannon's confusion. The linear stage of permutation distributes redundancy throughout the data structure, creating diffusion.

An S-box (English substitution box or S-box) replaces a small block of input bits with another block of output bits. This substitution must be one-to-one to ensure reversibility. The purpose of the S-box is a non-linear transformation, which prevents linear cryptanalysis. One of the properties of the S-box is the avalanche effect, that is, changing one bit at the input leads to a change in all bits at the output.

P-box (English permutation box or P-box) – permutation of all bits: the block receives the output of the S-box as input, swaps all the bits and gives the result to the S-box of the next round. An important quality of the P-box is the ability to distribute the output of one S-box among the inputs of as large S-boxes as possible.

Each round uses its own key derived from the initial one. Such a key is called a *round* key. It can be obtained both by dividing the original key into equal parts, or by some transformation of the entire key.

In the *Feistel network*, the encryption algorithm builds an encryption scheme based on the function F (D, K), where D is a piece of data half

[2] Permutation-permutation net or SP-net is also the development of Horst Feistel.

the size of the encryption block, and K is the «pass key» for this pass. A function is not required to be reversible – its inverse function may not be known. The advantages of the Feistel network are that the decryption and encryption almost completely coincide (the only difference is the reverse order of the «pass keys» in the schedule), which greatly facilitates the hardware implementation.

The permutation operation shuffles the bits of the message according to a certain law. In hardware implementations, it is trivially implemented as wire entanglement. It is the permutation operations that make it possible to achieve the «avalanche effect» 3. The permutation operation is linear – f (a) xor f (b) == f (a xor b).

Substitution operations are performed as replacing the value of some part of the message (often 4, 6, or 8 bits) with a standard, hard-coded other number in the algorithm by referring to a constant array. The substitution operation introduces non-linearity into the algorithm.

The complete loss of all statistical patterns of the original message is an important requirement for a symmetric cipher. As noted earlier, for this, the cipher must have an «avalanche effect»[3].

Another important requirement is the lack of linearity (that is, the condition f (a) xor f (b) == f (a xor b)).

Often the strength of an algorithm, especially against differential cryptanalysis, depends on the choice of values in lookup tables (S-boxes). At a minimum, it is considered undesirable to have fixed elements S (x) = x, as well as the lack of influence of some bit of the input byte on some bit of the result – that is, cases when the result bit is the same for all pairs of input words that differ only in this beat.

---

[3] The Avalanche effect is a cryptographic concept commonly applied to block ciphers and cryptographic hash functions. An important cryptographic property for encryption, which means that changing the value of a small number of bits in the input text or in the key leads to an «avalanche» change in the values of the output ciphertext bits. In other words, it is the dependence of all output bits on each input bit.

The term «avalanche effect» was first coined by Feistel in an article on Cryptography and Computer Privacy, published in Scientific American in May 1973, although the concept was used by Shannon.

An example of an avalanche effect for the Ek (K, M) encryption algorithm:

– Ek (key = «aaaa», plaintext = «aaaa») = «5188»;
– Ek (key = «aaaa», plaintext = «aaca») = «f7e5».

Currently, there are many (at least at least two dozen) used symmetric cipher algorithms, the essential parameters of which are:

- cryptographic strength;
- key length;
- number of rounds;
- the length of the processed block;
- the complexity of the hardware / software implementation;
- the complexity of the transformation.

The advantages of symmetric cryptosystems are:

- speed of encryption / decryption;
- ease of implementation (due to simpler operations);
- smaller required key length for comparable strength (as compared to systems with a public key);
- better knowledge (due to the greater age), again compared to systems with a public key.

The disadvantages of symmetric cryptosystems are:

- the complexity of key management in a large network;
- the complexity of the key exchange. To apply the algorithms, it is necessary to solve the problem of reliable key transfer to each subscriber, since a secret channel is needed to transfer each key to both parties.

To compensate for the shortcomings of symmetric encryption, a combined (hybrid) cryptographic scheme is now widely used, where using asymmetric encryption, a session key is transmitted that is used by the parties to exchange data using symmetric encryption.

An important disadvantage of symmetric ciphers is the impossibility of using them in mechanisms for generating electronic digital signatures and certificates, since the key is known to each party.

2.2 Substitutions, Permutations and Gamming

2.2.1 Encryption Using Permutations

A permutation cipher is a symmetric encryption method in which the characters of the encrypted plaintext are reversed. As an element of the encrypted text, as a rule, one character is chosen, however, larger constructions are also used – groups of characters.

Annograms[4] are a classic example of permutations. Historically, permutations are among the earliest ancient methods of cryptography. It is not known when the first permutations appeared. Perhaps the ancient scribes used anagrams or permutations of letters in the name of their king in order to hide his real name or for ritual purposes.

The first devices and algorithms operating on the principles of permutations have been known since the 5th century BC. These include, for example, the same script.

Permutations fall into two broad classes in classical cryptography:

– single (simple) permutation ciphers – during encryption, the plaintext characters are moved from their original positions to new ones once;
– multiple (complex) permutation ciphers – during encryption, the plaintext characters are moved from their original positions to new ones several times.

Simple permutation ciphers have found their use, as a rule, only in manual encryption methods. They often use tables that provide simple encryption procedures for rearranging letters in a message. The key in them is the size of the table, a phrase specifying a permutation or a special feature of the table.

Simple keyless permutation is one of the simplest encryption methods, akin to the cipher cipher. For encryption, the original plaintext is written to a table, for example, column by column. Reading to receive the ciphertext is performed line by line.

---

[4] Anagram (Greek ανα – "again" and γράμμα - "record") is a literary device consisting in rearranging letters or sounds of a word (or phrase), which results in another word or phrase. For example: an orange is a spaniel, a colonel is a bug, a vodka is a bagel, a petal is a telescope. The ancestor of the anagram is considered to be the ancient Greek poet and grammar Lycophron, who lived in the 3rd century BC. e. According to the surviving records of the Byzantine John Tsetz, from the name of Tsar Ptolemy Lycophron composed the first of the known anagrams: Ptolemaios - Aro Melitos, which means «from honey», and from the name of Queen Arsinoe: Arsinoe - Ion Eras («Hera's violet»).
In the XVII - XIX centuries. it was customary among natural scientists to encrypt their discoveries in the form of anagrams, which served two purposes: to hide the hypothesis until its final verification and to approve the authorship of the discovery when it is confirmed. For example, in 1610, Galileo Galilei encrypted the Latin phrase «Altissimun planetam tergeminum observavi» («Observed the highest triple planet») to secure the authorship for the discovery of the satellites of Saturn as follows: «Smaismrmilmepo etaleumibunenugttauiras» (letters «v» and «u» in Latin texts were often considered interchangeable). Further development and popularization of anagrams was associated with the development of Christianity and Latin.

For example, the message «НЕЯСНОЕ СТАНОВИТСЯ ЕЩЕ БОЛЕЕ НЕПОНЯТНЫМ» ("UNCLEAR BECOMES EVEN MORE UNCLEARABLE») is written to the table column by column. For a table with 5 rows and 7 columns, it looks like this:

Table 2.1 – Permutation table

Н	**О**	**Н**	**С**	**Б**	**Н**	**Я**
Е	**Е**	**О**	**Я**	**О**	**Е**	**Т**
Я	**С**	**В**	**Е**	**Л**	**П**	**Н**
С	**Т**	**И**	**Щ**	**Е**	**О**	**Ы**
Н	**А**	**Т**	**Е**	**Е**	**Н**	**М**

After the plaintext is written in columns, it is read line by line to form an encryption. If you write it down in groups of 5 letters, you get: «НОНСБ НЯЕЕО ЯОЕТЯ СВЕЛП НСТИЩ ЕОЫНА ТЕЕНМ».

The key is the size of the table and the write / read algorithms. Combining letters into groups is not included in the cipher key and is used only for the convenience of writing meaningless text.

A more practical encryption method called single key permutation is very similar to the previous one. It differs only in that the columns of the table are rearranged by a keyword, phrase or a set of numbers as long as a table row.

For example, using the word «ЛУНАТИК» as a key, we get the following table:

Table 2.2 – Results of a single permutation

Before permutation

Л	**У**	**Н**	**А**	**Т**	**И**	**К**
4	7	5	1	6	2	3
Н	О	Н	С	Б	Н	Я
Е	Е	О	Я	О	Е	Т
Я	С	В	Е	Л	П	Н
С	Т	И	Щ	Е	О	Ы
Н	А	Т	Е	Е	Н	М

After permutation

А	И	К	Л	Н	Т	У
1	**2**	**3**	**4**	**5**	**6**	**7**
С	Н	Я	Н	Н	Б	О
Я	Е	Т	Е	О	О	Е
Е	П	Н	Я	В	Л	С
Щ	О	Ы	С	И	Е	Т
Е	Н	М	Н	Т	Е	А

Its top line contains the key, and the numbers under

the key are determined by the natural order of the corresponding letters of the key in the alphabet. If identical letters were found in the key, they would be numbered from left to right.

It turns out the cipher: «СНЯНН БОЯЕТ ЕООЕЕ ПНЯВЛ СЩОЫС ИЕТЕН МНТЕА».

The route permutation method has also become widespread. In them, the key is a certain geometric figure. The transformation is carried out due to the fact that writing the text goes along one path, and reading – along another. Again, the most famous example is the scital.

One of the ways of route permutation is called «intersection». The example below draws enough cruciform shapes to contain all the letters of the message. The plain text is written around these figures in a predetermined way – in our case, clockwise. Letters are taken line by line. First, the specified number of letters (N) from the first line is taken, then the doubled number of letters (2N) from the second and again N letters from the third line.

For example, the message «COMING SIXTH» might look like this:

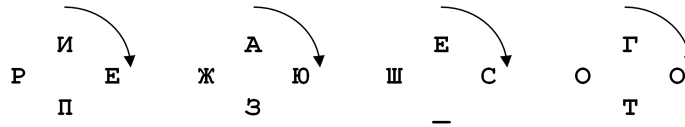


Figure 2.2 – An example of placing plaintext in the «Crossroads» cipher

For example, for N = 2, the ciphergram will look like «ИА RECHYU ПЗ ЕГ ШСОО _ Т»[5].

Also, for route permutations, other geometric shapes can be used, for example, triangles and trapezoids. Plain text fits into these shapes according to the word count and shape of the selected shape, which can be stretched or shrunk to fit the message.

For the first shape, the triangle, the plaintext is written line by line from top to bottom. The keyword is written below.

[5] Spaces in the ciphertext are inserted to illustrate how the algorithm works and when receiving the ciphertext. Naturally, they will not be present in real encryption.

If the base of the triangle is wide and longer than the length of the keyword, then the keyword is repeated. The letters of the keyword string are numbered sequentially according to their alphabetical order. The encrypted message is written out in columns according to the numbering performed.

Open text				П			
			Р	И	Е		
		З	Ж	А	Ю	_	
	Ш	Е	С	Т	О	Г	О
Key	Л	У	Н	А	Т	И	К
Column permutation algorithm	4	7	5	1	6	2	3

Figure 2.3 – An example of using the permutation cipher when fitting into a triangle

For example, for the plain text «COME _ THE SIXTH» and the keyword «ЛУНАТИК» the cipher obtained from the triangle will look like «ПИАТ _ ГОШЗЕЕЮОЗЕ».

In 1550 the Italian mathematician Gerolamo Cardano proposed a new encryption technique – the Cardano lattice.

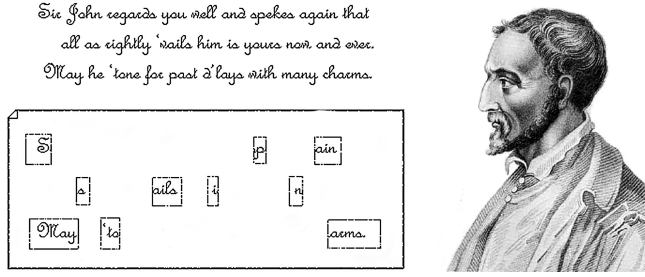


Figure 2.4 – Cardano Lattice6 and its creator, Gerolamo Cardano[6] (1501-1576)

Initially, the Cardano grille was a stencil with holes cut in it. The letters, syllables and words of the message were written in these holes on a sheet of paper that was placed under the lattice. Then the stencil was removed, and

[6] Note text: "Sir John highly values you and repeats again that everything that is available to him is now yours, forever. Can he deserve forgiveness for his past delays through his charm.» Encrypted message: «Spain will send its ships to war in May.»

the free space was filled with more or less meaningful text to mask the secret message. This method of hiding information belongs to steganography.

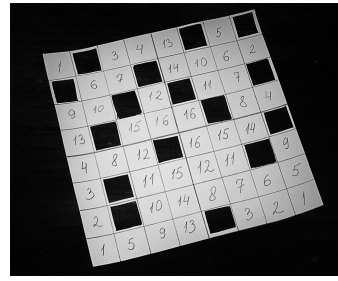


Figure 2.5 – Rotating lattice

Later, the cipher «rotary lattice» was proposed, or, as it is also called, «trellis for climbing plants», since it resembled holes in the wooden trellises of garden buildings. This cipher is considered the first transpositional (geometric) cipher. Naturally, it is more convenient and easier to use a square, not a rectangle, for a rotary lattice. An example of a square lattice is shown in Figure 2.5.

Although there is a big difference between Cardano's original proposal and the pivoting grid cipher, stencil-based information hiding methods are commonly referred to as Cardano grids[7].

For encryption and decryption using this cipher, a rectangular stencil with an even number of rows and columns is made. Cells are cut in the stencil in such a way that when it is applied to a table of the same size in four possible ways, its cutouts completely cover all the table cells exactly once.

During encryption, the stencil is applied to the table. The letters of the original text are written out in the visible cells of the table from left to

[7] It is known that Cardinal Richelieu was an adherent of the Cardano grid and actively used it in personal and business correspondence.

Also, the encryption method based on the rotary lattice was used by the Dutch rulers for secret messages in the 1740s. It was also used by Kaiser Wilhelm's army in World War I. For encryption, the Germans used gratings of various sizes, which were given their own code names by French cryptanalysts: Anna (25 letters), Bertha (36 letters), Dora (64 letters) and Emile (81 letters). However, the grilles were used for a very short time (only four months), to the great disappointment of the French, who had just begun to pick up keys for them.

right from top to bottom. Then the stencil is rotated and the next part of the letters fits in. This operation is repeated two more times. The cipher program is written out from the summary table along a certain route.

Thus, the key for encryption is the stencil, the order of its turns and the route of writing.

Another kind of permutation is magic squares. Magic squares are square tables with sequential natural numbers inscribed in their cells, starting from 1, which in the sum for each column, each row and each diagonal give the same number.

For the first time, these squares appeared in China, where some «magical power» was attributed to them[8].

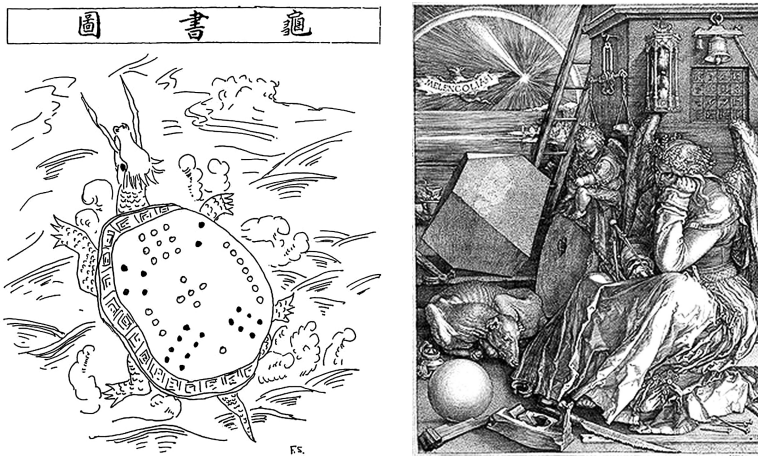


Figure 2.6 – The magic square of Lo Shu and the magic square on the engraving by Albert Durer «Melancholy»

[8] According to the legend, described in one of the five canonical books of Ancient China - Shu-Jing (Book of written traditions), in 2200 BC. a huge turtle (according to another version – a dragon) emerged from the Lo River, a symbol of eternity. On her carapace, spots were visible, forming an amazing pattern.

When the turtle came out of the water, the puddles were drying up after a recent rainstorm. Great Yu took this turtle and examined the strange pattern on its shell. This pattern inspired him to create a treatise called «Hong Fan» («The Great Plan»), which spoke about physics, astrology, divination, morality, politics and religion.

Magic squares were widely used to convey classified information. When encrypting, the original message was inscribed in a square according to the numbering given in them, after which the ciphergram was written out line by line. The number of possible magic squares (keys) increases rapidly with their size. So, there is only one 3x3 magic square, if you do not take into account its rotations. There are already 880 magic 4x4 squares, and the number of 5x5 magic squares is about 250,000. Therefore, large magic squares could be a good basis for a reliable encryption system of that time, because manual enumeration of all key variants for this cipher was practically impossible.

For example, consider a 4x4 square. It fits numbers from 1 to 16. Its magic lies in the fact that the sum of numbers in rows, columns and full diagonals is equal to the same number – 34.

Magic square encryption was performed as follows. The letters of this phrase are inscribed sequentially in a square according to the numbers written in them: the position of the letter in the sentence corresponds to the ordinal number. A period or any letter is placed in empty cells.

For example, you need to encrypt the phrase: «ПРИЕЗЖАЮ _ ШЕСТОГО» («I ARRIVE _ ON THE SIXTH») using a magic square 4x4.

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

Figure 2.7 – One of the possible 880 options – 4x4 magic square

16 О	3 И	2 Р	13 Т
5 З	10 Ш	11 Е	8 Ю
9 _	6 Ж	7 А	12 С
4 Е	15 Г	14 О	1 П

Figure 2.8 – An example of encryption using a magic square

The encrypted text is written to a line (reading is performed from left to right, top to bottom, line by line) – «ОЗРТЗШЕЮ _ ЖАСЕГОП».

Complex permutation ciphers are based on the idea of re-encryption by methods of permutations already encrypted from the text, that is, multiple permutations.

The most famous complex permutation method is the double permutation method. In this case, permutations are determined separately for columns and separately for rows. The plain text fits into the table. After that, the columns are rearranged, and then the rows. When decoding, the order of permutations is reversed.

For example, let's encrypt the message «ПРИЕЗЖАЮ ШЕСТОГО». Key «2413» and «4123».

Table 2.3 – Results of double permutation

Initial Table

	2	4	1	3
4	П	Р	И	Е
1	З	Ж	А	Ю
2		Ш	Е	С
3	Т	О	Г	О

Column permutation

	1	2	3	4
4	И	П	Е	Р
1	А	З	Ю	Ж
2	Е		С	Ш
3	Г	Т	О	О

Row permutation

	1	2	3	4
4	А	З	Ю	Ж
1	Е		С	Ш
2	Г	Т	О	О
3	И	П	Е	Р

It turns out the encryption «АЗЮЖЕ СШГТООИПЕР».

The number of options for double permutation is also great: for a 3x3 table there are 36, for 4x4 there are 576, and for 5x5 there are already 14,400. However, a double permutation is a very weak type of cipher, easily readable for any size of the encryption table.

2.2.2 Encryption Using Substitutions

Substitution is a symmetric encryption method based on replacing characters in the original alphabet with other characters according to a specific rule. As with substitutions, a single letter, pair, or group of letters or numbers can be used as a plaintext character.

In classical cryptography, four types of substitution ciphers are distinguished:

– mono-alphabetic substitution cipher (simple substitution cipher) – a cipher in which each plaintext character is replaced by some character

of the same alphabet fixed for a given key. Examples: Julius Caesar's code, Polybius square, Atbash;
- a one-sound substitution cipher is similar to a one-alphabetic one, but in it the plaintext character can be replaced by one of several possible characters;
- a polygram substitution cipher replaces not one character, but an entire group. Examples: Playfer cipher, Hill cipher;
- a polyalphabetic substitution cipher consists of several simple substitution ciphers. Examples: Vigenère cipher, Beaufort cipher, one-time pad.

With multi-alphabet substitution, the transformation law changes from symbol to symbol.

Mono-alphabetic cipher[9] is a class of encryption methods that boil down to the creation of an encryption table according to a certain rule, in which for each plaintext letter there is a single ciphertext letter (symbol) associated with it. The encryption itself consists in replacing letters according to this table. The same table is used for decryption.

Simple replacement ciphers include many encryption methods that arose in antiquity or the Middle Ages, such as Atbash, Caesar's cipher, Polybius's square, the method of dancing men[10] and many others.

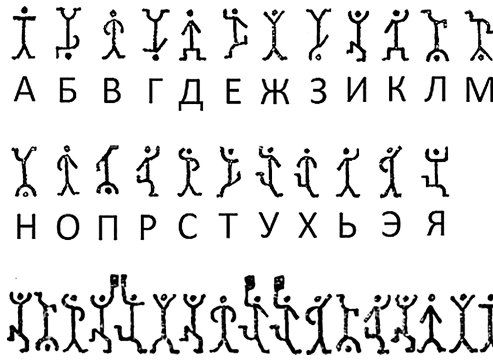


Figure 2.9 – Encryption table for the one-alphabetic cipher «Dancing Men» and one of the encryption codes from the work «The Return of Sherlock Holmes»

[9] Other common names are simple substitution ciphers, simple substitution ciphers, and mono-alphabetic ciphers.

[10] «Dancing Men» is one of 56 stories by the English writer Arthur Conan Doyle about the detective Sherlock Holmes and Dr. Watson, included by the writer in the collection of 13 stories «The Return of Sherlock Holmes».

The cryptographic strength of this encryption method is determined by the capacity of the alphabet used.

A cryptographic protection system based on a mono-alphabetic cipher is easily vulnerable. If the adversary has a cipher and corresponding source text, or cipher text of the adversary's chosen source text, then determining the key and decrypting the original text is a trivial task.

This cipher uses numbers that replace letters. There is no logic in these numbers. Such a simple cipher can be decrypted with a cipher table.

Nowadays, simple substitution ciphers are easily broken even if only the ciphertext is available. In any language, various letters and combinations of two, three or more letters have characteristic repetition rates in texts.

For example, for the Russian language (yes, like any other language) there are tables of character frequencies

Table 2.4 – Frequencies of characters in the Russian-language text

Letter	Frequency	Letter	Frequency	Letter	Frequency
а	0,075	к	0,034	ф	0,002
б	0,017	л	0,042	х	0,011
в	0,046	м	0,031	ц	0,005
г	0,016	и	0,065	ч	0,015
д	0,030	о	0,110	ш	0,007
е, ё	0,087	п	0,028	щ	0,004
ж	0,009	р	0,048	ь, ъ	0,017
	0,018	с	0,055	ы	0,019
и	0,075	т	0,065	э	0,003
и	0,012	у	0,025	ю	0,022
				я	0,022

For example, the frequencies of characters for a work of art, scientific article, or technical documentation written in the same language will be almost the same.

Accordingly, if we carry out a frequency analysis of the text encrypted by the simple replacement method, determine the frequency of the symbols, then it is not a problem to replace the cipher symbols with the corresponding letters of the language.

It should be noted that the simple substitution cipher does not always imply the replacement of a letter with some other letter. It is allowed to use

the replacement of a letter with a number. For example, in the square of Polybius, letters are replaced by a certain cipher alphabet.

Monosonic substitution ciphers are completely similar to mono-alphabetic ciphers, except for the fact that during the encryption process, the plaintext character can be replaced by one of several variants, each of which uniquely corresponds to the original one. A monosonic substitution cipher, in contrast to a mono-alphabetic cipher, cannot be broken using frequency cryptanalysis, since it masks the frequency characteristic of the text, although it does not hide all statistical properties.

Examples of such one-sound ciphers are the nomenclator, the great Rossignol cipher, and the book cipher.

Polygram ciphers are based on the fact that in order to increase the cryptographic strength of the cipher, characters are replaced not one character at a time, but several at once (the replacement is done with grams)[11]. Examples of such ciphers can be the Bigram Port and Playfair ciphers, Hill's cipher and others.

The Ports cipher is probably the first known bigram cipher. His algorithm was published in 1563 in the book of Giovanni Porta «On Secret Correspondence»[12].

Porta suggested using a square table with a periodically shifted mixed alphabet and password. He advised choosing a long key. For the first time, he proposed a simple bigram replacement cipher, in which pairs of letters were represented by one special graphic symbol. They filled out a 20x20 square table (it did not have the letters J, K, U, W, X and Z), the rows and columns of which were numbered with alphabet letters.

In principle, any numbers, letters or symbols could be written in the cells of the table – Giovanni Porta himself used symbols – provided that the contents of none of the cells were repeated.

Despite the fact that for this cipher Ports later began to be called the father of modern cryptography, at that time his system was not recognized

[11] It is believed that the «father» of bigram ciphers is the German abbot Johann Trysemus, who back in 1508 in his work «Polygraphy», first noted the possibility of encryption with bigrams, that is, two-letter combinations. Their resistance to opening turned out to be much higher than that of other predecessors, therefore, some bigram ciphers remained relevant until the Second World War.

[12] Porta can be said to have anticipated what is called the «probable word method» and provides examples of lists of probable words from various fields. In fact, this book was a textbook on cryptography, containing the cryptographic knowledge of the time.

by Italian cryptographers and did not find widespread use. The reason for this was the complexity of the encryption and the need to constantly have the entire cipher table with you.

	А	Б	В	Г	Д	Е (Ё)	Ж	З	И (Й)	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
А	001	002	003	004	005	006	007	008	009	010	011	012	013	014	015	016	017	018	019	020	021	022	023	024	025	026	027	028	029	030	031
Б	032	033	034	035	036	037	038	039	040	041	042	043	044	045	046	047	048	049	050	051	052	053	054	055	056	057	058	059	060	061	062
В	063	064	065	066	067	068	069	070	071	072	073	074	075	076	077	078	079	080	081	082	083	084	085	086	087	088	089	090	091	092	093
Г	094	095	096	097	098	099	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124
Д	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155
Е (Ё)	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186
Ж	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217
З	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248
И (Й)	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279
К	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310
Л	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341
М	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372
Н	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403
О	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434
П	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465
Р	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496
С	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527
Т	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558
У	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589
Ф	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620
Х	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651
Ц	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682
Ч	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713
Ш	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744
Щ	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775
Ъ	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806
Ы	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837
Ь	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868
Э	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899
Ю	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930
Я	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961

Figure 2.10 – An example of a table of cipher substitutions for the Ports cipher (Russian alphabet)

One of the most famous polygram ciphers is the Playfair cipher. This manual algorithm for bigram substitution was invented in 1854 by the English physicist Charles Wheatstone, but named after Lord Lyon Playfer.

Its first description was recorded in a document signed by Wheatstone on March 26, 1854[13].

Figure 2.11 – Charles Wheatstone (1802-1875) and Lord Lyon Playfer (1818-1898)

The Playfair bigram cipher was developed to encrypt messages with pairs of letters (bigrams).

The basis of this cipher is a table, the key is the number of rows and columns (table size) and a keyword.

The encryption process begins with the stage of preparing the plain text, which must meet the following requirements:

– have an even number of letters, in the case of an original message of odd length, an insignificant character (for example, a space or a period) must be added to the end of the message
– after splitting into pairs of letters, there should not be bigrams containing two identical letters. Two letters repeating in a row occur quite often in any language, so it is necessary to make sure that they are in different bigrams, for example, in the word «ДИАГРАММА» when dividing into bigrams, the fourth bigram consists of two identical letters «ММ» («ДИ АГ РА ММ А_»). The best

[13] The code was used for tactical purposes by the British military in the Second Boer War and in the First World War, and by the Australians and Germans during the Second World War. The reason for using the Playfer cipher was its sufficient ease of use and the absence of the need for additional special equipment. The main purpose of using this encryption system was to protect important but unclassified information during combat. By the time enemy cryptanalysts hacked the message, the information was already useless to them.

The use of the Playfair cipher is currently impractical because modern computers can easily break the cipher within a few seconds. The first published algorithm for breaking the Playfer cipher was described in 1914 by Joseph O. Mowborn.

way to correct this situation is to add a space at the beginning of the word. Then consecutive letters will fall into different bigrams: «_Д ИА ГР АМ МА».

At the final stage of encryption, the plain text is divided into pairs of letters, which are sequentially converted using the cipher table into ciphertext bigrams according to the following rules:

- if both letters of the bigram of the original text do not lie in the same line or in the same column, then the letters are found in the corners of the rectangle defined by the given pair of letters. The first letter of the bigram of the cipher-text becomes a letter located in the same line as the first letter of the original bigram, and in the same column as the second letter of the plaintext, the second letter of the bigram of the ciphertext is at the intersection of the line containing the second letter and the column containing the first letter of the plaintext;
- if both letters of the bigram of the plaintext belong to the same line of the table, then the first and second letters of the bigram of the ciphertext are the letters lying to the right, respectively, of the first and second letters of the bigram of the plaintext;
- if both letters of the plaintext bigram belong to one column of the table, then the first and second letters of the bigram of the ciphertext are the letters that lie, respectively, under the first and second letters of the plaintext bigram;
- it is considered that the table is cyclically closed by rows, that is, the end of any row is associated with its beginning, therefore, if the letters of the bigram are located in one line and one of them is in the last column of the table, then the letter from the first column of this row is taken for the ciphertext, or if is the end of any column is closed at its beginning. Therefore, if the letters of the bigram are located in one column and one of them is in the last row of the table, then the letter from the first row of this column is taken for the ciphertext.

For example, let us encrypt the message «DURING THE FIRST WORLD WAR, BIGRAMMY CODES WERE USED» with the Playfair bigram cipher.

At the stage of preparing the text, we take into account that in the original message 61 characters (odd number) and one of the bigrams (51 and 52 characters) contains the same letters «ММ».

To increase the number of characters in the message to an even number and separate the repeating letters by

different bigrams, add one space before the word «USED». Adding a space before the word «BIGRAMMY» would lead to a situation where there are two spaces in one bigram.

Dividing the text into bigrams, we get: «ВО», « _ В», «РЕ», «МЯ», « _ П», «ЕР», «ВО», «Й _ », «МИ», «РО», «ВО», «Й _ », «ВО», «ЙН», «Ы _ », « _ И», «СП», «ОЛ», «ЬЗ», «ОВ», «АЛ», «ИС», «Ь _ », «БИ», «ГР», «АМ», «МН», «ЫЕ», « _ Ш», «ИФ», «-РЫ».

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

Figure 2.12 – An example of the implementation of the Playfair cipher

For the first bigram «ВО» we use the first encryption rule. It is replaced with the «ГМ» ciphertext bigram. Further, according to the same rule, we replace « _ В» with «ЭЕ».

Similarly, using the algorithm, the rest of the bigrams are replaced.

As a result of encryption of the original message by the Playfair method using the « ЛУНАТИК « key, we will receive the following bigrams of the ciphertext:

«ГМ», «ЭЕ», «ЩР», «ПЭ», «ЯР», «РЩ», «ГМ», «Т.», «СУ», «СП», «ГМ», «Т.», «ГМ», «КА», «Щ,», «.Т», «ФР», «НЗ», «ЛХ», «МГ», «ТУ», «ЁЪ», «Э.», «ЁЛ», «ЕО», «ПУ», «ОУ», «ЩЖ», «ЯЩ», «КС», «ФЩ».

Hill's cipher[14] is the first all-polygram substitution cipher that allowed, in practice (albeit with difficulty), to simultaneously operate with more than three characters (grams). The cipher is based on linear algebra and modular arithmetic. Invented by the American mathematician Lester Hill in 1929.

When encrypting a letter, first, a number is matched. For the Latin alphabet, the simplest scheme is often used: A = 0, B = 1, ..., Z = 25. A block of n letters is considered as an n-dimensional vector and multiplied

[14] The Hill cipher was first described in the article «Cryptography in an Algebraic Alphabet» published in The American Mathematical Monthly in June-July 1929. In August of that year, Hill expanded on the topic and gave a speech on cryptography to the American Mathematical Society.

by an n × n matrix modulo 26[15]. The entire matrix is cipher key. The matrix must be invertible in order for the decryption operation to be possible.

For n = 3, the Hill cipher can be described in matrix form:

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} (\mathrm{mod}\, 26) \tag{2.1}$$

or

$$C = KP \pmod{26} \tag{2.2}$$

where P and C are column vectors of height 3, representing plain and cipher text, respectively;

K is a 3 × 3 matrix representing the encryption key. Operations are performed modulo 26.

In order to decrypt the message, you need to get the inverse matrix[16] of the key K^{-1}.

In order to decrypt the message, it is necessary to turn the ciphertext back into a vector and then simply multiply by the inverse key matrix:

$$P = K^{-1}C \pmod{26} = K^{-1}[KP \pmod{26}] \pmod{26} = P \bmod 26 \tag{2.3}$$

with P less than 26.

When working with two characters at a time, the Hill cipher does not provide any specific advantages over the Playfair cipher and is even inferior to it in terms of cryptographic strength and ease of calculations on paper. As the dimension of the key increases, the cipher quickly becomes inaccessible for human calculations on paper. The Hill cipher of dimension 6 (six-gram) when working with two characters at a time, the Hill cipher does not provide any specific advantages over the Playfair cipher and is even inferior to it in cryptographic strength and simplicity of calculations on paper. As

[15] If you use a number greater than 26 as the base of the unit, you can use a different number scheme to match letters to numbers and add spaces and punctuation.

[16] There are standard methods for calculating inverse matrices, but not all matrices have an inverse. The matrix will have an inverse if and only if its determinant is not zero and has no common divisors with the modulus base. If the determinant of the matrix is zero or has common divisors with the modulus base, then such a matrix cannot be used in the Hill cipher, and another matrix must be chosen (otherwise the ciphertext cannot be decrypted).

the dimension of the key increases, the cipher quickly becomes inaccessible for human calculations on paper. Hill cipher dimension 6 (hex).

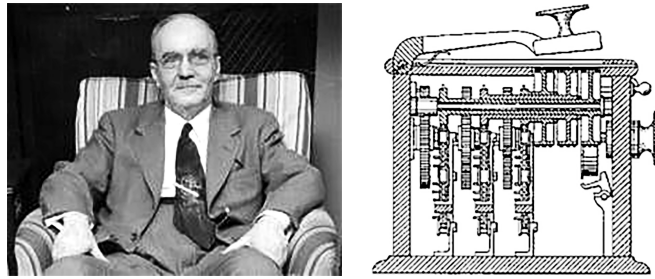


Figure 2.13 – Leicester Hill (1890-1961) and his cipher machine

Hill's cipher, although it was a polygram cipher, did not find practical application in cryptography due to its weak resistance[17] to cracking and the lack of descriptions of algorithms for generating large direct and inverse matrices[18].

With the development of «Black Cabinets»[19] in European countries in the 18th century, all ciphers of mono-alphabetic and polygram substitutions have lost all reliability. This fact contributed to the forced transition to the use of polyalphabetic ciphers.

Another reason for the popularization of a more complex type of encryption was the development of the telegraph and the emerging need to protect messages from interception.

---

[17] The standard Hill cipher is vulnerable to a selected plaintext attack because it uses linear operations. A cryptanalyst who intercepts n2 pairs of message symbols / ciphertext symbols will be able to compose a system of linear equations, which is usually easy to solve. If it turns out that the system is not solvable, then it is necessary to add a few more pairs of message symbols / ciphertext symbols.

Calculations of this kind using conventional linear algebra algorithms do not require a significant investment of time. In this regard, in order to increase the cryptographic strength, some nonlinear operations must be added to the Hill cipher. The combination of linear operations, as in the Hill cipher, and non-linear steps led to the creation of a permutation-permutation network (for example, the Feistel network). Therefore, modern block ciphers can be considered as a type of polygram ciphers.

[18] In World War II, Hill cipher machines were only used to encrypt the three-character code of radio signals.

[19] The Black Office is the body that deals with the perlustration and decryption of correspondence, and the room serving for this purpose, usually a secret room in the post office. The name comes from the corresponding French service Cabinet Noir.

A polyalphabetic substitution cipher consists of several simple substitution ciphers. Examples: Vigenère cipher, Beaufort cipher, one-time pad. Polyalphabetic substitution ciphers were invented by Lean Battista in 1568. The main idea of poly-alphabetic systems is that throughout the entire text, the same letter can be encrypted in different ways. That is, substitutions for a letter are selected from many alphabets, depending on the position of the letter in the text. This is a good defense against simple frequency counting, since there is no single (replacement) masking for each letter in the cryptotext. These ciphers use multiple one-letter keys, each of which is used to encrypt a single plaintext character. The first key encrypts the first character of the plaintext, the second encrypts the second, and so on. After using all the keys, they are repeated cyclically.

The Viginera cipher[20] refers to one of the first polyalphabetic ciphers. A distinctive feature of the Viginer cipher is that it is easy to understand and implement, and is also inaccessible to simple cryptanalysis methods.

Figure 2.14 – Leon Battista Alberti (1404-1472), Johann Trysemus (Trithemius) (1462-1516), Giovanni Battista Bellasso (1505-...), Blaise de Viginer (1523-1596)

The Vigenère cipher had a reputation for being extremely resistant to «manual» breaking[21].

---

[20] The first accurate documented description of this polyalphabetic cipher was formulated by Leon Battista Alberti in 1467. The encryption algorithm used a metal encryption disk to switch between alphabets. The Alberti system switches alphabets after a few cipher words. Later, in 1518, Johann Trysemus invented the tabula recta, the central component of the Vigenere cipher, in his work «Polygraphy».

What is commonly called the Vigenère cipher was first described by Giovanni Batista Bellazo, who introduced the concept of a «key» to switch between alphabets after each letter. Blaise Vigenère presented his description of this cipher before the commission of Henry III in France in 1586.

[21] It is officially believed that the Vizhiner cipher was recognized as cryptographically unstable after the publication of Kasiski's algorithm in the 19th century, although there are cases of breaking this cipher by some cryptanalysts back in the XVI century.

The Vigenère cipher is simple enough to be used in the field, especially if cipher discs or cipher rulers (Saint-Cyr rulers) are used. For example, the command of the Confederate army during the American Civil War used a copper encryption disk for the Vigenère cipher.

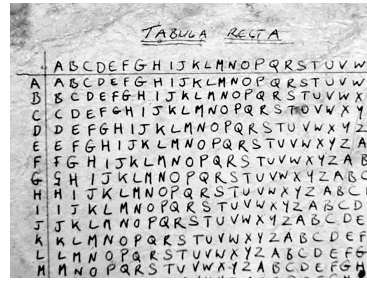


Figure 2.15 – Tabula recta or Vigenère table

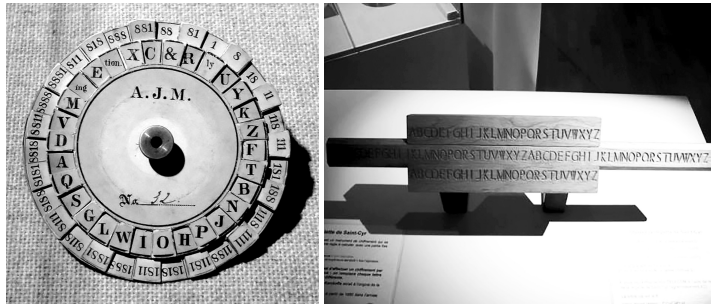


Figure 2.16 – Confederate army encryption disk and Saint-Cyr ruler[22]

The encryption algorithm does not contain complex transformations. First, a Vigenere table is created for encryption. As applied to the Latin alphabet, the Vigenere table is composed of lines of 26 characters, with each next line shifted by several positions[23].

Secondly, a key phrase is selected. If the length of this phrase (key) is less than the length of the encrypted text, then it is repeated several times: until the key length is equal to the length of the encrypted text.

[22] In the military academy of Saint-Cyr, they came up with a simple device consisting of two parts – an alphabet ruler and a movable slider with a written alphabet and a slot. In the Saint-Cyr line, a substitution cipher with a variable shift, the so-called Blaise de Vigenere cipher, was implemented.

[23] We can say that the Vigenere table gives 26 different Caesar ciphers.

Third, to encrypt the character of the encrypted text, the row of the Viginer table corresponding to the key character is selected. Thus, at each stage of encryption, different alphabets are used, selected depending on the character of the keyword.

For example, for the plain text «I COME ON THE SIXTH» and the keyword «ЛУНАТИК», we will apply the Vigenere algorithm. To do this, first, let's create a Vigenere table.

An example of a Vigenere table is shown in Figure 2.17.

We write down the ЛУНАТИК key in a loop until its length matches the length of the original text:

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

Figure 2.17 – Vigenère table for the Cyrillic alphabet for an example

The first character of the original text «Р» is encrypted with the sequence «Л», which is the first character of the key. The first character «Ы» of the ciphertext is at the intersection of line «Л» and column «П» in the Vigenere table.

Likewise, the second key character is used for the second character of the source text; that is, the second ciphertext character «Д» is obtained at the intersection of row «У» and column «Р». The rest of the source text is encrypted in a similar way.

```
Source text     –     ПРИЕЗЖАЮ _ ШЕСТОГО
Key             –     ЛУНАТИКЛУНАТИКЛУ
Cipher text     –     ЫДЦЕЪПКЙ _ ЁЕДЫЩОВ
```

Decryption is performed as follows: find in the Vigenere table the line corresponding to the first character of the keyword; in this line we find the first character of the ciphertext. The column containing this character corresponds to the first character of the source text. The following ciphertext characters are decrypted in a similar manner.

It seems that if the table is more complex than circular shifting of rows, then the cipher will become more reliable. This is true if you change it more often, for example, from word to word. But the compilation of such tables, where any letter occurs in a row or column once, is laborious and practically impossible to solve without using a computer. For a manual poly-alphabetic cipher, they rely only on the length and complexity of the key, using the above table, which can not be kept secret, and this simplifies encryption and decryption.

The Vigenere cipher «blurs» the characteristics of the frequencies of the appearance of symbols in the text, but some features of the appearance of symbols in the text remain. The main disadvantage of the Vigenere cipher is that its key is repeated and its length is fairly easy to calculate[24]. After that, it is not a problem to «crack» the Vigenère cipher.

In 1863, Friedrich Wilhelm Kasiski found a way to break the Vigenere cipher with a short codeword, the use of which was most common. In the case when encryption was performed using a key commensurate with the plaintext, the passphrase can be selected, provided that it consists of meaningful words. Attempts to invent a new burglar-resistant cipher for a long time did not lead to success, so cryptographers came up with such implementations of existing polyalphabetic ciphers in order to avoid them being cracked using the Kasiski method and the method of guessing the code word.

A further modification of the Vigenère system is the autokey cipher system. The idea of the auto-key is attributed to the XVI century mathematician J. Cardano. Encryption starts with the «primary key» (which is the real key in our sense) and continues with a message or cryptogram offset by the length of the primary key, then addition is performed modulo the cardinality of the alphabet.

[24] Friedman and Kasiski's tests determine the key length of the Vigenere cipher.

```
For example:

Message          ПРИЕЗЖАЮ _ ШЕСТОГО
Initial key      ЛУНАТИК
Auto key         ПРИЕЗЖАЮШ
Ciphertext       ЫДЦЕЪПКН _ БЙЩЩОБЖ
```

Legal decryption of a message using a known key is not difficult: the beginning of a message is obtained using the primary key, after which the found part of the original message is used as a key.

Gilbert Vernam of AT&T (American Telephone & Telegraph) tried to improve the cryptographic strength of the cracked Viginer cipher (the algorithm was later called the Vernam-Vigenere cipher in 1918, or simply the Vernam cipher).

In the classical sense, the Vernam cipher is a transformation of the plain text by a large non-repeating sequence of key characters.

The sender used each character in the key to encrypt only one character in the plaintext. Encryption is the addition modulo n (cardinality of the alphabet) of a plaintext character and a key character from a one-time pad[25]. Each key character is used only once and for a single message, otherwise, even if you use a large notebook, when the cryptanalyst receives several texts with overlapping keys, he will be able to recover the original text.

The cryptanalyst will shift each pair of ciphertexts relative to each other and count the number of matches in each position. If the ciphertexts are biased correctly, the match ratio will skyrocket. From this point of view, cryptanalysis is not difficult. If the key is not repeated and is random, then the cryptanalyst, whether he intercepts the texts or not, always has the same knowledge. A random key sequence, folded with a non-random plaintext, gives a completely random cryptotext, and no amount of computing power can change that.

The Vernam cryptosystem was proposed to encrypt telegraph messages, which were binary texts in which the plaintext is represented in Baudot code (in the form of five-digit «pulse combinations»). In this code, for example, the letter «A» looked like (11000). On the paper tape, the number «1» corresponded to the hole, and the number «0» – its absence. The secret key was to be a chaotic set of letters of the same alphabet and was marketed as a

[25] Vernam did not use the concept of «Exclusive OR» in the patent, but he implemented this very operation in relay logic. Each character in the message was bitwise XORed with a paper tape key.

disposable teletype tape. To obtain the ciphertext, the plaintext is combined with the exclusive-OR secret key.

So, for example, when using the key (11101) with the letter «A» (1 1 0 0 0) we receive an encrypted message (00101): (11000) (11101) = (00101). Knowing that for the received message we have the key (11101), it is easy to get the original message by the same operation: (00101) (11101) = (11000).

Gilbert Vernam created a device that performs these operations automatically, without the participation of an encryptor, and in 1919 received a patent for it. The Vernam apparatus contained magnets, relays and collector plates. Taking into account the fact that the encryption and decryption procedures are mathematically the same, this special apparatus could be used in the latter case. Encryption was carried out by entering pulses into the summation module from 2 readout panels: one read the «gamma», and the other – an open message. Combinations of values «+» and «-» was transmitted to the line as normal teletype data. At the receiving point, a second similar special apparatus added pulses read from a punched tape with an identical «gamut» and restored the original pulses of the original message.

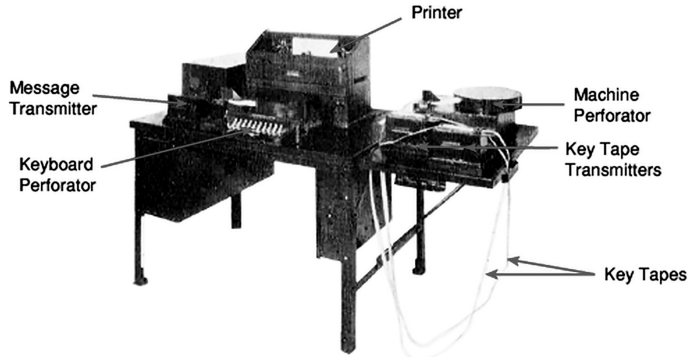


Figure 2.17 – A single-tape electromechanical cipher machine created in the USA around 1933 by Western Union Telegraph Company[26]

[26] These machines were based on the Vernam cipher.

With this device, Gilbert Vernam initiated the so-called *«linear encryption»* when the processes of encryption and transmission of a message occur simultaneously. Until that time, encryption was preliminary, so linear encryption significantly increased communication efficiency.

Also well known is the so-called Vernam cipher modulo m, in which the characters of the plaintext, ciphertext and key take values from the residue ring Zm. The cipher is a generalization of the original Vernam cipher, where m = 2.

For example, encoding with a Vernam cipher modulo m=26 (A=0,B=1,…, Z=25):

```
Key:              EVTIQWXQVVOPMCXREPYZ
Open text:        ALLSWELLTHATENDSWELL
                  All's well that ends well
Ciphertext:       EGEAMAIBOCOIQPAJATJK
```

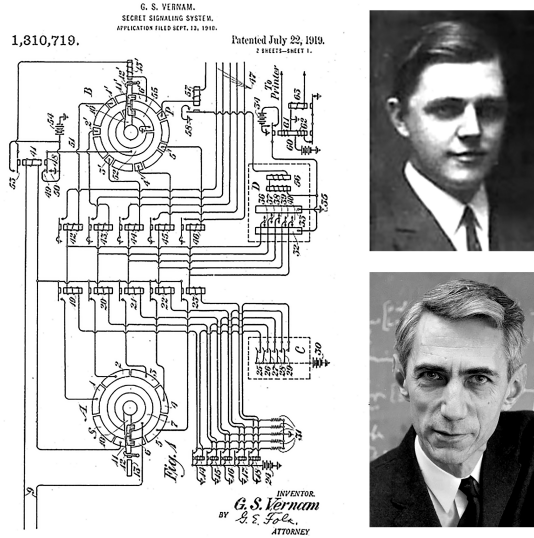


Figure 2.18 – Patent for Vernam's invention, Gilbert Stepford Vernam (1890-1960) and Claude Elvoord Chenon (1916-2001)

Major Joseph Mauborgne began to further refine the Werman method. He combined the randomness of the gamut with the one-time code pad rule. Now, three restrictions have been introduced for the encryption algorithm:

– the cipher pad was implemented as an encryption scale, equal in length or exceeding the encrypted message;
– the gamut signs were completely random or equally probable;
– each scale was used once and only once, after which it was destroyed by the transmitting or receiving correspondent.

There was also an additional rule: only two copies of the cipher key were made, one copy for the sender, the second copy for the receiving correspondent.

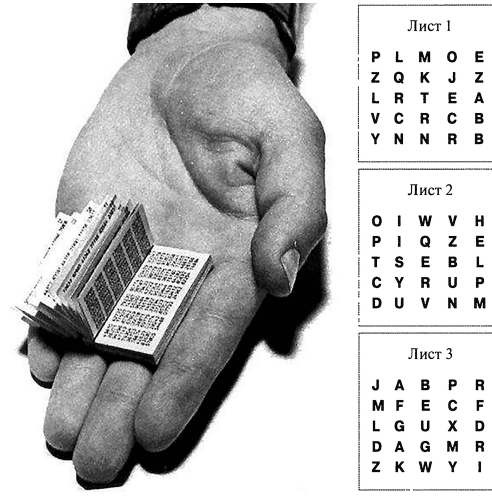


Figure 2.19 – Photo of the cipher note and three sheets of it, each of which is a possible key for the cipher

That being said, it should be noted that some cryptographers believed that they could create a huge number of random keys, at random, for example, by typing. However, at the same time, the typist (or the operator of the printing device) each time tried to type letters as follows: one letter with his left hand, the next with his right hand, and so on, alternately hitting the keys on one side or the other. In this way, it was indeed possible to quickly create a key, but the resulting sequence had a structure and, as a result, was no longer random – if the driver hit the key with the letter D on the left side of the keyboard, then the next letter is likely to be the letter on the right side of the keyboard. If the one-time cryptographic key is really random, then in about half of all cases, the letter on the left side of the keyboard should be followed by another letter on the left side of the keyboard.

With the advent of the one-time pad, it became clear that the best random keys are generated from natural physical processes, such as radioactivity. A cryptographer can take a large piece of radioactive ore and measure the radiation with a Geiger counter. Sometimes ionizing radiation particles are emitted one by one very quickly, sometimes a rather long time passes between separate emission events, therefore the time between these events is an unpredictable and random quantity. In this case, the alphabet runs through the alphabet quickly but at a constant speed on the random number generator in a cyclic mode, instantly stopping when the counter is triggered.

Whatever the letter of the generator, it can be used as the next letter of the random key. After that, scrolling through the alphabet again begins in a cyclic mode until the next operation of the counter, which occurs as a result of an ionizing particle hitting it.

Such a device is guaranteed to generate a truly random key, but it is unsuitable for day-to-day cryptography.

Even if we could generate sufficiently random keys, there would be another problem: the complexity of their distribution. Imagine a war zone where hundreds of radio operators form a single communications network. To begin with, they must all have identical copies of a one-time cipher pad. Then, when new cipher notes are prepared, they must be simultaneously transmitted to everyone. Finally, everyone needs to be sure that the right piece of one-time pad is used at the right time. Moreover, if the enemy captures at least one set of keys, the reliability of the entire communication system will be destroyed.

It is tempting to reduce the effort of preparing and distributing keys by reusing one-time cipher pads, but reusing one-time cipher pads allows an adversary's cryptanalyst to easily decrypt messages.

The practical shortcomings of the theoretically perfect one-time cipher pad meant that Mauborn's idea could never be widely applied in practice.

In 1945, Claude Shannon wrote the work «The Mathematical Theory of Cryptography», in which he proved the absolute cryptographic strength of the Vernam-Mauborgne cipher, called «one-time pad»[27].

From the point of view of cryptography, it is impossible to come up with a system safer than a one-time pad or, in the future, just a Vernam cipher. However, the requirements for the implementation of such an encryption scheme are quite nontrivial, since it is necessary to ensure the imposition

[27] It is not surprising, but the Vernam cipher class is the only cipher class for which it can be proved (and was proved by Shannon) non-disclosure in the absolute sense of this term.

of a unique gamut equal to the message length[28] followed by its guaranteed destruction. In this regard, the commercial use of the Vernam cipher is not so widespread, unlike public key schemes, and it was used mainly for the transmission of messages of particular importance by government agencies.

Currently, both the one-time pad and the Vernam cipher are rarely used. This is largely due to the significant key size, which must be the same length as the message. Nevertheless, completely strong ciphers like Vernam have found practical use for protecting critical communication lines with a relatively small amount of information. For example, the British and Americans used Vernam type ciphers during World War II. Vernam cipher mod 2 was used on the government hotline between Washington and Moscow, where key materials were paper ribbons, on which key sequence characters were applied using perforation.

In practice, it is possible to physically transmit a storage medium once with a long truly random key, and then send messages as needed. This is the basis of the idea of encrypted notepads: the encryptor is supplied with a notepad via diplomatic post or in person, each page of which contains keys. The host has the same notebook. The used pages are immediately destroyed after a single use[29].

With the development of computers, all polyalphabetic ciphers ceased to be so resistant to crypto attacks, and, just like mono-alphabetic ciphers in their time, receded into the background, becoming part of the history of cryptography.

---

[28] To get around the problem of pre-transmission of a large secret key, engineers and inventors have come up with many ingenious schemes for generating very long streams of pseudo-random digits from several short streams according to some algorithm. In this case, the recipient of the encrypted message must be equipped with exactly the same generator as the sender. But such algorithms add regularity to the ciphertext, detection of which can help the analyst to decrypt the message. Another way is to specify the location of the key as a place in the book. All characters included in the alphabet, starting from the specified place in the book, are used as a one-time key for any message. But in this case, the key will not be random and information on the frequencies of the distribution of letters can be used.

[29] There have been cases when the same notebook page was used twice for various reasons. For example, among the entire volume of Soviet encrypted correspondence intercepted by US intelligence in the 40s of the last century, messages were discovered that were covered with a twice used gamut. This period did not last very long, because after the first successes of American cryptanalysts in the late 1940s, the USSR secret services learned about serious problems with the reliability of their encrypted correspondence. Such messages were decrypted over the next 40 years as part of the secret Venona project, whose documents were recently declassified and posted on the NSA website.

2.2.3 Gumming

The gamma method consists in sequentially «superimposing» symbols of some special sequence, called a gamma or *gamma sequence*, on the characters of the encrypted plaintext. The summation is usually performed in some finite field.

For example, in the Galois field GF (2), the summation takes the form of an exclusive OR (xor) operation.

When using the «exclusive or» function, encryption is performed as follows:

$$c_i = m_i \oplus k_i \text{ for } i = 1,2,3... \quad (2.4)$$

where ci – ciphertext sign;
mi – plaintext sign;
ki – key gamma sequence sign;
⊕ – modulo addition 2.

Since re-XORing restores the original value, decryption is done by reapplying the gamma:

$$m_i = c_i \oplus k_i \text{ for } i = 1,2,3... \quad (2.5)$$

The principle of gamma encryption is to generate an infinite key (gamma cipher) using pseudo-random number generators (PRN) and superimpose the resulting gamma on the original data in a reversible way.

The process of decrypting data is reduced to re-generating a cipher gamut with a known key and superimposing such a gamut on the encrypted data.

If the gamma period exceeds the length of the entire ciphertext and no part of the original text is known, then we get a one-time notepad and the ciphertext can be considered impossible to open.

When encrypting with gamma methods, special requirements are imposed on the gamma sequence:

- to form a gamma (a sequence of pseudo-random numbers), you need to use hardware random number generators based on physical processes. If the gamma is not random, in order to obtain the plain text, it is necessary to select only the initial state of the pseudo-random number generator;
- the gamma length must not be less than the length of the protected message (plain text). Otherwise, to get the plaintext, you will need to

select the length of the gamma, analyze the blocks of the ciphertext of the guessed length, and select the bits of the gamma.

However, in practice, this requirement is met only in serious government agencies and software methods are used to generate gammas..

2.3 Key Generators

In 1949, Claude Shannon published a paper in which he identified three requirements for a key as a gamma sequence:

- the gamma sequence must be truly random;
- the gamma sequence must be the same size or larger than the specified plaintext[30];
- the gamma sequence should only be applied once.

Any sequence of random symbols can be used as such a scale. In practice, they use long random or pseudo-random keys generated with the help of special technical devices or software and hardware systems:

1) based on the use of devices based on physical processes, for example, registering nuclear decay, white noise, natural background radiation, cosmic radiation, etc.

2) based on the use of deterministic algorithms for generating pseudo-random numbers using Random functions, hash functions or recurrent formulas. It should be remembered that no deterministic algorithm can generate completely random numbers[31]. Any PRNG generator with limited resources sooner or later gets stuck in a loop – it starts repeating the same sequence of numbers.

An example of such an «unfortunate» algorithm is the infamous RANDU algorithm (one of the variants of the linear congruent pseudo-random number generator)[32], which has been used on mainframes for decades. It is defined by the recurrence relation:

---

[30] For modern symmetric algorithms (AES, CAST5, IDEA, Blowfish, Twofish, GOST 28147-89), the key strength is the key length. Encryption with keys of 128 bits or greater is considered strong because it takes years of powerful supercomputers to decrypt information without a key. Accordingly, keys containing repeating bit sequences will be weak. For example, the key «01010101» with a length of 8 bits can be represented as a shorter key «01» with a length of only 2 bits.

[31] As John von Neumann said, «anyone who has a weakness for arithmetic methods of obtaining random numbers is without a doubt a sin.»

32 According to the Kerchhoffs principle, the security of a cryptographic system should be determined by the concealment of secret keys, but not by the concealment of the algorithms used or their features.

$$V_{i+1} = (65539 \cdot V_i) \bmod 2^{31}, \quad (2.6)$$

where V_0 odd number.

An example of a pseudo-random sequence generated by the RANDU algorithm with an initial value $V_0 = 1$:

```
1
65539
393225
1769499
7077969
26542323
...
388843697
238606867
79531577
477211307
1 (repeat for item № 536 870 913).
```

In general, a linear congruent pseudo-random number generator is given by

$$X_{i+1} = (a X_i + b) \bmod m, \quad (2.7)$$

where a, b and m are some coefficients.

Unfortunately, linear congruential generators cannot be used in cryptography because they are predictable. Linear congruent generators were first hacked by Jim Reeds and then by Joan Boyar. She also managed to open up quadratic generators

$$X_{i+1} = (a X_i^2 + b X_i + c) \bmod m \quad (2.8)$$

and cubic generators

$$X_{i+1} = (a X_i^3 + b X_i^2 + c X_i + d) \bmod m. \quad (2.9)$$

The Blum – Blum – Shub (BBS) Algorithm, proposed in 1986 by Lenore Blum, Manuel Blum and Michael Shub, is a cryptographically strong PSN generator.

Figure 2.20 – Joan Boyar, Manuel and Lenore Blum

The BBS recurrent formula is as follows:

$$X_{i+1} = X_i^2 \bmod m, \tag{2.10}$$

where m = p • q – is the product of two large prime p and q, comparable to 3 modulo 4.

At each step of the algorithm, the output is obtained from X_i by taking either the parity bit or one or more least significant bits X_i.

```
Example using two small primes
p = 7 (7 mod 4 = 3) и q = 19 (19 mod 4 = 3).
m = 7 * 19 = 133.
X0 = 53.
```

№	X		Even parity bit	Least significant bit	2 least significant bits
	Dec-code	Bin-code			
0	53	110101	0	1	01
1	16	10000	1	0	00
2	123	1111011	0	1	11
3	100	1100100	1	0	00
4	25	11001	1	1	01
…	…	…	…	…	…
Gamma			01011…	10101…	0100110001…

Figure 2.21 – An example of generating a gamma using the BBS algorithm

A feature of the BBS algorithm is that to obtain X_n, it is not necessary to calculate all n–1 previous numbers, if the initial state of the generator X0 and the numbers p and q are known, then the n–th value can be calculated «directly» by the formula:

$$X_n = X_0^{2^n \bmod((p-1)(q-1))} \bmod m. \quad (2.11)$$

Also, so-called M-sequences have found application for generating gammas. An M or Maximum Length Sequence (MLS) is a pseudo-random binary sequence generated by a linear feedback shift register and has a maximum period.

The use of M-sequences as pseudo-random numbers is primarily due to the fact that M-sequences have very good periodic correlating functions (PCFs) and are generated using a simple scheme: an m – bit register covered by feedback through an adder modulo 2. Moreover, the length sequence, defined as

$$N = 2^m - 1, \quad (2.12)$$

practically unlimited: M-sequences of length up to (234 - 1) = 17,179,869,183 are known. Of all binary sequences, M-sequences have been studied most fully.

M-sequences are also called shift register sequences, linear recurrent sequences.

Thus, we can conclude that for the generation of gamma sequences, it becomes clear that the use of ordinary PRNG generators does not provide the necessary cryptographic strength. It is necessary to use specialized, so-called cryptographically strong PRNG generators. The required «quality» of randomness of the generated cryptographically strong PRNG generator varies from task to task[33].

Ideally, the generation of random numbers in a cryptographically strong PRNG generator uses a highly reliable source of entropy.

The requirements for a conventional pseudo-random number generator are met by cryptographically strong PRN generators, although the opposite is not true. Requirements for cryptographically strong PRNG generators

[33] For example, the generation of one random number in some protocols requires only uniqueness, while the generation of a master key or one-time cipher pad requires high entropy.

can be divided into two groups: first, they must pass statistical tests for randomness; and secondly, they must remain unpredictable, even if a part of their initial or current state becomes known to the cryptanalyst – compressed. Namely:

- a cryptographically strong PRN generator must satisfy the «next bit test»[34] (34);
- a cryptographically strong PRN generator must remain reliable even in the case when part or all of its states have become known (or have been correctly calculated). This means that it should not be possible to obtain a random sequence generated by the generator prior to the cryptanalyst acquiring this knowledge. In addition, if additional entropy is used during operation, the attempt to use knowledge of the input data should be computationally impossible.

Currently, the following cryptographically strong PRN[35] generators are widely used:

- Yarrow algorithm[36];
- Fortuna algorithm, which is the successor to the Yarrow algorithm;
- Blum-Micali algorithm;
- Fibonacci method with lags;
- Microsoft CryptoAPI;
- Java SecureRandom;
- Mersenne vortex;
- SAAC (Indirection, Shift, Accumulate, Add and Count) – developed in 1996 by Robert J. Jenkins Jr., as a development of the IA and IBAA[37] algorithms developed by him, etc.

Some of the listed PRNG generators use hash functions (SHA-1 and MD5).

2.4 Block Ciphers

Most modern software-implemented symmetric encryption algorithms

---

[34] The meaning of the test is as follows: there should be no polynomial algorithm that, knowing the first k bits of a random sequence, can predict the (k + 1) th bit with a probability of more than 50%. Andrew Yao proved in 1982 that a generator that passes the «next bit test» will pass any other statistical randomness test that runs in polynomial time.

[35] The previously considered BBS algorithm has a significant drawback - it is very «slow».

[36] Yarrow's algorithm is used in FreeBSD, OpenBSD and Mac OS X.

[37] This generator belongs to the category of cryptographically secure pseudo-random number generators although a complete and rigorous proof has not been carried out.

are divided into two classes: block and stream ciphers. A block cipher is a kind of a symmetric cipher that operates with groups of bits of a fixed length – blocks, the characteristic size of which varies within the range of 32-256 bits. If the original text (or its remainder) is less than the block size, it is supplemented before encryption. In fact, a block cipher is a substitution on the alphabet of blocks, which, as a consequence, can be mono- or polyalphabetic. The block cipher is an important component of many cryptographic protocols and is widely used to protect data transmitted over a network.

Unlike a one-time pad, where the key length is equal to the message length, a block cipher is able to encrypt one or more messages with a single key with a total length greater than the key length. Transmitting a key that is small compared to a message over an encrypted channel is a much simpler and faster task than transmitting the message itself or a key of the same length, which makes it possible to use it on a daily basis. However, in this case the cipher ceases to be unbreakable.

The modern block cipher model is based on the idea of iterative block ciphers proposed in a 1949 publication by Claude Shannon. This concept allows you to achieve a certain level of security by combining easy-to-perform substitution and swap operations.

The first in the field of block ciphers was the Lucifer[38] cipher, developed in 1970 by IBM as part of a research project and based on the SP-network. The project involved the later famous cryptographers Horst Feistel and Don Coppersmith.

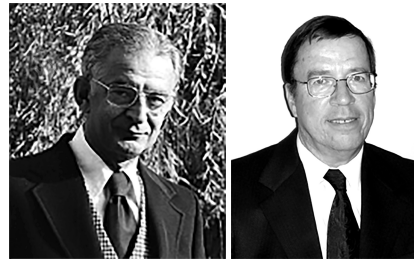


Figure 2.22 – Horst Feistel (1915-1990) and Don Coppersmith (1950)

[38] The first version of the algorithm, from 1971, used 48-bit blocks and keys and was based on SP networks. A subsequent modification of the algorithm was patented in November 1971 (U.S. Patent 3 796 830). This cipher used 64-bit keys and 32-bit blocks. The last, third, version, proposed in 1973, operated with 128-bit blocks and keys.

The structure of the Lucifer algorithm of the June 1971 sample is an SP-network (or permutation-permutation network) – a «sandwich» of two types of layers used in turn. The first type of layer is a 128-bit P-block, followed by a second layer, which is 32 modules, each of which consists of two 4-bit S-blocks, whose corresponding inputs are shorted and the same value is supplied to them from the output the previous layer. But the substitution blocks themselves are different (differ in the substitution tables). The module outputs values from only one of the S-boxes, which one is determined by one of the bits in the key, the number of which corresponded to the number of the S-box in the structure.

A simplified diagram of the algorithm with a lower bit depth and an incomplete number of rounds is shown in Figure 2.24. It uses 16 S-box selection modules (32 S-boxes in total), so this scheme uses a 16-bit key.

Figure 2.25 shows how the ciphertext changes in the given algorithm when only one bit is changed. For simplicity, the tables of S-boxes substitutions are taken such that if all zeros are fed to the input of the S-box, then all zeros will be at the output. In real systems, such substitution tables are not used, since they greatly simplify the work of a cryptanalyst, but in the example they clearly illustrate a strong intersymbol relationship when one bit of an encrypted message is changed.

Figure 2.24 shows that thanks to the first P-box, the only unit is shifted to the center of the block, then the next nonlinear S-box «multiplies» it, and already two units change their position due to the next P-box, and so on.

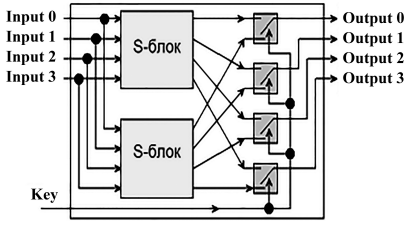


Figure 2.23 – The module that selects the used lookup table by the bit key

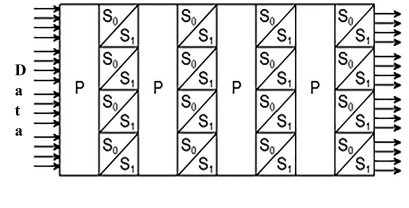


Figure 2.24 – Simplified diagram of S- and P-layers in the «Lucifer» algorithm (June 1971)

At the end of the encryption device, due to the strong intercharacter link, the output bits have become a complex function of the input and the

key used. On average, at the output, half of the bits will be «0» and half will be «1».

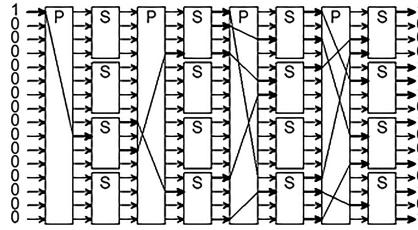


Figure 2.25 – Scheme of generation and distribution of units

The idea of the «Lucifer» cipher was to use combinations of simple, and, therefore, quickly calculated both hardware and software operations. However, the algorithm turned out to be unsuccessful: it was too cumbersome, which led to a low encryption speed both in software implementation (about 8 kb / s) and in hardware (97 kb / s).

In the early 70s, concerns about the robustness of this algorithm began to appear. However, the principles developed during the construction

«Lucifer» (SP-network and Feistel network, named after one of the developers), formed the basis for the construction of block ciphers.

In 1973, the National Institute of Standards and Technology (NIST) announced a competition to develop a data encryption standard, the winner of which was the Data Encryption Standard (DES) in 1974, which is, in fact, an improved version of Lucifer[39] (39).

The block size for DES is 64 bits. The algorithm is based on a Feistel network with 16 cycles (rounds) and a 56-bit key. 8 bits were used to form imitations.

The DES encryption scheme is shown in Figure 2.26. The source text is divided into blocks of 64 bits.

The encryption process consists of an initial permutation, 16 encryption cycles and a final permutation.

[39] The publication of the DES cipher in 1977 was fundamental to the public understanding of the modern block cipher model..

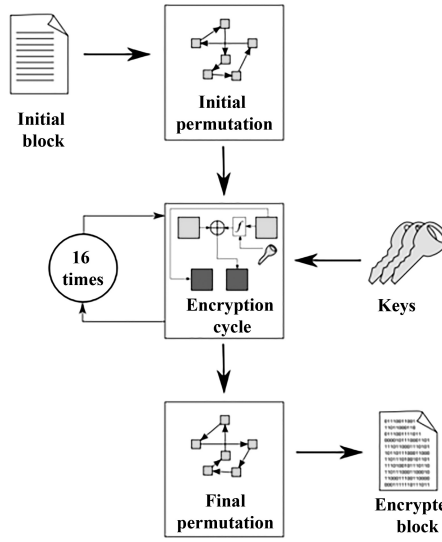


Figure 2.26 – DES encryption scheme

In the initial permutation, the original text T (block of 64 bits) is transformed using the permutation, which is defined in Table 2.5.

Table 2.5 – Initial permutation of the DES ID algorithm

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

According to table 2.5 of the initial permutation, the first 3 bits of the resulting block after the initial permutation are bits 58, 50, 42 of the source block, and its, for example, the last 3 bits are bits 23, 15, 7 of the input block.

The 64-bit block of text obtained after the initial permutation participates in 16 Feistel transform cycles.

To do this, in the conversion cycles, the input block is initially divided into two parts L_0 and R_0, where L_0 and R_0 are respectively 32 most significant bits and 32 least significant bits of the text block or ID $(T_0) = L_0R_0$.

The result of the i-th iteration of the cycle T_i is defined as:

$$L_i = R_{i-1}, \qquad (2.13)$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, k_i).$$

where is the Feistel function, which plays the role of encryption in these cycles.

The arguments of the function f (R_{i-1}, k_i) are the 32-bit vector R_{i-1} and the 48-bit key k_i, which is the result of transforming the 56-bit original key of the cipher K.

To calculate the function f (R_{i-1}, k_i), the following are sequentially used:

1. extension function E;
2. operation of addition modulo 2 with the key E (R_{i-1}) k_i;
3. transformation S, consisting of 8 transformations of S-boxes S_1, S_2, S_3, ..., S_8:
4. permutation P.

Function E expands a 32-bit vector R_{i-1} to a 48-bit vector E (R_{i-1}) by duplicating some of the bits from R_{i-1}; bit order of vector E (R_{i-1}) is shown in Table 2.6.

Table 2.6 – Expansion function E

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

Table 2.6 shows that bits 1, 4, 5, 8, 9, 12, 13, 16, 17, 20, 21, 24, 25, 28, 29, 32 are duplicated. The block E (R_{i-1}) obtained after the permutation is added modulo 2 with keys ki and then represented as eight sequential blocks B_1, B_2, B_3, ..., B_8.

Each B_j is a 6-bit block. Further, each of the blocks B_j is transformed into a 4-bit block B_j ‘with the help of transformations S. The transformations S_j are defined in Table 2.7.

Таблица 2.7 – Преобразования S_j, i = 1…8

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	S1
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0	
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	S2
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	S3
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	S4
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	S5
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11	S6
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	S7
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	S8
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

$$B_1, B_2, B_3, \ldots, B_8 = E(R_{i-1}) \oplus k_i. \qquad (2.14)$$

Suppose B_3 = 101111 and we want to find B_3'. The first and last bits of B3 are binary notation of the number a, 0 <= a <= 3, the middle 4 bits represent the number b, 0 <= b <= 15. Rows of table S_3 are numbered from 0 to 3, columns of table S_3 are numbered from 0 to 15. A pair of numbers (a, b) defines the number at the intersection of row a and column b. The binary representation of this number gives B_3'. In our case, a = 112 = 3, b = 01112 = 7, and the number defined by the pair (3.7) is 7. Its binary representation is B_3'= 01112.

The value of the function f (R_{i-1}, k_i) (32 bits) is obtained by the permutation P applied to the 32-bit block B_1, B_2, B_3, ..., B_8. Permutation P is given by Table 2.8.

Table 2.8 – Permutation P

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

The transformation P is represented by the formula

$$f(R_{i-1}, k_i) = P(B_1', B_2', B_3', \ldots, B_8') \quad (2.15)$$

According to table 2.8, the first four bits of the resulting vector after the action of the function f are bits 16, 7, 20, 21 of the vector B_1', B_2', B_3', …, B8'.

Keys ki of DES algorithm are derived from an initial 56-bit key K as follows:

– bits are added at positions 8, 16, 24, 32, 40, 48, 56, 64 of the key K so that each byte contains an odd number of ones. This is used to detect errors in key exchange and storage.
– do a permutation for the extended key (except for the added bits 8, 16, 24, 32, 40, 48, 56, 64). Such a permutation is defined in table 2.9.

Table 2.9 – First permutation for key generation

57	49	41	33	25	17	9	1	58	50	42	34	26	18	C_0
10	2	59	51	43	35	27	19	11	3	60	52	44	36	
63	55	47	39	31	23	15	7	62	54	46	38	30	22	D_0
14	6	61	53	45	37	29	21	13	5	28	20	12	4	

This permutation is defined by two blocks C_0 and D_0 of 32 bits each. The first 3 bits of C_0 are bits 57, 49, 41 of the extended key. And the first three bits of D_0 are bits 63, 55, 47 of the extended key. C_i, D_i for i = 1,2,3 ... are obtained from C_{i-1}, D_{i-1} by one or two left cyclic shifts according to table 6.

Table 2.10 – Shift number C_{i-1}, D_{i-1} for the i-th iteration of the encryption key generation

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

The key k_i, i = 1, ... 16 consists of 48 bits, selected from the bits of the vector C_i, D_i, containing 64 bits, according to table 2.11. The first and second bits k_i are bits 14, 17 of the vector C_i, D_i.

Table 2.11 – Key sampling table k_i

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

The final text encryption permutation T0, called ID^{-1}, acts on T16 and is the inverse of the original ID permutation. The final permutation is determined by Table 2.12.

Table 2.12 – Reverse permutation ID^{-1}

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

The detailed DES encryption scheme is shown in Figure 2.27.

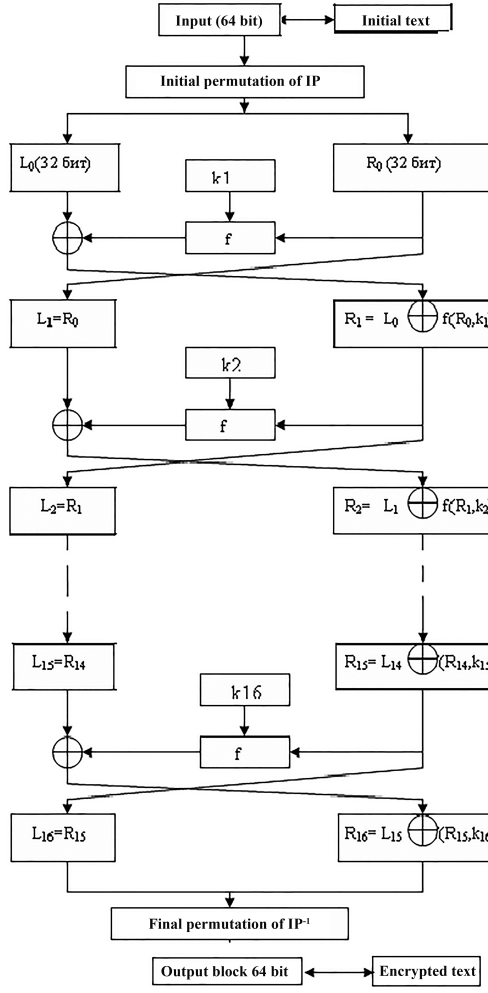


Figure 2.27 – Detailed DES encryption scheme

When decrypting data, all actions are performed in reverse order. In 16 decryption cycles, the inverse transformation by the Feistel network is used.

$$R_i = L_{i-1}, \qquad (2.16)$$
$$L_i = R_{i-1} \oplus f(L_{i-1}, k_i).$$

The decryption scheme is shown in Figure 2.28.

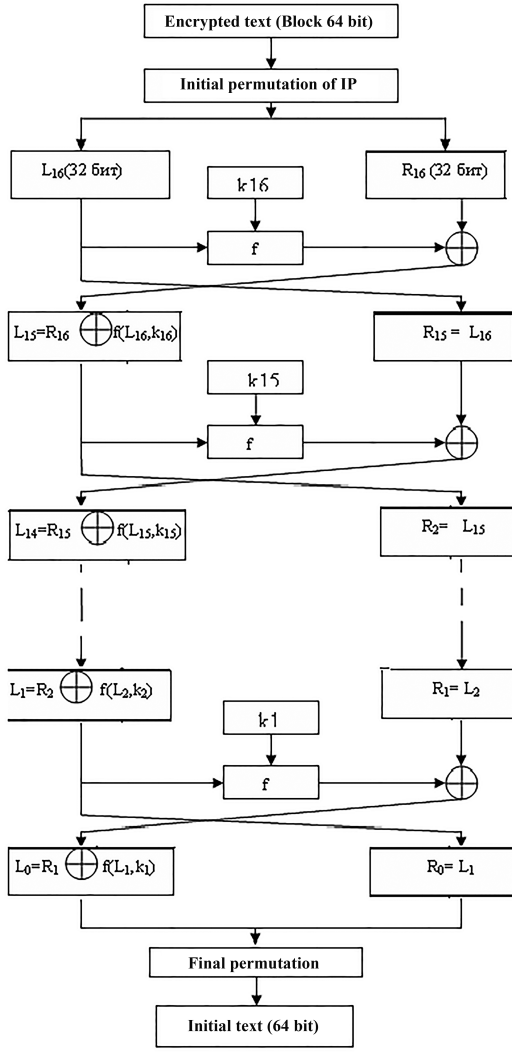


Figure 2.28 – DES decryption scheme

Key K, i = 16, ..., 1, function f, permutation of IP and ID^{-1} are the same as in the encryption process.

If the previously generated keys were not saved in the implementation, the keys we need (C_i, D_i, i = 1,2,3 ...) are obtained from C_{i-1}, D_{i-1} by right cyclic shifts according to Table 2.13.

Table 2.13 – Shift number C_{i-1}, D_{i-1} for generating decryption keys

Iteration i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Shift number	0	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

After being approved by the American National Standards Institute in 1981, the algorithm has been used for a long time in the civilian sector and even went beyond the United States.

Several modes of use have been recommended for DES:

- ECB (electronic code book) – «electronic code book» mode (for modern block ciphers it is called «simple replacement»). In this mode, each 64-bit block of text is encrypted independently of each other;
- CBC (cipher block chaining) – block chaining mode – one of the encryption modes for a symmetric block cipher using a feedback mechanism. Each block of plaintext (except the first) is bitwise added modulo 2 (XOR operation) with the previous encryption result;
- CFB (cipher feed back) – ciphertext feedback mode. In this mode, to encrypt the next block of plaintext, it is added modulo 2 with the re-encrypted (block cipher) encryption result of the previous block;
- OFB (output feed back) – output feedback mode. A feature of this mode is that the message itself is not used as input data for the block cipher algorithm. Instead, a block cipher is used to generate a pseudo-random stream of bytes, which is XORed onto the plaintext blocks. This encryption scheme is called a stream cipher;
- Counter Mode (CTR) – counter mode. This mode assumes the return to the input of the corresponding block cipher algorithm of the value of some counter accumulated since the start. The mode also makes a stream cipher out of a block cipher.

Despite the widespread use of the DES algorithm, the cipher had a significant drawback – a small key length, which gave rise to many attacks related to parallel search. For DES, the enumeration complexity corresponds to 2^{56} or approximately 10^{17} operations, which can be «hacked» by a cluster

of several dozen parallel computers or one supercomputer in a fairly short time[40].

Also, the DES algorithm «suffered» the problem of having weak and partially weak keys.

The lack of decent protection against DES cipher attacks gave rise to many algorithms that are like a more complex version of DES (2DES – plain text was encrypted sequentially using two different keys (112 bit key), 3DES – plain text was encrypted sequentially using three different keys (168 bit), DESX, G – DES) and completely different schemes (NewDES, FEAL, IDEA).

The DES algorithm was a national US standard in 1977-1980, but at present DES is used (with a 56-bit key) only for legacy systems, most often its more crypto-resistant form (3DES, DESX)[41] is used.

Algorithm IDEA (Intrnational Data Encryption Algorithm), developed and patented by the Swiss company Ascom, was conceived as a replacement for the encryption standard DES and did not become such because of its patency and the need for licensing for commercial applications. The final version of the algorithm was published in 1992.

The IDEA algorithm, like DES, operates on 64-bit plaintext blocks. The IDEA algorithm has several advantages over the DES algorithm. It is significantly more secure than DES because the 128-bit IDEA key is twice the DES key. Internal structure of IDEA algorithm provides better resistance to cryptanalysis. Existing software implementations are about twice as fast as DES implementations. The disadvantage of the algorithm is its focus on 16-bit architecture, which reduces the efficiency of use on 32 and 64-bit computing means.

---

[40] In 1990, DES was cracked in 39 days using a vast network of tens of thousands of computers.
The public organization «EFF», dealing with the problems of information security and personal secrets on the Internet, initiated a study «DES Challenge II» in order to identify problems with DES. As part of the study, RSA Laboratory employees built a $ 250,000 supercomputer. In 1998, the supercomputer decrypted DES-encoded data using a 56-bit key in less than three days. The supercomputer was named «EFF DES Cracker».

[41] The DES algorithm is still widely used to protect financial information: for example, the THALES (Racal) HSM RG7000 module fully supports TripleDES operations for issuing and processing credit cards VISA, EuroPay and others. THALES (Racal) DataDryptor 2000 channel encoders use TripleDES to transparently encrypt data streams. The DES algorithm is also used in many other devices and solutions THALES-eSECURITY..

Figure 2.29 – Xuejia Lai and James Massey, developers of the IDEA algorithm

1997 marked the beginning of the US government's program to adopt the AES (Advanced Encryption Standard)[42], developed by Belgians V. Rijmen and J. Daemen.

The AES algorithm is sometimes also called Rijndael, after the name of the algorithm that formed its basis. Strictly speaking, AES and Rijndael are not exactly the same, since AES has a fixed block size of 128 bits and key sizes of 128, 192 and 256 bits, while any block and key size can be specified for Rijndael, from minimum of 32 bits to maximum of 256 bits in 32-bit steps.

Figure 2.30 – Ioan Daimon and Vincent Reiman

[42] On January 2, 1997, NIST (National Institute of Standards and Technology) announced its intention to choose an encryption algorithm to replace DES, and in September 1997, formal requirements for the algorithms were presented. These requirements stated that NIST's goal is to develop an unclassified, well-analyzed encryption algorithm that is available for widespread use. The algorithm must be symmetric, block-based, support 128-bit block lengths and 128, 192 and 256-bit keys. In August 1998, NIST announced fifteen candidates for the AES algorithm at the first AES candidate conference. These algorithms were developed by industry and academia in twelve countries around the world. The second AES Candidate Conference was held in March 1999 to discuss the analysis of the proposed algorithms. In August 1999, five finalists selected by NIST were presented. They are MARS, RC6TM, Rijndael, Serpent and Twofish. On October 2, 2000, it was announced that the Rijndael algorithm became the winner of the competition, and the procedure for its standardization began. The draft was published on February 28, 2001, and on November 26, 2001, AES was adopted as FIPS 197 (Federal Information Processing Standard).

The algorithm represents each block of encoded data in the form of a two-dimensional byte array of 4x4, 4x6, or 4x8, depending on the specified block length. Further, at the appropriate stages, conversions are performed either on independent columns, or on independent rows, or generally on individual bytes in the table.

All transformations in the cipher have a strict mathematical justification. The structure itself and the sequence of operations make it possible to execute this algorithm efficiently on both 8-bit and 32-bit processors. The structure of the algorithm provides for the possibility of parallel execution of some operations, which on multiprocessor workstations can still increase the encryption speed by 4 times.

Table 2.14 shows the basic designations of functions and parameters of the AES critical conversion.

Table 2.14 – Algorithm parameters, symbols and functions

Name	Meaning
1	2
AddRoundKey()	An Encryption and Decryption Conversion in which the Round key is added to state using the XOR operatio
InvMixColumns()	Conversion to Decryption, which is the inverse of MixColumns..
InvShiftRows()	Conversion to Decryption, which is the inverse of ShiftRows.
InvBytesSub ()	Conversion to Decryption, which is the inverse of SubBytes
K	The encryption key is an array of 128 bits or 16 bytes.
MixColumns()	Transformation in the process of Encryption that takes all state columns and mixes their data (independently of each other) to get new columns.
Rcon()	An array of persistent round Words.
RotWord()	A function used in the Key Expansion operation, it takes a four-byte word and performs a circular swap.
ShiftRows()	Transformation in the process of Encryption, which is performed on state, cyclically shifting the last 3 lines of state by different values.
BytesSub ()	Transformation in the process of Encryption, which is performed on state and consists in replacing each byte using a replacement table (S-box).

N_k	Key length in words, for AES 4 words.
N_b	Key length in words, for AES 4 words
N_r	The number of rounds for encryption – 10.

The key size in the AES algorithm is 128 bits, so the key is usually represented as a 4x4 byte matrix.

At the beginning of the encryption process, the plaintext input is split into blocks of 16 bytes or 128 bits. If the total data size is not a multiple of 16 bytes, the data is padded to a size that is a multiple of 16 bytes. The data block in the AES algorithm is called state and is usually represented as a 4x4 byte matrix.

The encryption operation for each data block is carried out independently of the contents of other blocks. At the end of block encryption, the matrix is filled with the next portion of data and the process is repeated. As already noted, due to the independence of the encryption of one block from another, the encryption process lends itself well to parallelization.

S_{00}	S_{01}	S_{02}	S_{03}
S_{10}	S_{11}	S_{12}	S_{13}
S_{20}	S_{21}	S_{22}	S_{23}
S_{30}	S_{31}	S_{32}	S_{33}

Figure 2.31 – Representation of AES data block – state

Each block is encrypted in several stages – rounds[43]. The crypto transformation scheme can be written with the following description:

1. the KeyExpansion key is expanded;
2. the initial operation is performed – AddRoundKey – summation with the main key;
3. 9 rounds of four steps are performed:
   3.1. the BytesSub procedure is executed – replacing the state bytes according to the replacement table;
   3.2. procedure ShiftRows is executed – cyclic shift of rows state;
   3.3. the MixColumns procedure is executed – permutation of the state columns;
   3.4. the AddRoundKey procedure is executed – summation with a round key;

[43] The number of rounds - from 10 to 14 - depends on the chosen block size and key length.

4. the final 10th round is performed:
   4.1. the BytesSub procedure is executed – replacing the state bytes according to the replacement table;
   4.2. procedure ShiftRows is executed – cyclic shift of rows state;
   4.3. the AddRoundKey procedure is executed – summation with a round key.

The BytesSub transform is a non-linear byte substitution performed on each state byte using an S-box substitution table (Table 2.15).

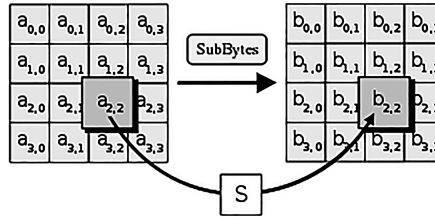


Figure 2.32 – ByteSub – 8x8 bit table substitution

The purpose of the BytesSub substitution table is to make linear and differential cryptanalysis difficult. The replacement table in the AES algorithm is fixed. In table 2.15 numbers are presented in hexadecimal notation, in this numeral system any byte value can be represented by no more than two hexadecimal digits

Table 2.15 – S-box byte replacement table for the AES algorithm

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	Fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73

90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Byte replacement in the S-box table is performed according to the following algorithm:

- byte Z is converted to hexadecimal notation, in the format XY h, X is the most significant bit, Y is the least significant bit. If there is no most significant bit, it is replaced by zero.
- in the S-box, row X and column Y are selected.
- the Z 'value at the intersection of row X and column Y of the S-box table is used as a replacement for Z.

The complete SubBytes process is to replace all 16 bytes of the *state* matrix.

In the ShiftRows transformation, the bytes in the last three strings of state are cyclically shifted to the left by a different number of bytes. Line 1 (line numbering from zero, shifted by one byte, line 2 – by two bytes, line 3 – by three bytes. Figure 2.33 illustrates the application of the ShiftRows transformation to state.

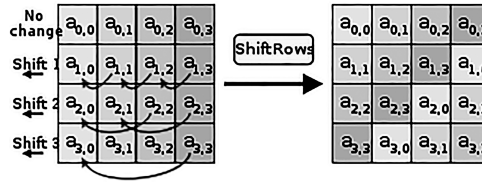


Figure 2.33 – ShiftRow – shifting rows in a two-dimensional array at different offsets

In the MixColumns transformation – mixing a column – the state columns are treated as polynomials over the field $F(2^8)$ and multiplied modulo $x^4 + 1$ by a constant polynomial:

$$a(x) = 3x^3 + 1x^2 + 1x^2 + 2 \tag{2.17}$$

The process of multiplying polynomials is equivalent to matrix multiplication

$$\begin{bmatrix} S'_{0c} \\ S'_{1c} \\ S'_{2c} \\ S'_{3c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \bullet \begin{bmatrix} S_{0c} \\ S_{1c} \\ S_{2c} \\ S_{3c} \end{bmatrix},$$

where c is the column number of the state array and $0 \le c \le 3$.

As a result of this multiplication, the bytes of the column c $\{S_{0c}, S_{1c}, S_{2c}, S_{3c}\}$ are replaced, respectively, by bytes

$$\begin{aligned} S'_{0c} &= (2 \bullet S_{0c}) \oplus (3 \bullet S_{1c}) \oplus S_{2c} \oplus S_{3c}; \\ S'_{1c} &= S_{0c} \oplus (2 \bullet S_{1c}) \oplus (3 \bullet S_{2c}) \oplus S_{3c}; \\ S'_{2c} &= S_{0c} \oplus S_{1c} \oplus (2 \bullet S_{2c}) \oplus (3 \bullet S_{3c}); \\ S'_{3c} &= (3 \bullet S_{0c}) \oplus S_{1c} \oplus S_{2c} \oplus (2 \bullet S_{3c}). \end{aligned} \tag{2.18}$$

The transformation (2.18) is applied to each of the four state columns.

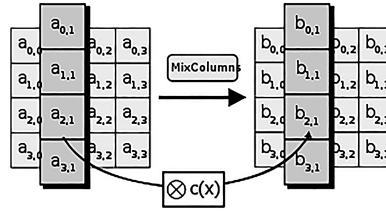


Figure 2.34 – MixColumn – mathematical transformation that mixes data inside a column

In the AddRoundKey transformation, the RK is added to state by bitwise XOR. Each round key consists of 16 extended key bytes. The round key bytes are written into a 4x4 matrix like state. Each byte of the round key is summed with the corresponding byte from state, as shown in Figure 2.35.

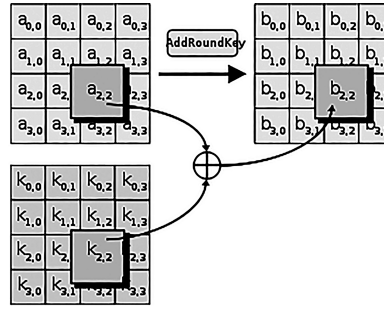


Figure 2.35 – AddRoundKey – adding a key material by XOR operation

In the last round, there is no column shuffling operation, which makes the entire sequence of operations symmetric.

For the algorithm to work, the specified key is expanded and, based on this extension, it creates round keys.

The extended key W contains 4 × (10 + 1) words – an initial key of 4 words and 4 words of the extended key for each of the 10 rounds. The extended key W consists of words (four bytes per word), denoted below as wi, where i is in the range [0..44]. Full length extended key 1408 bit, 128 bits for each round.

The key expansion process uses an array of Rcon constants. The elements of the Rcon array are numbered from 1 to 256 + 3. The values of the array elements are defined by the following set of functions:

$Rcon_1 = 1;$
$Rcon_k = 2 \bullet Rcon_{k-1} = 2^{k-1}$, for k = 2, 3,...255;
$Rcon_k = 0$, for k = 256, 257, 258;

Key expansion can be described by the following sequence of operations:

– four words of the encryption key K are copied into the first four words of the extended key W: $w_i = k_i$ for i = 0, 1, 2, 3;
– other words of the extended key W for i = 4, 5, ... 44 are generated as follows: if i is a multiple of 4, then w_i = SubBytes(RotByte(w_{i-1})) Rcon(i/4; therwise, if i is not a multiple of 4, то $w_i = w_{i-4}\ w_{i-1}$.

The RotByte function permutes the four bytes of the source word {a_0, a_1, a_2, a_3} using a cyclic permutation, turning it into a word {a_3, a_1, a_2, a_0}. The SubBytes function applies an S-box substitution to each of the four bytes of the word.

The round key RK for round k is selected from the extended key W as words w_{4k} through $w_{4(k+1)}$.

All encryption transformations are unambiguous and, therefore, have the inverse transform, i.e. can be inverted and reversed to perform decryption for the AES algorithm.

The encryption scheme for decryption can be represented by the following algorithm:

1. the KeyExpansion key is expanded;
2. 9 rounds of four steps are performed each;
   2.1. the AddRoundKey procedure is executed – summation with a round key;
   2.2. the InvMixColumns procedure is executed – reverse permutation of the state columns;
   2.3. procedure InvShiftRows is executed – reverse cyclic shift of rows state;
   2.4. the procedure InvSubBytes is performed – reverse replacement of the state bytes according to the replacement table;
3. the final 10th round is performed;
   3.1. the AddRoundKey procedure is executed – summation with a round key;
   3.2. procedure InvShiftRows is executed – reverse cyclic shift of rows state;
   3.3. the procedure InvSubBytes is executed – reverse replacement of the state bytes according to the replacement table.

The InvMixColumns transformation is the inverse of the MixColumns transformation. In the InvMixColumns transformation, the state columns are treated as polynomials over the field $F(2^8)$ and multiplied modulo x^{4+1} with a constant polynomial $d(x) = a^{-1}(x)$, в поле $F(2^8)$:

$$d(x) = 0Bhx^3 + 0Dhx^2 + 09hx + 0Eh. \quad (2.19)$$

The InvShiftRows transformation is inversely the ShiftRows transformation. The bytes of the last three rows of the state array are cyclically shifted to the right. Line 1 (numbering from zero) is shifted by 1 byte, line 2 – by 2 bytes, line 3 – by 3 bytes.

The InvSubBytes transform performs the reverse byte replacement using the InvS-box reverse replacement table.

Table 2.16 – Table InvS – box for reverse byte replacement

	0	1	2	3	4	5	6	7	8	9	A	b	c	d	e	f
00	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
10	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
20	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
30	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
40	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
50	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
60	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
70	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
80	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
90	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a0	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b0	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c0	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d0	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e0	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f0	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

In June 2003, the US National Security Agency ruled that the AES cipher was strong enough to be used to protect state secrets. Up to the SECRET level, it was allowed to use keys with a length of 128 bits, for the TOP SECRET level, keys with a length of 192 and 256 bits were required. However, the US government has mandated that AES should be periodically reviewed and improved to keep encrypted data secure[44]. The cryptographic strength of modern symmetric encryption algorithms for brute force attacks (brute force methods) is largely determined by the key length. Thus, Table 2.17 shows the characteristics of cryptographic strength in the number of operations and estimated time for various algorithms.

[44] AES support (and only it) was introduced by Intel into the x86 processor family starting with Intel Core i7-980X Extreme Edition, and then on Sandy Bridge processors.

Table 2.17 – Crypto resistance of encryption algorithms[45]

Key length (in bits)	Number of combinations	Estimated break time
16	$2^{16} = 65536$	
56 (DES)	$2^{56} = 7{,}2 \cdot 10^{16}$	399 секунд
128 (AES-128)	$2^{128} = 3{,}4 \cdot 10^{38}$	$1{,}02 \cdot 10^{18}$ лет
256 (AES-256)	$2^{256} = 6{,}2 \cdot 10^{57}$	$1{,}872 \cdot 10^{37}$ лет
512 (AES-512)	$2^{512} = 1{,}1 \cdot 10^{77}$	$3{,}31 \cdot 10^{56}$ лет

Alternative cryptoalgorithms have been implemented based on the Rijndael algorithm, which is the basis of AES. Among the most well-known algorithms are participants in the Nessie: Anubis competition on involutions, which was authored by Vincent Raiman and a strengthened version of the cipher – Grand Cru by Johan Borst. When considering symmetric block ciphers, one cannot but dwell on one more world-famous algorithm – GOST 28147-89 – the Soviet and Russian standard..

GOST[46], introduced in 1990, is also the standard of the CIS countries. The full name is «GOST 28147-89 Information processing systems. Cryptographic protection. Algorithm of cryptographic transformation». The algorithm, when using the gamma ciphering method, can perform the functions of a stream cipher algorithm[47].

---

[45] The announced results of the research by Alex Biryukov, Orr Dunkelman, Nathan Keller, Dmitry Khovratovich and Adi Shamir at the CRYPTO 2011 conference, the results of the cryptanalysis of the AES algorithm indicate the attack method, allowing four times to reduce the complexity of performing operations on the selection of a secret key. In other words, in fact, the cryptographic strength of AES-128 is reduced to AES-126, and AES-192 to AES-189. However, even with these solutions, the "break-in" time exceeds billions of years.

[46] According to some reports, the history of this cipher is much more ancient. The algorithm that later became the basis of the standard was born, presumably, in the bowels of the Eighth Main Directorate of the KGB of the USSR (now in the structure of the FSB of the Russian Federation), most likely in one of the closed research institutes under its jurisdiction, probably back in the 1970s within the framework of projects creation of software and hardware implementations of the cipher for various computer platforms.

From the moment the GOST was published, it had a restrictive stamp "For official use", and formally the code was declared "fully open" only in May 1994. Unfortunately, the history of the creation of the cipher and the criteria for the developers have not yet been published.

[47] By analogy with AES (and unlike DES), GOST is allowed to protect secret information without restrictions, in accordance with the way it is specified in the Russian standard. Thus, GOST is not an analogue of DES, but a competitor to 3DES with three independent keys or AES-256.

The GOST 28147-89 algorithm uses a 256-bit encryption key and 32 conversion cycles of 64-bit blocks of the original plaintext. The algorithm implements the classic Feistel network scheme. A simplified algorithm for implementing the generating function is shown in Figure 2.36.

Initially, the right half of the block and the 32-bit round key are added modulo 2. The result of the addition is divided into eight 4-bit sequences, each of which is fed to the input of the corresponding S-block.

Each block is a lookup table that replaces the incoming number in the range [0..15] with another number in the same range. The outputs of all S-boxes are concatenated into a 32-bit word, which is then cyclically shifted left 11 bits and then XORed to the left of the block..

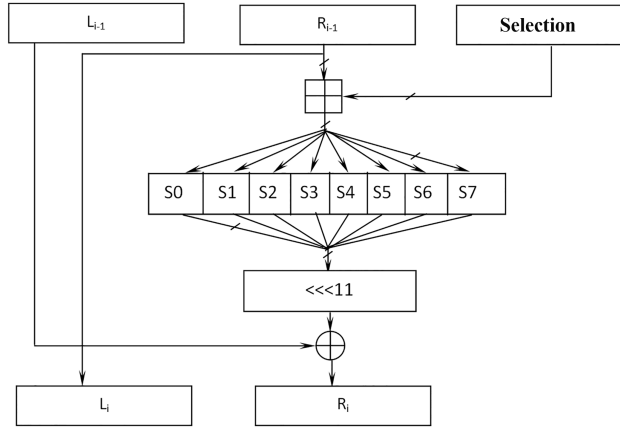


Figure 2.36 – Generating function of the GOST 28147-89 algorithm

Round keys are generated according to the following scheme: a 256-bit key is split into eight 32-bit machine words. They are numbered K0 through K7.

32 round keys are obtained by applying these machine words in the following order:

$$K_0, K_1, K_2,..,. K_7, K_0, K_1, K_2,...,K_7, K_0, K_1, K_2,\ldots, K_7, K_7, K_6, K_5,..., K_0, \quad (2.20)$$

that is, in the last 8 rounds, keys are served in reverse order.

The GOST algorithm is symmetric, and for decryption, it is enough to submit to the algorithm input blocks of encrypted messages and round keys in the reverse order of their encryption.

The peculiarity and, probably, the main drawback of the algorithm is the absence in the standard of any recommendations on the choice of the contents of the lookup tables (S-boxes).

Originally 8 * 16 * 4 = 512 bits of lookup tables were also part of the key information. Subsequently, the requirement for the secrecy of the contents of the tables was abolished, however, the static content of the lookup tables, which ensures a high cryptographic strength of the algorithm, was never published.

The set of S-boxes recommended by the GOST R34.11-94 hashing standard, which uses the GOST 28147-89 block cipher as the main transforming operation, is shown in Table 2.18.

Table 2.18 – S-blocks of the GOST 28147-89 algorithm

S0	4	10	9	2	13	8	0	14	6	11	1	12	7	15	5	3
S1	14	11	4	12	6	13	15	10	2	3	8	1	0	7	5	9
S2	5	8	1	13	10	3	4	2	14	15	12	7	6	0	9	11
S3	7	13	10	1	0	8	9	15	14	4	6	12	11	2	5	3
S4	6	12	7	1	5	15	13	8	4	10	9	14	0	3	11	2
S5	4	11	10	0	7	2	1	13	3	6	8	5	9	12	15	14
S6	13	11	4	1	3	15	5	9	0	10	14	7	6	8	2	12
S7	1	15	13	0	5	7	10	4	9	2	3	14	6	11	8	12

The advantages of the GOST algorithm can be called the simplicity of implementation, both software and hardware, since the algorithm does not use bit permutations, the large key size makes a force attack on the cipher unpromising, a large number of rounds make differential and linear cryptanalysis difficult. The algorithm is optimized for 32-bit processors.

The only problem in the practical application of the algorithm is, as already mentioned, the formation of lookup tables[48].

---

[48] In addition to the very large key size, GOST has a significantly lower execution cost compared to AES. In fact, it costs a lot less than AES, which requires four times as many hardware logic gates for much lower declared security.

It is not surprising that GOST has become an Internet standard, in particular, it is included in many crypto libraries such as OpenSSL and Crypto ++. In 2010, GOST was declared for ISO standardization as a worldwide encryption standard.

An extremely small number of algorithms have been able to become international standards. International standard ISO / IEC 18033-3: 2010 describes the following algorithms: four 64-bit ciphers – TDEA, MISTY1, CAST-128, HIGHT – and three 128-bit ciphers – AES, Camellia, SEED.

Analysis of the GOST cryptoalgorithm allows us to speak of its practical invulnerability for another 200 years (naturally, with the development of computing technology according to Moore's Law).

The work of block ciphers differs from stream ciphers by processing bits in groups, and not by a stream. At the same time, block ciphers are more reliable, but slower than stream ciphers.

The advantages of block ciphers include the similarity of encryption and decryption procedures, which, as a rule, differ only in the order of actions. This simplifies the creation of encryption devices, since it allows the use of the same blocks in the encryption and decryption chains. The flexibility of block ciphers allows them to be used to build other cryptographic primitives: pseudo-random sequence generator, stream cipher, impersonation and cryptographic hashes.

2.5 Stream Ciphers

A stream or stream cipher is a symmetric cipher in which each plaintext character is converted to a ciphertext character, depending not only on the key used, but also on its location in the plaintext stream.

Stream ciphers based on shift registers were actively used during the Second World War, long before the advent of electronics. They were easy to design and implement.

In 1965, Ernst Sejersted Selmer, the chief cryptographer of the Norwegian government, developed the shift register sequence theory. Later, Solomon Wolf Golomb, a mathematician of the US National Security Agency[49], wrote a book called «Shift Register Sequences», in which he outlined his main achievements in this area, as well as those of Selmer.

Since practically all data transmission channels for streaming encryption systems are interfered with, cryptanalysts are forced to solve the problem of synchronizing encryption and decryption of text to prevent loss of information. Therefore, according to the method of solving this problem, cipher systems are divided into synchronous and self-synchronized systems.

The encryption scheme using synchronous stream ciphers is shown in the figure 2.37.

[49] Solomon Wolf Colomb is an outstanding mathematician who made a significant contribution to the development of information technology in the 20th century. Also known as the inventor of the polyomino (generalized domino), which inspired the Russian programmer Alexei Pazhitnov to create the computer game «Tetris.

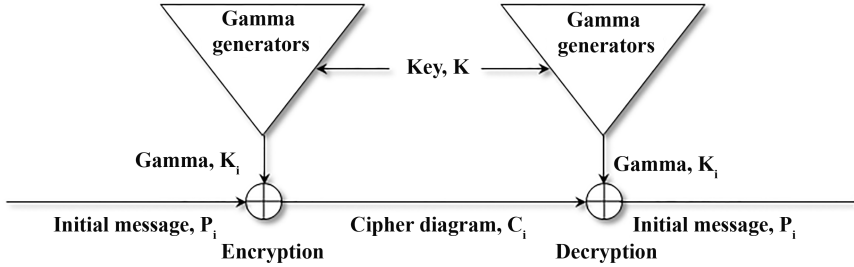


Figure 2.37 – Encryption scheme using a synchronous stream cipher

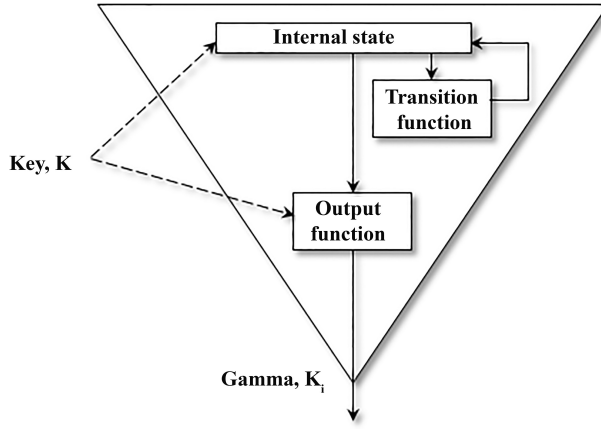


Figure 2.38 – Device of a gamma generator with internal feedback

In this case, the gamma generator, as a rule, consists of three main blocks (Figure 2.39).

The internal state describes the current state of the gamma generator. The initial internal state is usually determined by the key K. Two generators, with the same key and the same internal state, create the same gamma. The transition function reads the current internal state and generates a new internal state. The output function reads the internal state and generates the Ki gamma bit (s).

In another variation, the so-called counter type generators, there is no block with a transition function. Unlike feedback generators, they allow you to calculate the i-th bit of the gamma without calculating all the previous bits. To do this, the generator is set to the i-th internal state, after which the corresponding i-th bit of the gamma is calculated. This property is useful

for providing random access to data files, which allows you to decrypt a single piece of data without decrypting the entire file.

In a synchronous stream cipher, gamma is generated independently of the message stream. On the encryption side, the gamma generator sequentially outputs Ki gamma bits. On the decoding side, the other gamma generator outputs identical gamma bits one after the other. This scheme works fine if both generators are in sync.

The main disadvantage of synchronous streaming ciphers is that if one of the generators misses one of the cycles or a bit of the cipher is lost during transmission, then all cipher characters following the error are decoded incorrectly. In this case, the sender and the receiver must synchronize the generators and re-transmit the incorrectly decrypted part of the message.

In stream ciphers, if a bit of the ciphertext or key is distorted, then this only leads to the loss of one bit of the decrypted text. The situation becomes more problematic if the bit is not distorted, but lost – in this case, all information starting from this bit is distorted. In a self-synchronizing stream cipher, each gamma bit is a function of a fixed number of previous cipher bits. The gamma generators used in this encryption are called ciphertext feedback generators.

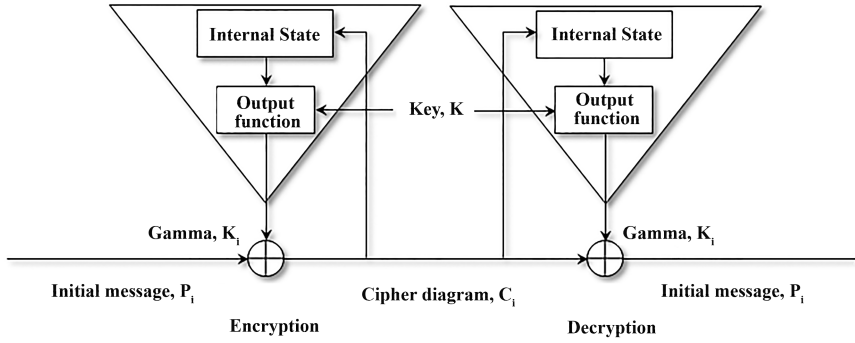


Figure 2.39 – Encryption scheme using self-synchronizing gamma generators

The internal state depends on the previous n bits of the cipher. Each message begins with a random header (initialization vector, sync burst) of length n bits, after passing through which both gamma generators are synchronized.

The disadvantages of self-synchronizing stream ciphers are as follows:

1) propagation of the error. For each bit of the cipher program corrupted during transmission, the decryption generator produces n incorrect gamma bits. Consequently, the changed bit affects the internal state, each cipher error will correspond to n plaintext errors;
2) when the C_i bit is lost, part of the message must be re-transmitted, but unlike synchronous stream ciphers, the synchronization of generators is much easier.

But unlike synchronous stream ciphers, self-synchronizing stream cipher generators may not work constantly, but only at the time of message transfer.

To date, a large number of stream encryption algorithms have been created, such as, for example, A3, A5, A8, MUGI, PIKE, RC4 and SEAL.

The RC4[50] algorithm (from the English Rivest cipher or Ron's code) is a synchronous stream cipher. The algorithm was created on the basis of the RC1-RC3 family of algorithms by RSA Security employee Ronald Rivest in 1987. For seven years the cipher was a trade secret, and the exact description of the algorithm was provided only after the signing of a nondisclosure agreement, but in September 1994 the description of the algorithm was anonymously sent to the Cypherpunks mailing list[51].

Figure 2.40 – Ernst Selmer (1920-2006), Solomon Golob (1932-2016) and Ron (Ronald) Rivest (1947-...)

The main advantages of the RC4 algorithm are high speed of operation, high cryptographic strength and variable key size. A typical implementation runs 19 machine instructions for every byte of text.

---

[50] The algorithm is also known as ARC4 or ARCFOUR.

[51] Since RC4 is a trademark of RSA Security and has not been officially published by the authors, in order to avoid claims from the trademark owner, the algorithm is called ARC4 or ARCFOUR (meaning the English alleged RC4 - "alleged" RC4).

The RC4 algorithm is based on a pseudo-random bit generator. A key is written to the generator input, and pseudo-random bits k are read at the output. The key length can be from 40 to 2048 bits[52]. The generated sequence bits are uniformly distributed.

Encryption is reduced to a gamma operation. The generated bit sequence is superimposed on the plain text by a modulo 2 addition operation (XOR operation). As a result, a cipher is learned c_i.

$$c_i = m_i \oplus k_i \tag{2.20}$$

Decrypting the text also comes down to two operations: generating a pseudo-random sequence of bits on the recipient's side and superimposing this sequence on the cipher, again by XOR.

$$m_i = c_i \oplus k_i = c_i \oplus k_i \oplus k_i \tag{2.21}$$

The main part of the algorithm is a pseudo-random sequence generator that is uniquely identified by the encryption key.

The operation algorithm of the pseudo-random bit sequence generator in RC4 consists of two stages:

- KSA[53] algorithm – the first and main stage in the generator is the implementation of initialization functions, which uses a variable-length key to create the initial state of the keystream generator. At this stage, the replacement table S is initialized.
- PRGA[54] algorithm – at this stage, pseudo-random numbers are calculated.

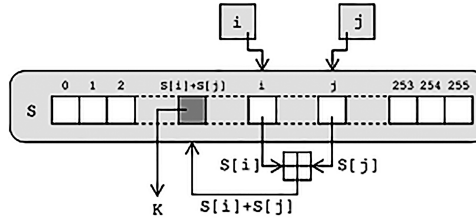


Figure 2.41 – RC4 key stream generator

The RC4 algorithm is actually a class of algorithms defined by its block size. This parameter n is the word size for the algorithm. Usually, n = 8, but

[52] In the United States, the nationally recommended key length is 128 bits.
[53] Key-scheduling algorithm.
[54] Pseudo-random generation algorithm.

it can be either decreased or increased. So, for n = 4 elements in the S-box 16, for n = 8 elements in the S-box 256. When n increases, say, up to 16 bits, the elements in the S-box become 65,536 and, accordingly, the initial iteration time will be increased. However, the encryption speed will also increase.

The array used as a table of substitutions, called an S-box, was designated as S. At each moment in time, table S contains all possible n-bit numbers in a shuffle form. The specific permutation of values in the table is determined by the key. Since each element of the table takes values in the range 0 to 255 (for n = 8), it can be interpreted in two ways: either as a number or as the number of another element in the table.

For example, for n = 8 S-box can be obtained by the algorithm below.

```
int keyLength = key.Length;
for (int i = 0; i < 256; i++)
{
S[i] = (byte)i;
}
int j = 0;
for (int i = 0; i < 256; i++)
{
j = (j + S[i] + key[i % keyLength]) % 256; S.Swap(i, j);
}
```

After table S has been prepared, you can start generating random n-bit words.

For example, for n = 8, a random 8-bit word can be obtained using the algorithm below.

```
x = (x + 1) % 256;
y = (y + S[x]) % 256;
S.Swap(x, y);
z = S[(S[x] + S[y]) % 256];
```

The resulting z value can be used as a key to encrypt the next block of the input data stream.

The RC4 encryption algorithm is used in several widely used encryption standards and protocols (for example, WEP, WPA, SSL, and TLS).

It should be noted that many vulnerabilities have been discovered in the RC4 algorithm. They are primarily associated with the use of non-random or associated keys and the situation when one key stream is reused. For example, such a problem made the WEP[55] protocol cryptographically insecure.

As of 2018, there is speculation that some government cryptographic agencies may have the ability to break RC4 when used in the TLS protocol. The IETF published RFC 7465 to prohibit the use of RC4 in TLS; Mozilla and Microsoft have issued similar guidelines.

Several attempts have been made to strengthen the RC4 algorithm. Such algorithms were, in particular, Spritz, RC4A, VMPC, и RC4+.

Questions for Self-Control

1. What is a symmetric cryptosystem?
2. What is a cryptographic key in a cryptosystem for?
3. Can block ciphers be used as stream ciphers?
4. What are the basic transformation operations underlying symmetric cryptoalgorithms?
5. What is a Feistel network?
6. What are permutation networks used for?
7. What algorithms are developed based on the AES cryptoalgorithm?
8. What is considered the weak point of the GOST 28147-89 cryptographic algorithm?
9. How many bits of information will be lost when decrypting the cipher code if one of the penultimate bit of the cipher code is lost in a self-synchronizing stream cipher?
10. What are the main vulnerabilities of the WEP protocol?

Recommended Reading

1. A.P.Alferov, A.Yu.Zubov, A.S.Kuzmin, A.V.Cheryomushkin. Fundamentals of Cryptography. – Helios ARV, 2002.

[55] All attacks against WEP are based on flaws in the RC4 cipher, such as the possibility of collision vectors of initialization and frame changes. All types of attacks require interception and analysis of wireless frames. Depending on the type of attack, the number of frames required for breaking is different. With programs such as Aircrack-ng, hacking a WEP encrypted wireless network is very fast and requires no special skills.

2. Kahn D. The Codebreakers: The Story of Secret Writing. – Macmillan, 1967.
3. A.V. Babash, G.P.Shankin. Cryptography. – M. SOLON-PRESS, 2007.
4. Fred B. Rickson. Codes, ciphers, signals and secret transmission of information. – Astrel, 2011.
5. Fomichev V.M. Discrete mathematics and cryptology: A course of lectures / ed. ND Podufalov. – M.: Dialog-MEPhI, 2013.
6. Zhelnikov V. Cryptography from papyrus to computer. – M.: ABF, 1996.
7. Gabidulin E.M., Kshevetskiy A.S., Kolybelnikov A.I. Information security: a tutorial. – M.: MFTI, 2011.
8. Chris Christensen. Lester Hill Revisited // Taylor & Francis Group, LLC: Article. 2014.
9. Mao V. Modern cryptography: Theory and practice. – M.: Williams, 2005.
10. V.N. Krishna, Dr. A. Vinaya Babu. A Modified Hill Cipher Algorithm for Encryption of Data In Data Transmission (English) // Computer Science and Telecommunications: Georgian Electronic Scientific Journal. 2007.
11. Federal Information Processing Standards Publication 197 November 26, 2001 Specification for the ADVANCED ENCRYPTION STANDARD (AES)
12. Barichev S.G., Goncharov V.V., Serov R.E. 2.4.2. AES standard. Algorithm Rijdael // Fundamentals of modern cryptography. – 3rd ed. – M.: Dialog-MI-FI, 2011.
13. GOST 28147-89 «Information processing systems. Cryptographic protection. Cryptographic transformation algorithm».

CHAPTER 3. OPEN (PUBLIC) KEY CRYPTOGRAPHY SYSTEMS

Keywords: asymmetric cryptosystem, one-way function, one-way function with secret, public or public key, secret key, inverted function, hash, hashing, elliptic curve, Diffie-Hellman algorithm, key and keyless hash function, cryptographic strength, collision, irreversible function, digital signature, qualified and simple EDS.

3.1 One-way Functions and Hook Functions

The XX century was marked by the emergence of a new type of cryptographic systems – systems without problems of transferring the encryption key between the sender and the recipient of cryptographically encrypted messages. These systems are *encryption algorithms with a public (or, more correctly, a public) key.*

These cryptographic transformations are based on the idea proposed by W. Diffy and M. Helman about separating the encryption key and the message decryption key. If we can ensure the impossibility of obtaining the decryption key from the encryption key, then it becomes possible to build sufficiently reliable cryptographic systems and there is no need to keep the encryption key secret. Accordingly, it is possible to make your encryption key public, so that everyone interested in it can encrypt a message for you (hence the name – public or public key). However, the message can only be decrypted by the owner of the key, who has its other half – the secret key for decryption.

The concept of one-way function[1], introduced in 1975 by Diffie and Hellman, is the basis for understanding the obtaining of the private and public keys.

One-sided is a mathematical function FK(X) → Y that has two unique properties:

– there is a polynomial algorithm for calculating the values of $F_K(x)$;

[1] Despite many years of work by mathematicians, a true one-way function has never been found. However, some of the properties of this hypothetical function are quite successfully used in cryptography. For example, such a property as the complexity (more precisely, the complexity classes) of calculating the direct and inverted functions. And if the complexity of calculating the inverted function becomes greater than modern computational capabilities, then it can be applied as a conditionally one-sided.

– there is no polynomial algorithm for inverting the function F, i.e. solutions of the equation F(X) = Y with respect to X.

The function is essential for understanding the process. However, in this form, this function was not used. Another concept, closer to the concepts used in traditional cryptography, is the concept of a one-way function with a secret[2].

A one-way function with a secret K is a function $F_K(X) \rightarrow Y$ that depends on the parameter K and has three properties:

– for any K there is a polynomial algorithm for calculating the values of $F_K(X)$;
– for unknown K, there is no polynomial inversion algorithm F_K;
– for a known K, there is a polynomial inversion algorithm F_K.

It is the secret K that acts as a decryption key.

There are many functions built for cryptography that can be considered one-way with a secret. This means that for them the second property has not yet been rigorously proven, but it is known that the inversion problem is equivalent to some difficult mathematical problem and cannot be solved at the modern technical level. It is also worth noting that for some of these functions, mathematicians have already found inverted functions and their application in cryptography does not provide information security.

Recently, such cryptosystems have also come to be called asymmetric, since they have asymmetries in encryption and decryption algorithms. In contrast to such systems, traditional ciphers with one secret key came to be called symmetric. For asymmetric systems, the encryption algorithm is well known, but it is impossible to recover the decryption algorithm from it in polynomial time.

3.2 Cryptosystem RSA

Whitfield Diffie and Martin Hellman are rightfully named winners of the Turing Prize for the development of asymmetric encryption algorithms. However, in their 1975 algorithm, they did not offer a one-way function that is convenient to implement.

This was done in 1977 by a trio of perhaps the most famous mathematicians of the twentieth century: Ronald Rivest, Eddie Shamir and Leonard Adleman of the Massachusetts Institute of Technology. The system proposed by them, based on the function of calculating the remainder of

[2] Sometimes the terms function with a trap, function of a sliding door (English name: one-way trap-door function) are also used.

an integer division, turned out to be extremely practical, and also became widespread under the name «RSA system» - after the first English letters of the authors' surnames.

The algorithm is based on two keys: public and secret. To obtain them, you must perform the following steps:

- take two large primes p and q;
- determine n as the result of multiplication p on q ($n = p \cdot q$);
- choose a random number, which we will call d. This number must be relatively prime (not have any common divisor, except 1) with the multiplication result $(p-1) \cdot (q-1)$;
- determine such a number e for which the following relation is true $(e \cdot d) \bmod ((p-1) \cdot (q-1)) = 1$.

Further numbers e and n are accepted as public or public key. Accordingly, the numbers d and n are accepted as the secret key.

In order to encrypt data using the public key {e, n}, you need the following:

- split the encrypted text into blocks, each of which can be represented as a number M (i) = 0,1,2 ..., n-1 (i.e. only up to n-1).
- encrypt the text, considered as a sequence of numbers M (i) according to the formula

$$C(i) = M(i)^e \bmod n. \quad (3.1)$$

To decrypt this data using the secret key {d, n}, you need to perform the following calculations:

$$M(i) = C(i)^d \bmod n. \quad (3.2)$$

As a result, a set of numbers M(i) will be obtained, which represent the original text.

Example: let's encrypt and decrypt the message «CAB» using the RSA algorithm. For simplicity, let's take small numbers - this will make the calculations simpler and clearer..

1) Choose p = 3 and q = 11.

2) Define $n = 3 \cdot 11 = 33$.

3) Find $(p-1) \cdot (q-1) = 20$. Therefore, d will be equal, for example, 3: (d = 3).

4) Let's choose the number e according to the following formula: $(e \cdot 3) \bmod 20 = 1$. So e will be equal, for example, 7: (e = 7).

5) Let's imagine the encrypted message as a sequence of numbers in the range from 0 to 32 (remember that it ends in n-1). Letter A = 1, B = 2, C = 3.

We simulate the work of the sender of the message and encrypt the message using the public key {7,33}

C1 = (37) mod 33 = 2187 mod 33 = 9;

C2 = (17) mod 33 = 1 mod 33 = 1;

C3 = (27) mod 33 = 128 mod 33 = 29;

Next, we simulate the work of the recipient of the message and decrypt the data using the private key {3,33}.

M1 = (93) mod 33 = 729 mod 33 = 3 (C);

M2 = (13) mod 33 = 1 mod 33 = 1 (A);

M3 = (293) mod 33 = 24389 mod 33 = 2 (B).

In fact, the described encryption method is very weak and is never used. The reason is simple – it's just a mono-alphabetical substitution - the same letter will be encrypted with the same number. People learned to break such ciphers in the last millennium. A cryptanalyst does not even need to figure out the keys – he decrypts the message without even knowing about the keys.

The most commonly used currently is the mixed encryption algorithm, in which the session key is first encrypted with the RSA algorithm, and then the participants use it to encrypt their messages with symmetric systems, for example AES. After the session ends, the session key is usually destroyed.

In August 1977, Martin Gardner's Math Games column in Scientific American, with permission from Ronald Rivest, published the first description of the RSA cryptosystem. Readers were also asked to decrypt the English phrase encrypted with the described algorithm:

9686	9613	7546	2206
1477	1409	2225	4355
8829	0575	9991	1245
7431	9874	6951	2093
0816	2982	2514	5708
3569	3147	6622	8839
8962	8013	3919	9055
1829	9451	5781	5154

Figure 3.1 – An example of Ron Rivest's cipher

Ron Rivest himself believed that this message, encrypted with a 425-bit key, could be decrypted for 40 quadrillion years[3] . Ron Rivest offered $ 100 for decrypting the message. However, just 15 years later, this message was deciphered by a group of 600 enthusiasts on 1600 computers within only six months[4] .

The key size in the RSA algorithm is related to the unit size n. Two numbers p and q, the product of which is the modulus, must have approximately the same length, since in this case it is more difficult to find the factors (factors) than in the case when the length of the numbers is significantly different. However, if two numbers are extremely close to each other or their difference is close to some predetermined value, then there is a potential security threat, but this probability - the proximity of two randomly selected numbers – is negligible.

The optimal module size is determined by the security requirements: a larger module provides more security, but also slows down the RSA algorithm.

The length of the module is selected primarily based on the importance of the protected data and the required durability of the protected data and, secondly, on the basis of an assessment of possible threats.

Back in 1997, an estimate showed that a 512-bit RSA key can be cracked (by factoring) for $ 1 million and eight months of operation. In 1999, a 512-bit key was cracked in seven months, which means that 512-bit keys no longer provide sufficient security, except for very short-term tasks.

Currently, RSA Lab recommends 1024-bit keys for general tasks and 2048 bits for mission-critical tasks.

It should also be noted that the key sizes in the RSA cryptosystem (as well as in other public key cryptosystems) are much larger than the key sizes of symmetric encryption systems. However, the strength of the RSA key is much less than the strength of a key of the same length in a symmetric encryption system.

The RSA algorithm is perhaps one of the most widely used cryptoalgorithms at the moment and is used in a large number of cryptographic applications, including PGP, S / MIME, TLS / SSL, IPSEC / IKE and many others.

---

[3] Analysts, however, believed that it took only 20,000 years to decrypt a message on computers in those years.

[4] This whole story was a great publicity stunt for Rivest, Adleman and Shamir, who patented RSA, and as a result, received $ 900 million in profit.

3.3 Cryptosystems Based on Elliptic Curves

Elliptic curves are a fairly well-studied mathematical apparatus. The oldest surviving source, in which such curves are considered, is the «Arithmetic» of the ancient Greek mathematician Diophantus. However, until the end of the XX century, they were not of practical value. Everything changed in 1985.

In 1985, independently professor of mathematics at the University of Washington Neal Koblitz and mathematician at the Princeton Institute IDA Research Center Victor Saul Miller proposed the use of algebraic properties of elliptic curves in cryptography. From this moment, the rapid development of a new direction of cryptography began, for which the term «cryptography on elliptic curves» is used.

The Koblinz-Miller discovery helped to create elliptical variants of the Diffie-Hellman, El-Gamal, MQV, DSS, GOST R 34.10-94 algorithms, which initially used the multiplicative group of a finite field. As a result, new algorithms (with the exception of GOST) received the EC or ECC prefix – Elliptic Curve Cryptography: ECDH, EC ElGamal, ECMQV, ECDSS, and the Russian GOST R 34.10-94 was transformed into GOST R 34.10-2001 (and then into the more reliable 34.10-2012).

In cryptography on elliptic curves, the role of the main cryptographic operation is performed by the operation of scalar multiplication of a point on an elliptic curve by a given integer, defined through the operations of adding and doubling points of an elliptic curve. The latter, in turn, are performed on the basis of addition, multiplication, and inversion operations in the finite field over which the curve is considered. Particular interest in elliptic curve cryptography is due to the advantages that its use in wireless communications gives – high speed and short key length.

Figure 3.2 – Neil Koblitz (1948) and Victor Miller (1947)

As the name suggests, elliptic curve cryptography is based on elliptic curves. Elliptical curves are not ellipses. They are called so simply because they are described by cubic equations, similar to those used to calculate the curve and ellipse.

In general, cubic equations for elliptic curves over finite fields have the form:

$$y^2 + axy + by = x^3 + cx^2 + dx + e, \tag{3.3}$$

where a, b, c, d and e are real numbers satisfying some simple conditions.

This equation E can be considered over arbitrary fields and, in particular, over finite fields F, which are of particular interest for cryptography.

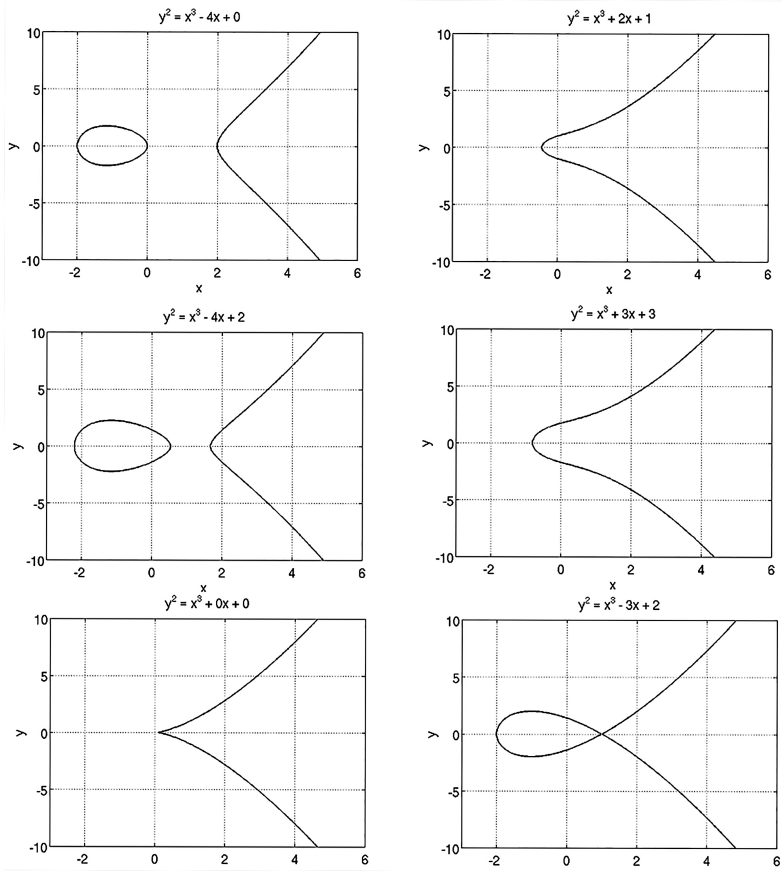


Figure 3.3 – Examples of graphs of elliptic curves

The elliptic curves shown in the first 4 figures are called non-singular or smooth. While the bottom two curves refer to the so-called singular elliptic curves.

The equation of an elliptic curve is given by the formula

$$y^2 = x^3 + ax + b \tag{3.4}$$

or

$$y = \pm\sqrt{x^3 + ax + b}.$$

The graph of this curve is symmetrical about the abscissa axis. The points of its intersection with the axis are the solution to the cubic equation

$$x^3 + ax + b = 0 \tag{3.5}$$

The discriminant of this equation $B = \left(\frac{a}{3}\right)^3 + \left(\frac{b}{2}\right)^2$.

There are three possible solutions to this situation:

– if D < 0, then the equation has three different real roots. Typical graphs are graphs 1 and 3 in Figure 3.2;
– if D = 0, then the equation has three roots, two of which are the same. In this case, there is a singular point and a singular curve. Typical graphs will be graphs 5 and 6 in Figure 3.2;
– if D > 0, then the equation has one real root and two complex ones. Typical graphs are 2 and 4 in Figure 3.2.

That is, the curve will be nonsingular provided that its discriminant is not equal to 0, which in turn is equivalent to the following expression:

$$4a^3 + 27b^2 \neq 0 \tag{3.6}$$

For smooth curves, any straight line passing through two different points on the curve intersects it (the curve) at a single point. In addition, only one tangent line can be drawn to any point on the curve.

Such properties of the curve allow you to specify a group operation called the addition of points of an elliptic curve. So the addition of two points can be represented graphically (Figure 3.4).

As can be seen from the figure, to add points P and Q, it is necessary to draw a straight line between them, which will necessarily intersect the curve at some third point R. This point R will be called the sum of two points P and Q.

Reflect point R relative to the horizontal coordinate axis and get the desired point P + Q.

To find the coordinates of the point P + Q, use the expression 3.7.

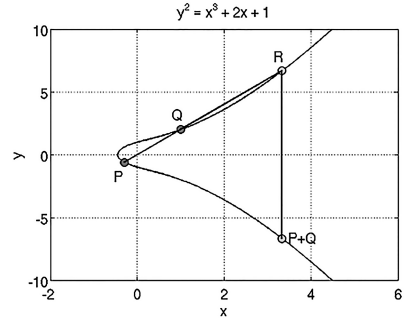


Figure 3.4 – An example of obtaining points P and Q on an elliptic curve $y^2 = x^3+2x+1$

$$x_{P+Q} = \alpha^2 - x_P - x_Q$$
$$y_{P+Q} = -y_P + \alpha(x_P - x_R), \qquad (3.7)$$

where $\alpha = (y_Q - y_P)/(x_Q - x_P)$.

The definition of an elliptic curve also includes an element denoted by O and called the «improper element»[5].

There is only one nuance left to clarify. All curves considered above are elliptic curves over real numbers. And this leads to the need for rounding and the problems it generates – using curves over real numbers, it is impossible to get a bijection between the original text and the encrypted data. To solve the rounding problem in cryptography, only curves over finite fields are used. This means that an elliptic curve is understood as a set of points whose coordinates belong to a finite field.

For singular curves, the solution to the inverse problem is much easier than for smooth curves. Therefore, their use in cryptography is highly undesirable. Because the wrong choice of an elliptic curve can lead to a decrease in the achieved level of security, standards organizations identify whole blocks of curves that have the necessary reliability. The use of standardized curves is also recommended because better compatibility between different implementations of information security protocols becomes possible.

[5] Also this element is sometimes called (“infinite element”, “zero element”, “point at infinity”).

There are two types of elliptic curves considered in cryptography:

– over a finite field Z_p;
– over the field GF (2^m).

Elliptic curves over the GF (2^m) field have one important advantage, the elements of the GF (2^m) field can be easily represented in the form of n-bit codewords, this allows increasing the speed of the hardware implementation of elliptic algorithms.

All mathematical operations on elliptic curves over a finite field are performed according to the laws of a finite field, over which an elliptic curve is constructed. Those. to calculate, for example, the sum of two points of the curve E over the residue ring Z_p, all operations are performed modulo the number p.

Another important concept of elliptic cryptography is the order of an elliptic curve, which shows the number of points on a curve over a finite field.

In the case of cryptography using elliptic curves, one has to deal with a reduced form of an elliptic curve, which is defined over a finite field. Of particular interest to cryptography is an object called an elliptic group mod p, where p is a prime number. An elliptic curve over a finite field is given by the equation

$$y^2 = x^3 + ax + b \pmod{p}. \tag{3.8}$$

The main arithmetic operation in elliptic cryptography is the operation of scalar multiplication of points on a curve, which allows you to determine the point Q

$$Q = k \bullet P \tag{3.9}$$

Scalar multiplication is done through several combinations of adding and doubling the points of an elliptic curve. For example, point $11 \bullet P$ can be represented as $11 \bullet P = 2 \bullet (2 \bullet (2 \bullet P) + P) + P$.

In addition to the equation, an important parameter of the curve is the base (generating) point G, which is selected for each curve separately.

The secret key in accordance with elliptic cryptography technology is a large random number k, and the reported public key is the product of k and the base point G.

Key generation and exchange in elliptic cryptography can be performed using the Diffie-Hellman scheme similar to the RSA algorithm. First, a large prime p and the parameters of the curve equation are selected. This specifies the group of points at which the base point G is selected. When choosing G, it is important that the smallest value of n at which $nG = O$ is

a very large prime. The equation of the curve and the point G are known to all participants in the process. The exchange of keys between the recipient and senders of the message can be carried out as follows:

1) participant A chooses an integer nA less than n. This number will be his private secret key. Then participant A generates a public key PA = nA¦G. A public key is a point on a curve;
2) in the same way, participant B chooses a private key nB and calculates a public key PB.
3) Participant A generates a secret key

$$K = nA \times PБ, \tag{3.10}$$

and participant B generates a secret key

$$K = nБ \times PA. \tag{3.11}$$

Secret key formulas give the same result:

$$nA \times PБ = nA \times (nБ \times G) = nБ \times (nA \times G) = nБ \times PA. \tag{3.12}$$

To crack this scheme, the adversary will have to solve the discrete logarithm problem on the curve, which is supposed to be an intractable problem.

Example: generate a shared key for two users using Diffie-Hellman scheme if an elliptical curve is selected E_{211} 0,-4) => $y^2 = (x^3 - 4) \bmod 211$ and point P (2, 2).

Then $y = \sqrt{x3-4} \bmod 211$. The result of calculating the square root modulo p will be two numbers y and p-y.

Then let user A choose a secret key with =121. Then user A calculates 121P = 121 (2, 2) = (115, 48). User A gives user B his public key - point (115, 48).

Let user B choose the secret key d = 203. Then user B calculates 203P = 203 (2, 2) = (130, 203). User B gives user A his public key – point (130, 203).

Next, users A and B perform the following calculations:

– user A calculates 121 (130, 203) = (161, 169);

– user B calculates 203 (115, 48) = (161, 169).

The shared secret will be dot (161, 169). Its values or values of any function can be further used as a secret key for symmetric encryption. For example, you can take the abscissa as a secret key $161_{10} = 10100001_2$.

The reliability and cryptographic strength of elliptic cryptography is based on the difficulty of solving the discrete logarithm problem on the Elliptic Curve Discrete Logarithm Problem (ECDLP), the essence of which is to find an integer k from the known points P and Q.

When using algorithms on elliptic curves, it is assumed that there are no subexponential algorithms for solving the discrete logarithm problem in groups of their points. Moreover, the order of the group of points of the elliptic curve determines the complexity of the problem. It is believed that to achieve the same level of cryptographic strength as in RSA, groups of smaller orders are required, which reduces the cost of storing and transferring information.

For example, at the RSA 2005 conference, the US National Security Agency announced the creation of «Suite B» («Set B»), which uses exclusively elliptic cryptography algorithms, and only 384-bit are used to protect information classified before «Top Secret» keys (for example, RSA recommends a key with a length of at least 2048 bits)[6]. All basic algorithms and protocols of «Suite B» were built on the basis of cryptography on elliptic curves, and for RSA the auxiliary role of the «first generation» was assigned, which was necessary only for a smooth transition to a new, more efficient so-called quantum-safe cryptography[7] .

3.4 Hash Functions and Hashing

Another important aspect of the application of cryptographic transformations is *hashing*, or *the computation of hash functions*. Hash functions have become the basis of modern authentication, EDS, blockchain technology and much more.

This term appeared in the middle of the last century among specialists involved in processing data arrays. So in 1953 Hans Peter Luhn proposed using «hash encoding» in IBM programs. In 1956, Arnold Dumey described the idea of «hashing» as it is used today. In 1957, the IBM Journal of Research and Development published an article by W. Wesley Peterson

[6] In fact, crypto mathematicians estimate that 256-bit ECC operations are equivalent to working with a 3072-bit modulus in RSA. And 160-bit and 224-bit ECC keys provide the same security levels as 1024-bit and 2048-bit RSA keys, respectively.

[7] Under this general term (in another version it also sounds like post-quantum cryptography) in the field of information security, it is customary to understand a wide range of all kinds of algorithms, protocols and communication devices capable of resisting threats from quantum computers.

on finding text in large files. This work is considered the first «serious» work on «hashing». Werner Buchholz published an extensive study of hash functions in 1963, and in 1968 Robert Morris[8] introduced the concept of hashing into scientific circulation with his review of hashing.

Initially, the word «hash» comes from the English «hash», one of the meanings of which is interpreted as «hash» or «confusion». Actually, this definition quite fully describes the real meaning of this term. It is often said about such a process that the process of «hashing» is taking place, which again is derived from the English hashing («chop», «chop», «confuse», etc.).

Figure 3.5 – Hans Peter Lohn (1896-1964), Wesley Peterson (1924-2009)

Currently, *hashing is the transformation of an array of input data of arbitrary length into an output bit string of a fixed length, performed by a certain algorithm.*

The function that implements the algorithm and performs the transformation is called a *hash function* or a *convolution function*. The original data is called an «input array», «key» or «message».

The transformation result (output) is called «hash», «hash code», «hash sum», «message summary».

A hash function is any function h: $X \rightarrow Y$ that is easily computable and such that for any message M the value h (M) = H (convolution) has a fixed bit length. Where X is the set of all messages, Y is the set of binary vectors of fixed length.

There are two important types of cryptographic hash functions – key and keyless.

[8] American Cryptographer (1932-2011). Not to be confused with Morris, Robert Tappan (born 1965) is an associate professor at the Massachusetts Institute of Technology; better known as the creator of the first network worm.

Keyless hash functions are called error detection codes. They make it possible with the help of additional means (encryption, for example) to guarantee the integrity of the data. The main property of the keyless hash function used was its sensitivity. This means that when the input data array changes at least one bit, the value of the hash function will also change.

This property of simple (not reliable, but easily calculated) hash functions has found application in controlling the integrity of transmitted data. For example, such a hash function is the function of calculating the checksum (CRC), which is used both when storing data arrays on media and when transmitting data in computer networks to detect hardware errors and failures – the so-called redundant coding). If the calculated hash value matches the one stored with the file or sent with the packet (the so-called checksum), then there was no loss during storage or transmission and the file / packet can be used.

A similar scheme is used in blockchain technology, where the hash acts as a guarantee of the integrity of the transaction (payment) chain and protects transactions from unauthorized changes. Hash and distributed computing make it almost impossible to hack the blockchain. Almost all cryptocurrencies have been developed on blockchain technology, for example, bitcoin.

The main condition for cryptographic hash functions is the impossibility of calculating the initial data array from the final result (hash). The second main condition of such hash functions is collision resistance, that is, a low probability of obtaining two identical hash sums from two different data arrays when processed by this function[9] . Calculations using such algorithms are more complicated, but the main factor here is not speed, but reliability.

Hashing is also used in the construction of an electronic digital signature as a tool that reduces the computational complexity of setting and verifying a signature, as well as its volume.

There are many hashing algorithms with different properties. The most frequently used properties are the bit width of the hash function, computational complexity and cryptographic strength. The choice of one or another hash function is determined by the specifics of the problem it solves.

The following requirements are imposed on keyless functions:

– unidirectionality;

[9] Also, the phenomenon of collisions when obtaining a hash is called the «twins effect» or «birthday effect».

– resistance to collisions;
– resistance to finding the second preimage.

Unidirectionality or irreversibility is understood as the high complexity of finding a message by a given convolution value. It should be noted that the existence of irreversible hash functions has not been proven, for which the calculation of any preimage of a given hash function value is theoretically impossible. Finding the reciprocal is usually only a computationally difficult task.

Collision resistance refers to the difficulty of finding a pair of messages with the same convolution values. Usually it is the cryptanalysts' finding of a method for constructing collisions that serves as the first signal that the algorithm is outdated and the need to replace it soon.

The resistance to finding the second preimage is understood as the complexity of finding a second message with the same convolution value for a given message with a known convolution value.

Among the family of keyless hashing algorithms, the most famous are the CRC16/32 algorithms, the MD2/4/5/6 family of algorithms, and the SHA family of algorithms.

Algorithms CRC16/32 or Cyclic Redundancy Code are designed to check data integrity and are used in error-correcting coding. The CRC algorithm is based on the properties of remainder division of binary polynomials, that is, polynomials over a finite field GF (2). The CRC value is the remainder of the division of the polynomial corresponding to the input data by some fixed generator polynomial.

To find the CRC from the file, the first word is taken. If the most significant bit in the word is «1», then the word is shifted to the left by one bit, followed by the XOR operation with the generating polynomial. Accordingly, if the most significant bit in the word is «0», then after the shift, the XOR operation is not performed. After the shift, the old MSB is lost, and the LSB is freed - its value is set to zero. In place of the least significant bit, the next bit from the file is loaded, and the operation is repeated until the last bit of the file is loaded. After passing through the entire file, the remainder remains in the word, which is the checksum. As already mentioned, the algorithm is not a cryptographic transformation and is only used to verify the integrity of the data.

The MD1/2/3/4/5/6 algorithms are the creation of Ronald Rivest, one of the authors of the RSA algorithm. The MD1 algorithm was the first in this line of algorithms, but its specification has never been published. The MD2 algorithm was developed by Rivest in 1989 for use as one of the

cryptographic algorithms included in the Privacy-Enhanced Mail (PEM) standard for secure e-mail. Its C implementation was provided in RFC 1115. And in 1990 MD2 was proposed as a replacement for BMAC (Bidirectional MAC). Subsequently, the specification and updated implementation of MD2 was published in RFC 1319. In 2011, MD2 was officially decommissioned due to many successful crypto attacks.

The MD3 algorithm has never been published. Apparently, the development of MD3 was abandoned. After MD2, MD4, MD5 and MD6 were developed in 1990, 1991 and 2008, respectively.

Ronald Rivest noted that MD4 was created primarily as a very fast hashing algorithm, so it can be bad in terms of cryptographic strength. As subsequent studies showed, he was right, and for applications where cryptographic strength is important, the MD5 algorithm was used. The vulnerabilities of the MD4 algorithm were demonstrated in an article by Bert den Boer and Anton Bosselars already in 1991. The first collision was found by Hans Dobbertin six years after publication in 1996.

The 128-bit MD5 hashing algorithm was once very popular, but the first prerequisites for hacking appeared in the late nineties, at the beginning of the XXI, a number of cryptanalysts demonstrated successful fast ways to find collisions. At the end of 2008, US-CERT urged software developers, website owners and users to stop using MD5 for any purpose, as research showed the unreliability of this algorithm, and in 2011 the algorithm was officially recognized as insecure and it was recommended to abandon its use.

The MD6 (Message Digest 6) algorithm is designed to receive message digests of arbitrary length. The algorithm was nominated for the SHA-3 competition, but, unfortunately, Rivest did not manage to bring it up to standard and withdrew it from the second stage of the competition[10].

In 2009, Rivest published this algorithm with corrections of the identified errors. Despite the fixes, the algorithm remains quite slow and loses in speed to the SHA[11] algorithms.

The SHA line of algorithms were developed by NIST. In 1993, the NSA partnered with NIST to develop a secure hashing algorithm now known as

[10] Bruce Schneier commented: “This is the first-class self-rejection case that we would expect from Ron Riveist, especially given the fact that there have been no attacks on the algorithm, while other algorithms are susceptible to serious attacks, but their authors continue to pretend, that nobody pays attention to it. «

[11] MD6-512 is one and a half times slower than SHA2-512 on 32-bit platforms and almost four times slower on 64-bit platforms.

SHA-0. However, the algorithm was soon withdrawn by the developers and in 1998, a 120-bit SHA-1 hashing algorithm of messages of arbitrary length was proposed to replace it. At the present time, many successful theoretical and practical attacks have been carried out on the algorithm[12]. Therefore, a number of companies abandoned the use of this algorithm: for example, Google abandoned it in 2014, and Yandex in 2016.

In 2002, the NSA and NIST published a new version of the hashing algorithm called SHA-2. More precisely, SHA-2 is the collective name for the SHA224, SHA256, SHA384, and SHA512 algorithms. In March 2012, the latest version of the algorithm was released. At the moment, the vulnerabilities found by Indian researchers Somitra Kumar Sanadia and Broadsword Sarkar are known for the algorithm. However, due to the algorithmic similarity of SHA-2 to SHA-1 and the presence of potential vulnerabilities in the latter, it was decided that SHA-3 will be based on a completely different algorithm. Accordingly, on October 2, 2012, NIST approved the Keccak algorithm as SHA-3, developed by a group of authors led by Ioan Dyman.

In the Russian Federation, in 1994, the 256-bit Russian standard - GOST 34.11-94 was recommended for obtaining hash functions for cryptography. In 2008, a team of experts from Austria and Poland discovered a technical vulnerability that reduced the search for collisions by 223 times. Thus, the number of operations required to find a collision is 2105, which, however, is practically not feasible at the moment. In 2012, the algorithm was outdated and decommissioned. Since January 1, 2013 it has been replaced by GOST R34.11-2012 «Stribog».

There are also key hash functions that do not use any basis such as block ciphers or keyless hash calculations, but are developed independently, taking into account the effective implementation on modern computers. For example, the key hash function used in the Message Authenticator Algorithm (MAA), as approved by the ISO 8731-2 standard.

Key hash functions are called message authentication codes. They make it possible, without additional means, to guarantee both the correctness of the data source and the integrity of the data in systems with users who trust each other.

[12] February 23, 2017 experts from Google and CWI announced a practical hacking of the algorithm.

3.5 Digital Signature

The development of the Internet has created many opportunities. One of such opportunities is *electronic document management*. Currently, to carry out legal transactions, there is no need to use paper media, edit and sign them, send them to each other and be afraid of their loss or theft. Electronic storage of information and means of data transmission have solved these issues. However, the situation was more complicated with the signing of these documents and their protection from forgeries and modifications.

Everyone probably remembers the riot of fraudulent documents after the appearance of color printers, which made it possible to print any previously scanned signature or seal with high quality. This even led to legislative requirements for stamping

«Copy» on color photocopies of documents, although, of course, this measure absolutely does not protect against these types of fraud. The situation became even more complicated with the appearance and a sharp decrease in the cost of laser engraving machines, which could make a copy of any seal or signature cliche within a minute for ridiculous money.

But these are the problems of paper media in which information is physically attached to the media - paper. In this case, at least some physical traces of document manipulation may remain. Dealing with electronic documents is more complicated. Here information is the document. In this case, there are also problems of trust between the participants in the processes, there are frauds of forgery, alteration, desertion and reuse of documents. To solve these problems of electronic document management, it was proposed to use mechanisms of electronic digital signature (EDS).

At present, *an EDS is a requisite of an electronic document obtained as a result of cryptographic transformation of information using a private or secret signature key.* EDS as a mechanism allows you to check the absence of distortion of information in an electronic document from the moment the signature was formed (integrity), whether the signature belongs to the owner of the signature key certificate (authorship), and in case of successful verification, confirm the fact of signing the electronic document (non-repudiation).

The digital signature is based on asymmetric encryption mechanisms. Therefore, naturally, its development began with the work of Diffie-Hellman in 1976 and the appearance of the Rivest-Shamir-Adleman RSA system in 1977. The RSA system became the first mechanism for setting and verifying a digital signature.

In the future, significant milestones in the development of EDS methods were:

- 1981 – the DSA algorithm was developed, which is still used as the US standard for electronic signature;
- 1984 – the creation of the El Gamal cryptosystem and the forumization by Shafi Goldwasser, Silvio Micali and Ronald Rivest of the security requirements for digital signature algorithms. They also described the models of attacks on EDS algorithms;
- 1991 – the DSS (Digital Signature Standard) standard for electronic signature was published, the developer of which was the US National Institute of Standardization and Technology (NIST). In the same year, a law on electronic digital signature was developed in the Russian Federation (although it will be adopted only in 2001);
- 1997 – the law on digital signature was adopted in Germany;
- 2003 – laws on digital signatures were adopted in the Republic of Kazakhstan and Ukraine.

From the point of view of legislation, at present, in many countries, the concepts of unqualified electronic signature and simple electronic signature have been introduced.

A qualified signature differs from an unqualified one in that a qualified signature is issued by an accredited certification center, and an unqualified signature is issued by a non-accredited center.

Now it is safer and safer to use a qualified electronic signature. An unqualified signature is now practically not used. The need for such a signature was at a transitional stage, when the law was in place, and there were no accredited certification centers yet.

A qualified electronic signature is intended to identify the person who signed an electronic document, and is an electronic analogue of a handwritten signature in cases provided for by law.

A qualified electronic signature is used when making civil transactions, providing state and municipal services, performing state and municipal functions, and performing other legally significant actions.

There are several schemes for building a digital signature:

- *based on symmetric encryption algorithms.* This scheme provides for the presence of a third party in the system – an arbitrator who is trusted by both parties. Document authorization is the very fact of encrypting it with a secret key and transferring it to the arbiter. However, such a scheme is very vulnerable to attacks on the arbiter. An arbitrator who has once been compromised is unlikely to be trusted;

– *based on asymmetric encryption algorithms*. At the moment, such electronic signature schemes are the most widespread and are widely used in both «citizen-government» and «business-to-business» relationships.

In addition, there are other types of digital signatures (group signature, indisputable signature, trusted signature), which are modifications of the schemes described above. Their appearance is due to the variety of tasks solved with the help of EDS.

Since the documents being signed are variable and often very large, in most cases the signature is placed not on the document itself, but on the hash function of this document.

Using hash functions has the following advantages:

– *the advantage of computational complexity*. Usually the hash of a digital document is made many times smaller than the size of the original document, and the algorithms for calculating the hash are faster than the EDS algorithms. Therefore, generating a hash of a document and signing it is much faster than signing the document itself;
– *compatibility advantage*. Most algorithms operate on strings of data bits, but some use different representations. The hash function can be used to convert arbitrary input text to a suitable format;
– *integrity benefits*. Without the use of a hash function, a large electronic document in some schemes must be divided into sufficiently small blocks to use the algorithm for setting an EDS. During verification, it is quite difficult to determine if all blocks were received and if they are in the correct order.

However, the use of a hash function is not necessary for an electronic signature, and the function itself is not part of the EDS algorithm, therefore, any hash function can be used or not used at all.

It should also be noted that the use of hashing introduces its own vulnerabilities into the EDS scheme.

Asymmetric EDS schemes refer to public key cryptosystems. Unlike asymmetric encryption algorithms, in which encryption is performed using a public key and decryption using a private key, in asymmetric digital signature schemes, signing is performed using a private key, and signature verification is performed using a public key.

For example, consider the algorithm for setting and verifying a digital signature. Instead of a document, we

use the text «1 2 3 4 5 6 7». For ease of understanding, let's take the algorithm for obtaining the hash function as the sum of the values of all the numbers in the message. When generating keys, we use the RSA system algorithm:

1) Choose p = 3 and q = 11.

2) Define n = 3 • 11 = 33.

3) Find (p-1) • (q-1) = 20. Therefore, d will be equal, for example, 3: (d = 3).

4) Choose the number e according to the following formula:

(e • 3) mod 20 = 1. So e will be equal, for example, 7: (e = 7).

5) We will accept the pair of values e and n as the public key {7,33} and publish it so that anyone can check the validity of the signature.

6) We accept a pair of values d and n as a secret signature key {3.33} and we will use it to issue an EDS

7) To set up an EDS for the text, we will receive a hash function, which will be equal to h = 1 + 2 + 3 + 4 + 5 + 6 + 7 = 28.

8) Let's encrypt h using the private signature key {3.33}

EDS1 = (23) mod 33 = 8 mod 33 = 8;
EDS2 = (83) mod 33 = 512 mod 33 = 17.

Thus, the message "1 2 3 4 5 6 7 «and his electronic digital signature» 8 17 «. The public key of the signature owner {7,33} will also be published.

Upon receipt of the message «1 2 3 4 5 6 7» and its EDS «8 17» perform the following sequence of actions:

1) Let's calculate the hash function of the message, which will be equal to h1 = 1 + 2 + 3 + 4 + 5 + 6 + 7 = 28.

2) Let's decrypt the EDS in order to obtain the hash function of the original message h using the public key

h(1) = (87) mod 33 = 2 097 152 mod 33 = 2;
h(2) = (177) mod 33 = 410 338 673 mod 33 = 8.

Since the calculated hash function h1 and the decrypted hash function coincide, we make a decision

that the message "1 2 3 4 5 6 7" has not been modified and signed by the owner of the key, which is true.

Consider an example when you receive another message that has the same signature and the same key. For example, the message «2 3 4 5 6 7 8» is received. Then the hash value calculated by the recipient will be 35, and the decrypted value will be equal, as we calculated above, 28. Accordingly, the values are not equal and the conclusion that this document (message) does not correspond to the submitted signature, which is also true.

Finally, consider an example that demonstrates the potential for collisions in a hash function and, therefore, the security implications. The hashing algorithm we use is not sensitive to the permutation of characters in the message, and also easily allows us to calculate twin messages that have the same hash value. For example, the recipient got the forged message «10 8 5 5», EDS and the key from the document of the first example. As a result of calculations, the recipient will receive h1 = 28, which will coincide with the decrypted value. Naturally, on the basis of such data, he will decide that the message «10 8 5 5» with the EDS «8 17» was sent by the owner of the key, although this is a fraud.

As can be seen from the examples, attacks of three types can be carried out on EDS: collisions of the first and second order and social attacks. Forgery of a document (collision of the first kind).

A first-order collision or document forgery means that an attacker can try to match a document to a given signature so that the signature matches it (as shown in the example). However, in most cases, there can be only one such document. The reason is this:

– the document is still a meaningful text and it is almost impossible to select changes that satisfy the attacker;

– the text of the document is drawn up in the prescribed form, which also significantly complicates the forgery of a document (of course, only in terms of an EDS);

– documents are rarely issued in the form of a text file, most often in DOC or PDF format.

If a fake set of bytes collides with the hash of the original document, then the fulfillment of these three conditions also makes the probability of this attack practically equal to zero.

Much more likely to be a Type II attack. In this case, the attacker fabricates two documents with the same signature, and at the right time substitutes one for the other. When using a reliable hash function, such an attack must also be computationally complex. However, these threats can be realized due to the weaknesses of specific hashing algorithms, signatures, or errors in their implementation. In particular, in this way it is possible to carry out an attack on SSL certificates and the MD5 hashing algorithm.

Social attacks are not aimed at cracking digital signature algorithms, but at manipulating users' public and private keys:

- an attacker who stole a private key can sign any document on behalf of the key owner;
- an attacker can trick the owner of the private key to transfer this key to him;
- an attacker can trick the owner into signing a document, for example, using a blind signature protocol;
- an attacker can replace the owner's public key with his own, impersonating him.

The use of key exchange protocols, establishing the lifetime of EDS keys and protecting the private key from unauthorized access reduces the risk of these attacks..

3.6 Prospects for the Development and Problems of Using Cryptosystems with Open (Public) Key

The development of cloud technologies, cryptocurrencies and just electronic money, blockchain technology, and even the «departure» to electronic business, of course, will increase the requirements for the applied technologies of cryptographic data protection, reliable user authentication and legal protection of transactions. The solution to these issues is impossible without the use of methods for distributing keys in an insecure digital environment – *methods of asymmetric cryptography.*

The expected appearance of quantum computers, parallelization of computations, requires researchers in the field of cryptographic protection to fundamentally strengthen cryptographic algorithms. And one of these steps in the field of asymmetric encryption was the emergence of the *SHA3 algorithm*.

Naturally, this will lead to an increase in the computational complexity of cryptoalgorithms. Therefore, one of the possible business approaches can be the active introduction of lightweight cryptoalgorithms that allow you to get practically the same cryptographic strength, with a much lower computational strength.

Also, one should not discard the fact that all modern asymmetric cryptography is based on the lack of results on the construction of mathematical or computational algorithms for solving inverse problems. And no one guarantees that such algorithms will not appear in the future. For example, in 1994, the first Russian standard in the field of electronic digital signatures was adopted – GOST R34.10-94 «Information technology. Cryptographic information protection. Procedures for generating and verifying an electronic digital signature based on an asymmetric cryptographic algorithm». He defined the procedures for working with EDS based on the El Gamal scheme. The impossibility of forging a signature is based on the complexity of solving the discrete logarithm problem in a field of p elements. However, mathematics does not stand still, and mathematicians have developed the so-called *number field sieve method*. With its help, you can «crack» the EDS generated according to the El Gamal scheme, at least in the case of a 512-bit module p.

Questions for Self-Control

1. What is an asymmetric cryptosystem?
2. Who uses the public (public) key in an asymmetric crypto system?
3. Who proposed the idea of asymmetric encryption?
4. What was the first asymmetric encryption system?
5. What encryption algorithm is the US based on post-quantum cryptography?
6. What is the recommended key length in asymmetric cryptosystems?
7. What is a hash function?
8. What hashing algorithms are currently recommended for use in the US?
9. What is a digital signature?
10. What algorithms for setting and verifying EDS are used in the Republic of Kazakhstan?

Recommended Readings

1. A.P. Alferov, A.Yu. Zubov, A.S. Kuzmin, A.V. Cheryomushkin. Fundamentals of Cryptography. – Helios ARV, 2002.
2. Kahn D. The Codebreakers: The Story of Secret Writing – Macmillan, 1967.
3. A.V. Babash, G.P.Shankin. Cryptography. – M. SOLON-PRESS, 2007.
4. Fomichev V.M. Discrete mathematics and cryptology: A course of lectures / ed. ND Podufalov. – M.: Dialog-MEPhI, 2013.
5. Gabidulin E.M., Kshevetskiy A.S., Kolybelnikov A.I. Information security: a tutorial. – M.: MFTI, 2011.
6. Mao V. Modern cryptography: Theory and practice. – M.: Williams, V. Miller, Use of elliptic curves in cryptography, Advances in cryptology-CRYPTO 85, Springer Lecture Notes in Computer Science vol 218, 1985.
7. Rosstandart order of August 7, 2012 No. 216-st. Retrieved May 31, 2013.
8. Diffie W., Hellman M. E. New Directions in Cryptography // IEEE Trans. Inf. Theory / F. Kschischang. – IEEE, 1976.
9. Rivest R., Shamir A., Adleman L. A method for obtaining digital signatures and public-key cryptosystems // Commun. ACM. – New York City: ACM, 1978.
10. Order of the Central Bank of the Russian Federation of 31.01.1995 N 02-13 «On the introduction of state standards of the Russian Federation in the system of the Central Bank of the Russian Federation» (Russian) ORDER of the Central Bank of the Russian Federation No. 02-13.
11. Rivest R. RFC 1321, The MD5 Message-Digest Algorithm: The MD5 Message-Digest Algorithm // Request for Comments. – Internet Engineering Task Force, 1992 /
12. Ah Kioon, Mary Cindy, Wang Z., Deb Das S. Security Analysis of MD5 Algorithm in Password Storage // Applied Mechanics and Materials. 2013.
13. Schneier B. Applied cryptography. Protocols, algorithms, source texts in C = Applied Cryptography. Protocols, Algorithms and Source Code in C. – M.: Triumph, 2002.
14. Ryabko B.Ya., Fionov A.N. Fundamentals of Modern Cryptography for Information Technology Professionals. – Scientific world, 2004.
15. Nils Ferguson, Bruce Schneier. Practical Cryptography: Designing and Implementing Secure Cryptographic Systems. – M.: Dialectics, 2004.
16. Law of the Republic of Kazakhstan dated January 7, 2003 «On an electronic document and electronic digital signature».

17. GOST R34.11-2012 «Information technology. Cryptographic information protection. Hash function».
18. B.A. Forousan. El Gamal digital signature scheme // Encryption Key Management and Network Security / Per. A.N.Berlin. – Lecture course.
19. Menezes A.J., Oorschot P. v., Vanstone S.A. Handbook of Applied Cryptography. – CRC Press, 1996.
20. Non-commercial project for testing the cryptographic strength of algorithms using distributed computing – www.distributed.net.

CHAPTER 4. CRYPTOGRAPHIC KEY MANAGEMENT

Keywords: key distribution center, service, session key,SSL, TSL

4.1. Key Management in Systems with Open (Public) Key

One of the main applications of the public key encryption scheme is to solve the key distribution problem. There are two very different uses for public key encryption in this area:

– distribution of public keys;
– using public key encryption to distribute secret keys.

Several methods have been proposed for distributing public keys. In fact, they can be grouped into the following general classes:

– public announcement;
– publicly accessible catalog;
– an authoritative source of public keys;
– public key certificates.

The simplest way to distribute is publicly disclosing public keys, or so-called *uncontrolled key distribution*. In this method, any party participating in the exchange of data can provide their public key to any other party or transfer the key by means of communication in general for everyone.

This approach is convenient, but it has one drawback: such a public announcement can be made by anyone, including an attacker. This means that someone posing as user A can send the public key to another network user or offer such a public key for public use. As long as user A opens the fraud and warns other users, the forger will be able to read all encrypted messages that have arrived for A during this time, and will be able to use the forged keys for authentication.

The second way is using a publicly accessible directory or a centralized scheme. In such a situation, some reliable center should be responsible for the maintenance and distribution of the public catalog. Such a scheme should include the following elements:

– an authorized entity that maintains a directory with entries of the form {name, public key} for each of the participants;
– each participant registers his public key. Such registration must take place either in person at the participant's presence or through secure communication channels;

– if the key is compromised, the participant can replace the existing key with a new one at any time using authentication means;
– the catalog is regularly updated.

This scheme is more secure than individual public announcements, but it is also vulnerable. If the adversary succeeds in obtaining the private key of the entity authorized to maintain the catalog, he can issue falsified public keys and, therefore, act on behalf of any of the participants in the exchange of data and read messages intended for any participant.

The third option is to use an authoritative source of public keys. This scenario assumes the existence of some kind of key distribution center authorized to maintain a dynamic catalog of public keys of all participants in the exchange of data. In addition, each of the participants reliably knows the center's public key, but only the center knows the corresponding private key. In this case, the following actions are performed:

– initiator A sends a message with a date / time stamp (rendering N_1) to an authoritative source of public keys with a request for the current public key of participant B;

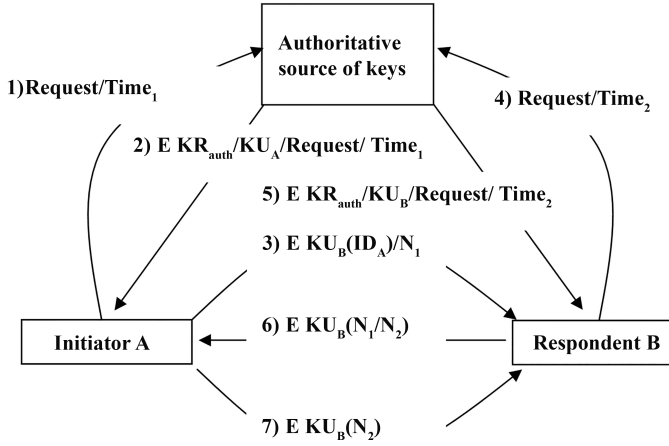


Figure 4.1 – Scenario for key distribution when using an authoritative key source

– the authoritative source replies with a message that is encrypted using the private key of the authoritative source KRauth. Initiator A can decrypt this message using the public key of the authoritative source. Therefore, sender A can be confident that the message comes from an authoritative source. This message should include: the public

key of participant B (KU_b), the original request (so that party A can make sure that the request has not been changed on the way to the authoritative source) and the original date / time stamp (opportunity N_1) so that the sender A can make sure that this is the answer to this particular request;
- the initiator A saves the public key of the participant B and uses it to encrypt the message sent to the recipient B and containing the identifier of the sender A (ID_A) and the opportunity N_1;
- respondent B receives the public key of participant A from an authoritative source in exactly the same way as sender A received the public key of receiver B.

At this point, the public keys have been delivered to participants A and B, so that now A and B can begin secure communication. But before that, it is advisable for them to perform the following two additional steps:

- respondent B sends a message to initiator A, encrypted with KU_A and containing the opportunity of the sender A (N_1), as well as a new opportunity generated by the participant B (N_2). The presence of N1 in this message convinces participant A that the sender of the received message was B;
- initiator A returns N_2 encrypted using the public key of participant B, so that he can verify that the sender of the response is A.

This key distribution option requires seven messages. However, sending the first four messages is infrequent, since both parties can save each other's public keys for later use, which is usually called caching.

The fourth option uses public key certificates. In the previous scenario, Authority Source is the bottleneck in the system. An alternative approach was proposed in 1978 by Loren Kohnfelder. In this method, each certificate contains a public key and other information, is generated by an authoritative certificate authority, and issued to the principal. The system has the following requirements:

- any participant must be able to read the certificate to determine the name and public key of the certificate owner;
- any participant should be able to verify that the certificate comes from an authoritative source of certificates and is not a fake;
- only an authoritative source of certificates should be able to create and modify certificates.

Denning added the following requirement to these rules – any participant must be able to check the validity of the certificate.

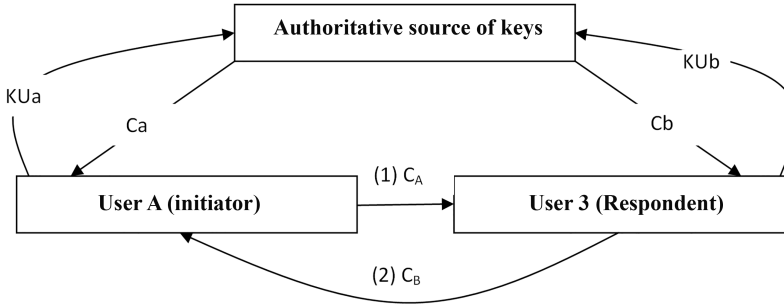


Figure 4.2 – Exchange of public key certificates

Each participant in such a system contacts the certificate authority, providing a public key and requesting a certificate for it through a secure form of communication.

The Center sends CA and CB certificates containing

– certificate validity period;
– owner ID;
– the public key of the owner of the certificate.

However, certificates are encrypted using the private key of an authoritative source.

In this case, user A can send the certificate to any participant. The recipient of the certificate uses the public key KUauth of the certificate authority to read the certificate. This gives a guarantee that the certificate came from him..

4.2 Secret Key Exchange Protocol

With symmetric encryption, both parties must have the same secret key. Therefore, the cryptographic strength of any symmetric cryptographic system is highly dependent on the key distribution system used.

For two sides A and B, the distribution of keys can be organized in four different ways:

– the key is selected by side A and physically delivered to B;
– the key is chosen by a third party and physically delivered to A and B;
– one of the parties transmits the new key in encrypted form using the old key;
– third party C delivers keys A and B via secure communication channels, i.e. a certain Key Distribution Center (KDC) is used.

In this case, for symmetric cryptosystems, the key distribution scheme (protocol) can be centralized and distributed (with an intermediary and self-sufficient).

The use of KRC assumes the organization of a hierarchy of keys (at least two levels). Communication between end users is encrypted using a temporary key called a session key. The session key is received from the KRC through the same communication channels that are used for data delivery. Session keys are transmitted in encrypted form, and for their encryption, a master or master key is used, which is common for the KRC and the given user.

This master key scheme requires N (per user). They are distributed in a non-cryptographic manner (physical delivery to the addressee).

Suppose that when using a centralized key distribution scheme, user A wants to transfer information to user B and a one-time session key is required to protect the data.

In this case, user A has a secret key K_a, known only to him and the KRC, and user B has K_b (K_a and K_b are the master keys of users A and B, respectively, Ks is a one-time session key).

The exchange of information is as follows:

1. User A sends a request to the KRC to obtain a session key to protect communication with B. In this case, the sent request must include:
   - information allowing to unambiguously determine A and B (ID_A, ID_B);
   - some identifier N1, unique for each request and called a case. The *occasion* can be a time, a counter, a random number.
2. The KRC responds to the request of user A by encrypting the answer with the key Ka (the master key of user A). The only user who can read the answer is A (therefore, A is sure that the message came from the KRC).

The KRC response message includes the following elements intended for A:

- a one-time session key K_s (for communication between user A and user B);
- request with opportunity N_1 so that user A can match the response with the request.

Thus, user A can make sure that his request has not been changed on the way to the KRC, and the opportunity does not allow confusing the answer to this request with the answer to previous requests.

The CRC response message includes the following elements intended for B.

– one-time session key Ks;
– User ID_A – ID_A.

Both elements are encrypted using the K_B key (master key of the KRC and user B).

1. User A saves his session key and sends to user B information from the KRC intended for B.

User B receives Ks and knows that the information received has come from the CRC (since it is encrypted by the CW, which only B and the KRC know).

Thus, both user A and B have a session key.But before exchanging data, it is advisable to do the following:

2. Using the received session key Ks user B sends user A a new opportunity N_2.

3. User A uses Ks to return f (N2). This is to convince user B that the message he originally received was not reproduced by the attacker.

Thus, not only key transfer is provided, but also authentication (steps 4 and 5).

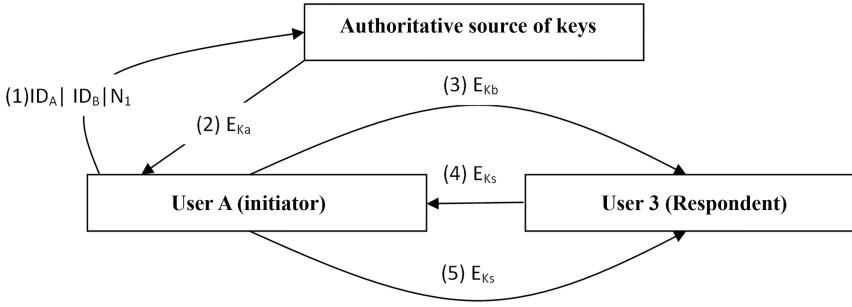


Figure 4.3 – Centralized scheme for the distribution of secret keys

It should be noted that it is not necessary to assign the key distribution function to one KRC. It is more profitable to use some hierarchy of the KRC. The more often the session keys change, the more reliable they are, but the distribution of the session keys delays the start of the communication session and increases the network load.

The use of the KRC assumes that the KRC must inspire confidence and be reliably protected from encroachment. These requirements can be waived by using a decentralized (self-sufficient) key distribution scheme.

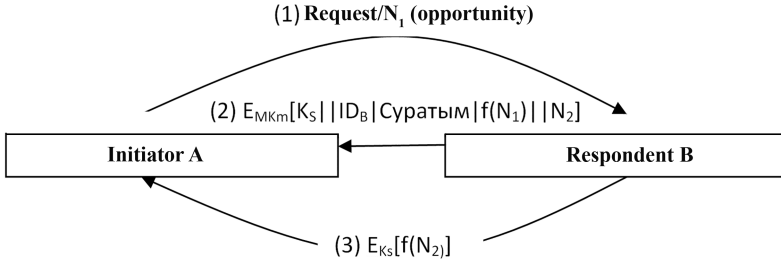


Figure 4.4 – Decentralized secret key distribution scheme

In a decentralized key distribution scheme, the session key can be determined by the following sequence of actions:

- user A sends a request to receive Ks + opportunity N_1;
- user B answers by encrypting the answer using a common master key EMKm between A and B;
- user A returns f (N_2), encrypting with Ks.

4.3 Security Certificates

Security certificates or public keys are data structures designed to store, distribute, or send public keys over insecure channels with a guarantee of their integrity and authenticity (belonging to specific subjects).

The purpose of using certificates is to make the public keys of some subjects (interacting parties) available to others so that their authenticity (ie belonging to specific subjects) and validity can be reliably verified.

In the certificate, the identifying information about the object (party, key owner) is reliably associated with its public key. Authenticity is ensured by verifying the signature of the publisher – a certification authority or certification authority that you trust.

A public key certificate is a digital document (data structure) consisting of a data section and a signature section.

The data section contains public data including, at a minimum, the public key and identifying information (object name and additional information). Custom name in distinguished name (DN) format. The DN defines the name

of the user, and any additional attributes required for a unique identifier for the user (for example, the DN might contain the user's employee number). A public key is required so that users can implement information protection services through the use of public keys in the certificate.

Additional information may include:

- the validity period of the public key;
- serial number or name identifying the certificate or key;
- additional information about the owner of the certificate (for example, a regular or network address);
- additional information about the key (eg algorithm and intended use);
- special characteristics related to the identification of the represented object, the generation of a key pair or other policy issues;
- information to facilitate signature verification;
- specific operations for which the public key should be used (data encryption or digital signature).

When they talk about issuing a certificate, it means signing the data section by a certification authority. By issuing a certificate, the publisher authenticates (gives its guarantees of authenticity) the relationship between the public key of the subject and the information identifying it.

One of the first protocols to use certificates was SSL. SSL (Secure Sockets Layer) is a cryptographic protocol that implies more secure communication. It uses asymmetric cryptography to authenticate exchange keys, symmetric encryption to preserve confidentiality, message authentication codes for message integrity.

SSL was originally developed by Netscape Communications to add HTTPS to its Netscape Navigator web browser. Subsequently, on the basis of the SSL 3.0 protocol, an RFC standard was developed and adopted, which was named TLS. In 2014, the US government reported a vulnerability in the current version of the SSL protocol. SSL should be dropped in favor of TLS.

Essentially, an SSL certificate is a digital signature of a site that confirms its authenticity and security to the client. Using a certificate helps protect both the site owner and his clients. An SSL certificate enables the owner to apply SSL encryption technology to his site. *Thus, the purpose of an SSL certificate is to ensure a secure connection between the server and the user's browser, reliably protect data from interception and spoofing.*

Certificates are usually purchased not directly from a certification authority, but through partners. In Kazakhstan and Russia, many companies sell certificates of well-known certification authorities, such as Comodo, Geotrust, GoDaddy, GlobalSign, Symantec and others. The SSL root

certificates of these authorities are preinstalled as trusted in all popular browsers.

There are certificates of different levels of verification. To protect personal data of users, they usually use a certificate with simplified verification – DV (Domain validation).

The next level is the *Organization validation (OV) certificate, which is used to validate the relationship between the domain name, the domain owner, and the company using the certificate. That is, such a certificate certifies not only the domain name, but also that the site belongs to a really existing organization.*

For a better check of the company and its authority to purchase certificates, so-called certificates with extended validation – EV (Extended validation) are used.

There are also national safety certificates.

4.4 Anonymous Key Distribution

If users themselves cannot choose their own keys, then they must use the services of a key distribution center. The problem is that keys have to be distributed in such a way that no one can determine who got which key. The key distribution procedure in this case is called «anonymous key distribution» and may look like this:

- user A selects a pair «public key, secret key»;
- KRC generates a continuous stream of keys;
- KRC encrypts the keys, one by one, with its public key;
- KRC transmits encrypted keys, one by one, to the network;
- user A chooses a key at random;
- user A encrypts the selected key with his public key;
- user A waits for some time and sends the double-encrypted key back to the KRC;
- KRC decrypts the twice encrypted key with its own private key, leaving the key encrypted once with the public key of user A;
- KRC sends the encrypted key back to user A;
- user A decrypts the key with his private key. A completely anonymous session can be established using, for example, SSL, the RSA or Diffie-Hellman algorithm to generate exchange keys. In the case of RSA, the client encrypts its private key (pre_master_secret) with the public key of an uncertified server. The client learns the public key from the key exchange message from the server. The result is sent in a key

exchange message from the client to the server. Since the interceptor does not know the server's private key, it will be impossible for him to decrypt the secret (pre_master_secret).

With the Diffie-Hellman algorithm, the server's public parameters are contained in the key exchange message from the server, and are sent to the client in the key exchange message. An interceptor that does not know the private values cannot find the secret (pre_master_secret).

Questions for Self-Control

1. What is the key exchange problem?
2. How is the key exchange for symmetric encryption implemented?
3. Who came up with the idea of key exchange?
4. Which key exchange system is currently the most common?
5. What key distribution centers are there?
6. What are the problems of using key distribution centers?
7. What protocols use cryptographic key management mechanisms?
8. Why did the US recommend that you stop using SSL certificates?
9. What is a TSL certificate?
10. What safety certificates are used in the Republic of Kazakhstan?

Recommended Readings

1. Babash A.B. Cryptographic methods of information protection. – M.: KNO-RUS, 2018
2. Sepehrdad P. Discovery and Exploitation of New Biases in RC4. – Springer Berlin Heidelberg, 2011.
3. Menezes A.J., Paul C. van Oorschot, Scott A. Vanstone. Handbook of Applied Cryptography. 1996.
4. Claessens J. Computer Security and Industrial Cryptography. – 3-t. – Leuven- Heverlee, Belgium, 2002.
5. Viega J. Network Security with OpenSSL. – 1-st. – O'Reilly Media, USA, 2002.
6. Rescorla E.. SSL and TLS: Designing and Building Secure Systems. – 1-st. – Addison-Wesley Professional, 2000.
7. Thomas S. SSL & TLS Essentials: Securing the Web. – 1-st. – Wiley, February 11, 2000.
8. Schneier B. Applied cryptography. – M.: Science, 2006

CHAPTER 5. BRIEF INFORMATION ABOUT CRYPTO ANALYSIS

Keywords: cryptanalysis, cryptographic attack, brute force, dictionary brute force, frequency analysis, differential analysis, linear analysis, side channel attacks.

5.1 Brief History of Cryptanalysis

The term «cryptanalysis» was introduced into scientific terminology in 1920 by the American cryptographer William F. Friedman. And although this term is less than a hundred years old, the very actions to decrypt secret messages began to be taken from the moment the first ciphers appeared[1].

In the VIII-XV centuries, Arab scientists made a significant contribution to the development of cryptanalysis. At that time, the works of Khalil al-Farahidi («The Book of the Secret Language»), Al-Kindi («Manuscript on Deciphering Cryptographic Messages»[2]), Shihab al-Kalkasandi («Subh al-Aasha»[3]) were written.

In the 15th-19th centuries, the development of mechanisms for decrypting intercepted messages began to actively develop in Europe. The works of such well-known European scientists of the modern time as Leon Battista Alberti, Johann Triemus, Friedrich Kasiski, Auguste Kerkhoffs are devoted to both the description of ciphers and issues of cryptanalysis.

Also during this period, the first state services began to stand out in Europe, in which they dealt with the decryption of intercepted secret messages. One of the first such services was the French «black cabinets» (French Cabinet Noir) organized by the Minister of War Francois Lavoie. The English Black Office appeared in 1655 under the Secret Service of the Parliament (English Secret Office) to transcribe letters. During the reign of Empress Maria Theresa, a decryption service (Geheime Kabinets-Kanzlei) was established in Austria. At its peak in the years 1730-1760, Kabinets-Kanzlei was considered the best organization of its kind in Europe. In Russia, the date of the establishment of the first state encryption service can be considered 1549 – the formation of an «ambassadorial order» with a «digital department».

---

[1] An example is one of the first known decryption devices - «antiscitalu».

[2] Al-Kindi's book contains the first known mention of frequency-based crypto-analysis.

[3] In the book of Shihab al-Kalkashandi, the first currently known tables of the frequency of occurrence of letters in the Arabic language based on the text of the Koran are presented.

After the First World War, formally closed in 1911, the «black offices» came under the jurisdiction of military intelligence and counterintelligence.

During this period, electromechanical decryption devices were actively introduced into cryptanalysis, the most famous of which was the «Bomb machine» by Alan Turing.

The advent of electronic computers gave rise to new directions of cryptanalysis.

Initially, cryptanalysis methods were based on the linguistic laws of natural text and were implemented using manual calculations. Over time, the role of purely mathematical methods of analysis in cryptanalysis increased, the implementation of which is impossible without the use of computer technology. At present, for cryptanalysis, as a rule, specialized cryptanalytic computers are used.

5.2 Methods of Cryptanalysis

Currently, the term «cryptanalysis» is understood as a science that deals with the assessment of the strengths and weaknesses of encryption methods, as well as the development of methods to break cryptosystems.

An attempt to break a specific cipher using cryptanalysis methods is now called a cryptographic attack on that cipher. A cryptographic attack in the course of which it was possible to uncover the cipher is called «breaking» or «breaking» the cipher.

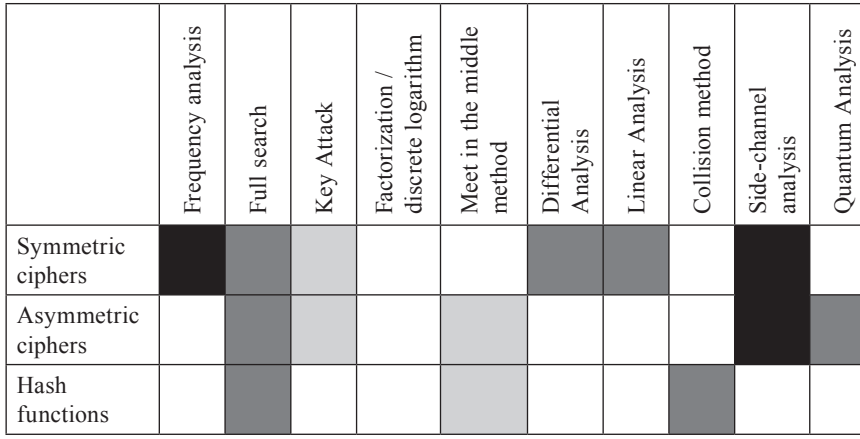


Figure 5.1 – «Objectives» of cryptanalysis methods

One of the first types of cryptographic attacks was frequency analysis. The method is based on the fact that the frequency of occurrence of a given letter of the alphabet in sufficiently long texts is the same for different texts of the same language. Moreover, in the case of mono-alphabetic encryption, if the ciphertext contains a character with a similar probability of occurrence, then we can assume that it is the specified encrypted letter. Similar reasoning applies to N-grams in the case of polyalphabetic ciphers. The frequency method gave rise to the requirement for a uniform distribution of characters in the ciphertext. Today, the principles of frequency analysis are widely used in password guessing programs and can reduce the search time by several orders of magnitude.

The brute-force method is also sometimes called the *«brute force method».* The ability to break ciphers by brute-force attacks appeared with the spread of high-performance computing technology for cryptanalysts.

When attempting a ciphertext-only attack, it is required to analyze the output of the algorithm and check its «meaningfulness». The task of highlighting a meaningful text, that is, determining the fact of correct decryption, is solved with the help of a computer using the so-called «Markov chains» or finite automata.

Key Attack. Most of the keys used by people have a phonetic similarity to words in natural language, caused by the ease of remembering this kind of information, as opposed to randomly generated keys. Therefore, the use of specially generated dictionaries can significantly reduce the key selection time.

Pseudo-random number generators are another source of threat to the strength of the cryptosystem. If a weak cryptographic algorithm is used to generate keys, then regardless of the cipher used, the entire system will be unstable.

The greatest progress in the development of methods for uncovering block ciphers was achieved at the very end of the twentieth century, and is associated with the emergence of two methods – differential, or differential, cryptanalysis and linear cryptanalysis.

Differential cryptanalysis method combines the idea of a general linear structure with the use of probabilistic-statistical research methods. It is based on the study of the differences between the encrypted values at different rounds for a pair of matched open messages when they are encrypted with the same key. The method was proposed in 1990 by Israeli specialists Eli Biham and Adi Shamir. This is a statistical attack that offers a list of the most likely encryption keys.

The emergence of this method of cryptanalysis led to the emergence of a requirement for the uniformity of the distribution of the difference of ciphertexts, for compliance with which the ciphers were checked at well-known competitions, such as AES and NESSIE. It should also be noted that differential analysis is applicable to hacking hash functions.

Like differential analysis, linear analysis is a combined method that combines the search for linear statistical analogs for encryption equations and statistical analysis of available plain and cipher texts, also using matching and brute force methods. This method examines the statistical linear relationships between the individual coordinates of the plaintext vectors, the corresponding ciphertext and the key, and uses these relationships to statistically determine the individual coordinates of the key vector.

The method of linear cryptanalysis made it possible to obtain the strongest results in the disclosure of a number of iterative block encryption systems, including the DES system. The method of linear cryptanalysis in an implicit form was proposed in the work of Sean Murphy also in 1990, where it was successfully used in the analysis of the FEAL block cipher system. In 1992, Mitsubishi Electric Company senior researcher Mitsuru Matsui formalized this approach, and later successfully applied it to the analysis of the DES cryptoalgorithm.

Almost all asymmetric cryptography algorithms used are based on factorization problems (for example, the RSA cryptosystem) and discrete logarithm in various algebraic structures (El-Gamal digital signature scheme). Therefore, for cryptanalysis of asymmetric cryptosystems, universal methods can be used – for example, the «meeting in the middle» method. Another approach is to solve the inverse mathematical problem underlying the asymmetric cipher. In recent years, significant progress has been observed in the study of the problem of factorization of integers and discrete logarithms. This can be confirmed by the following fact: in 1977 it was believed that the factorization of a 125-bit number would take 40 quadrillion years, but already in 1994 a number consisting of 129 binary digits was factorized.

The most efficient algorithms for factorization and discrete logarithms today have not exponential, but subexponential time complexity. These are algorithms that use a factor base. The first subexponential algorithm to compute the discrete logarithm in a prime field Zp was proposed by Leonard Adleman. In practice, Adleman's algorithm was not efficient enough; Don Coppersmith, Andrew Odlyzko and Richard C. Schroeppel proposed their

version of the subexponential discrete logarithm algorithm – «COS», and the number field sieve algorithm proposed by Oliver Shirokauer works more efficiently for p> 10100 and various modifications of the COS method.

A number of successful attacks on systems based on the complexity of discrete logarithms in finite fields led to the fact that the American and Russian standards for electronic digital signature, which were adopted in the nineties and based on the El Gamal scheme, were updated in the 2000s and translated into elliptic curves.

Side channel attacks use information that can be obtained from the encryption device and is neither clear text nor ciphertext. Such attacks are based on the correlation between the values of physical parameters measured at different moments during computations and the internal state of the computing device related to the secret key. This approach is less generalized, but often more powerful than classical cryptanalysis.

Figure 5.2 – Mitsuru Matsui (1961), Andrew Michael Odlizko (1949), Richard Schreppel (1948)

In recent years, the number of cryptographic attacks exploiting weaknesses in the implementation and placement of cryptoalgorithm mechanisms has increased dramatically. The adversary can measure the time it takes to complete a cryptographic operation, or analyze the behavior of the cryptographic device when certain computation errors occur. Another approach involves tracking the energy consumed by the system during operations with the secret key (for example, decrypting or generating a signature). It is sometimes not difficult to collect side information - today more than ten side channels have been identified, including electromagnetic radiation, errors in the communication channel, cache memory and light radiation.

In 1994, Peter Shor discovered the so-called «bounded probabilistic» factorization algorithm, which allows using a quantum computer to factor a number in polynomial time in the dimension of the problem. Shor's algorithm for factoring numbers was a major advance in the field of quantum computing algorithms. It was from this moment that increased funding for work on the creation of quantum computers began.

It is important to note that Shor's algorithm is extremely simple and is content with much more modest hardware than that of a general-purpose quantum computer. There are already concrete results in the field of quantum cryptography. So, the IBM corporation in 2017 at the IEEE Industry Summit on the Future of Computing announced the creation of a prototype of a 50-qubit quantum computer, and from 2016 everyone can work on a cloud 20-qubit quantum computer.

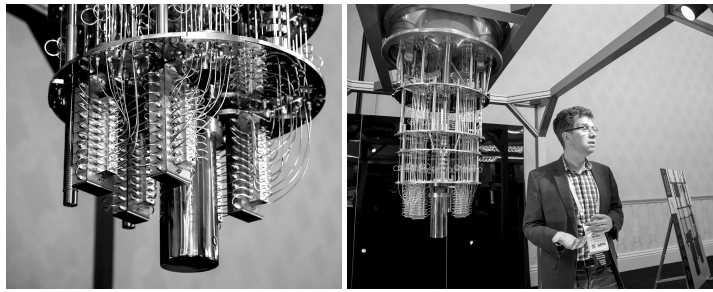


Figure 5.3 – IBM 50-qubit quantum computer (2017)

Questions for Self-Control

1. What is cryptanalysis?
2. What attacks are used against mono-alphabetic and polyalphabetic ciphers?
3. What is the main vulnerability of a mono-alphabetic cipher?
4. Why is the brute force method not applied to semi-structured and logically unrelated messages?
5. What attacks are used to break symmetric ciphers?
6. What is the meaning of differential cryptanalysis?
7. Why shouldn't encryption keys contain logically interconnected character strings?
8. What is linear cryptanalysis?

9. Which of the cryptosystems has the highest cryptographic strength at the present time: the El-Gamal system or the system on elliptic curves?
10. Why are quantum computers dangerous for cryptography?

Recommended Readings

1. Rejewski, Marian. Summary of Our Methods for Reconstructing ENIGMA and Reconstructing Daily Keys, and of German Efforts to Frustrate Those Methods. Appendix C to Kozaczuk, 1984
2. Panasenko, S. Modern methods of opening encryption algorithms. – Chief Information Officer, 2006.
3. Avdoshin S., Savelyeva A. Cryptanalysis: yesterday, today, tomorrow. – Open systems, 2009
4. Schneier B. Cryptanalysis // Applied Cryptography. Protocols, algorithms, source texts in C language = Applied Cryptography. Protocols, Algorithms and Source Code in C. – M.: Triumph, 2002.
5. Pilidi V.S. Cryptography. Introductory chapters. – Rostov-on-Don: SFedU, 2009.

CONCLUSION

The history of cryptography goes back several millennia – from primitive cryptography based on the simplest mono-alphabetic substitution to post-quantum cryptography. The technical devices of cryptography also changed, from papyrus and scytals, Jefferson cylinders, rotary encryption machines, to modern quantum cryptographic computers and electronic data storage and transmission systems.

The history of cryptography has seen many ups and downs. At some points in science, methods of protection prevailed, at others – methods of cryptanalysis. There were times when cryptography was equated with forbidden knowledge. However, science is successfully developing and serves as a tool for the development of human communications.

In many ways, cryptography is now central to software and hardware security regulators. For example, for laptop computers, tablets or smartphones, which are extremely difficult to physically protect, only cryptography can guarantee the confidentiality of information even in the event of theft.

The challenges of the XX century – *the emergence of computing technology – gave rise to the emergence of a new direction of symmetric encryption – block encryption, Feistel networks, and permutation-permutation networks.*

Nowadays, knowledge of cryptographic methods of data protection, identification and authentication of users in connection with the development of cloud technologies and electronic document management is acquiring a special role. The emergence of these technologies on a special role has put forward such problems as the secure distribution of encryption keys in a hostile environment, the possibility of parallelizing computations during brute force attacks. These challenges of the time have given rise to the emergence of solutions such as encryption with a public (public) key, hashing, digital signature.

The development of information resources and global networks gave rise to the emergence of security certificates and, in particular, made popular SSL certificates /

The emergence of cryptocurrencies at the end of the XX century – bitcoin, darkcoin, litecoin – gave rise to the emergence of blockchain technologies.

The latest challenges – the emergence of quantum computers – required cryptography to develop new encryption algorithms and standards built on

top of them that are protected from hacking on quantum computers. These challenges spawned the emergence of elliptic curve encryption algorithms and the emergence of the SHA3 hashing algorithm.

New challenges of the new time dictate the requirement for the development of new directions of cryptography – quantum cryptography, probabilistic, functional and honey encryption, DNA encryption and much more.

LIST OF RECOMMENDED SOURCES

1. Ah Kioon, Mary Cindy, Wang Z., Deb Das S. Security Analysis of MD5 Algorithm in Password Storage // Applied Mechanics and Materials. 2013.
2. Chris Christensen. Lester Hill Revisited // Taylor & Francis Group, LLC: Article. 2014.
3. Claessens J. Computer Security and Industrial Cryptography.- 3-t.- Leuven- Heverlee, Belgium, 2002.
4. Diffie W., Hellman M. E. New Directions in Cryptography // IEEE Trans. Inf. Theory / F. Kschischang. – IEEE, 1976.
5. Federal Information Processing Standards Publication 197 November 26, 2001 Specification for the ADVANCED ENCRYPTION STANDARD (AES)
6. Gardner M. A new kind of cipher that would take millions of years to break.- Mathematical Games, Scientific American, 1978. – 237(2).
7. ISO/IEC 10116 1997. Information technology – Security techniques – Modes of operation for an n-bit block cipher
8. ISO/IEC 10118-1 2000. Information technology – Security techniques – Hash- functions – Part 1: General
9. ISO/IEC 10118-2 2000. Information technology – Security techniques – Hash- functions – Part 2: Hash-functions using an n-bit block cipher
10. ISO/IEC 11770-1 1996. Information technology – Security techniques – Key management – Part 1: Framework
11. ISO/IEC 11770-2 1996. Information technology – Security techniques – Key management – Part 2: Mechanisms using symmetric techniques
12. ISO/IEC 13335-1 2004. Information technology – Security techniques – Management of information and communications technology security – Part 1: Concepts and models for information and communications technology security management
13. ISO/IEC 13888-1 2004. IT security techniques – Non-repudiation – Part 1: General
14. ISO/IEC 13888-3 1997. Information technology – Security techniques – Non- repudiation – Part 3: Mechanisms using asymmetric techniques
15. ISO / IEC 17799 2000. Information technology - Practical rules for information security management
16. ISO/IEC 9796-2 2002. Information technology – Security techniques – Digital signature schemes giving message recovery – Part 2: Integer factorization based mechanisms.

17. ISO/IEC 9796-3 2000. Information technology – Security techniques – Digital signature schemes giving message recovery – Part 3: Discrete logarithm based mechanisms.
18. ISO/IEC 9797-2 2002. Information technology – Security techniques – Message Authentication Codes (MACs) – Part 2: Mechanisms using a dedicated hash- function.
19. ISO/IEC 9798-1 1997. Information technology – Security techniques – Entity authentication – Part 1: General
20. ISO/IEC 9798-2 1999/ Information technology – Security techniques – Entity authentication – Part 2: Mechanisms using symmetric encipherment algorithms
21. ISO/IEC 9798-3 1998. Information technology – Security techniques – Entity authentication – Part 3: Mechanisms using digital signature techniques
22. ISO/IEC 9798-4 1999. Information technology – Security techniques – Entity authentication – Part 4: Mechanisms using a cryptographic check function
23. ISO/IEC 9979 1999. Information technology – Security registration of cryptographic algorithms
24. ISO/IEC TR 13335-3 1998. Information technology – Guidelines for the management of IT Security – Part 3: Techniques for the management of IT Security
25. Kahn D. The Codebreakers: The Story of Secret Writing. – Macmillan, 1967.
26. Luciano D., Prichett G. Cryptology: From Caesar Ciphers to Public-Key Cryptosystems. – The College Mathematics Journal. – Mathematical Association of America, 1987. – Vol. 18, Iss. 1.
27. Menezes A.J., Paul C. van Oorschot, Scott A. Vanstone. Handbook of Applied Cryptography.- 1996.
28. Miller, Use of elliptic curves in cryptography, Advances in cryptology – CRYPTO 85, Springer Lecture Notes in Computer Science vol 218, 1985.
29. Rejewski, Marian. Summary of Our Methods for Reconstructing ENIGMA and Reconstructing Daily Keys, and of German Efforts to Frustrate Those Methods. – Appendix C to Kozaczuk, 1984
30. Rescorla E.. SSL and TLS: Designing and Building Secure Systems. – 1-st. – Addison-Wesley Professional, 2000.
31. Rivest R. RFC 1321, The MD5 Message-Digest Algorithm: The MD5 Message-Digest Algorithm // Request for Comments. – Internet Engineering Task Force, 1992/

32. Rivest R., Shamir A., Adleman L. A method for obtaining digital signatures and public-key cryptosystems // Commun. ACM. – New York City: ACM, 1978.
33. Sepehrdad P. Discovery and Exploitation of New Biases in RC4. – Springer Berlin Heidelberg, 2011.
34. Thomas S. SSL & TLS Essentials: Securing the Web. – 1-st. – Wiley, February 11, 2000.
35. V.N.Krishna, Dr. A.Vinaya Babu. A Modified Hill Cipher Algorithm for Encryption of Data In Data Transmission (англ.) // Computer Science and Telecommunications : Georgian Electronic Scientific Journal. 2007.
36. Viega J. Network Security with OpenSSL. – 1-st. – O'Reilly Media, USA, 2002.
37. Abdikalikov K.A., Zadiraka V.K. Elements of modern cryptology and methods of protecting banking information. - Almaty .: Republican Publishing Office of the Kazakh Academy of Education named after I. Altynsarin, 1999
38. Avdoshin S., Savelyeva A. Cryptanalysis: yesterday, today, tomorrow. – Open Systems, 2009
39. Amirov A. Zh., Sultanova BK, Shakhanov D. Zh. History of the development of cryptology. Stages. - Young scientist. 2016. - No. 1.
40. Babash A.B. Cryptographic methods of information protection. – M.: KNO-RUS, 2018
41. Babash A.V., Shankin G.P. Cryptography (security aspects). – M.: SOLON-PRESS, 2007. – 512 p.
42. Barichev S.G., Goncharov V.V., Serov R.E. 2.4.2. AES standard. Algorithm Rijdael // Fundamentals of modern cryptography. – 3rd ed. – M.: Dialog-MI-FI, 2011.
43. Bolotov A.A., Gashkov S.B., Frolov A.B., Chasovskikh A.A. Algorithmic foundations of elliptic cryptography. – M.: MAI, 2000
44. Gabidulin E.M., Kshevetskiy A.S., Kolybelnikov A.I. Information security: a tutorial. – M.: MFTI, 2011.
45. Gabidulin E.M., Kshevetskiy A.S., Kolybelnikov A.I. Information security: a tutorial. – M.: MFTI, 2011.
46. GOST 28147-89 «Information processing systems. Cryptographic protection. Cryptographic transformation algorithm».
47. GOST 28147-89 Information processing systems. Cryptographic protection. Cryptographic pre-formation algorithms.
48. GOST R34.10-94 Information technology. Cryptographic information protection. Procedures for generating and verifying an electronic digital signature based on an asymmetric cryptographic algorithm.

49. GOST R34.11-2012 «Information technology. Cryptographic information protection. Hash function».
50. Zhelnikov V. Cryptography from papyrus to computer. - M .: ABF, 1996.
51. Law of the Republic of Kazakhstan dated January 7, 2003 «On electronic documents and electronic digital signatures».
52. Information portal «Law KZ». www.zakon.kz
53. Information portal «Information Security». www.sec.ru
54. Kotukhov M.M., Markov A.S. Legislative and legal and organizational and technical support of information security of automated systems. – SPb.: BHV-Petersburg, 1998
55. Kulyabov D.S. Protection of information in networks. – SPb.: BHV-Petersburg, 2004. – 130 p.
56. Maksimov Yu.N. Technical methods and means of protecting information. – SPb.: Polygon, 2000
57. Mao V. Modern cryptography: Theory and practice. – M.: Williams, 2005.
58. Non-commercial project for testing the cryptographic strength of algorithms using distributed computing - www.distributed.net.
59. Panasenko S. Modern methods of breaking encryption algorithms. – Chief Information Officer, 2006.
60. Pilidi V.S. Cryptography. Introductory chapters. – Rostov-on-Don: SFedU, 2009.
61. Order of Rosstandart dated August 7, 2012 No. 216-st. Retrieved May 31, 2013.
62. Order of the Central Bank of the Russian Federation of 31.01.1995 N 02-13 «On the introduction of state standards of the Russian Federation in the system of the Central Bank of the Russian Federation» (Russian) ORDER of the Central Bank of the Russian Federation No. 02-13.
63. Rickson F.B. Codes, ciphers, signals and secret transmission of information. – Astrel, 2011.
64. Ryabko B.Ya., Fionov A.N. Fundamentals of Modern Cryptography for Information Technology Professionals. – Scientific world, 2004.
65. Singh S., Book of ciphers. The secret history of ciphers and their decryption. – M.: Astrel, 2007. – 448 p.
66. Soboleva T.A. The history of encryption in Russia. – M.: OLMA-PRESS Education, 2002.
67. Spivak S.I., Vildanov A.N., Zaripova L.I. Achievements and applications of modern informatics, mathematics and physics: materials of the III All-

Russian scientific and practical correspondence conference (Neftekamsk, October 20-22, 2014). – Ufa: RITs BashGU, 2014.
68. Criminal Code of the Republic of Kazakhstan
69. Ferguson N., Schneier B. Practical Cryptography: Designing and Implementing Secure Cryptographic Systems. – M.: Dialectics, 2004.
70. Fomichev V.M. Discrete mathematics and cryptology: A course of lectures / ed. ND Podufalov. – M.: Dialog-MEPhI, 2013.
71. Forousan B.A. El Gamal digital signature scheme // Encryption Key Management and Network Security / Per. A.N.Berlin. – Lecture course.
72. Khorev A.A. Protection of information from leakage through technical channels. Part 1. Technical channels of information leakage. Tutorial. – M.: State Technical Commission of Russia, 1998
73. Cheryomushkin A.V. Lectures on the arithmetic foundations of cryptography. – M.: MTsNMO, 2002
74. Chmora A.L. Modern applied cryptography. – M.: «Helios ARV», 2001.
75. Schneier B. Cryptanalysis // Applied Cryptography. Protocols, algorithms, source texts in C language = Applied Cryptography. Protocols, Algorithms and Source Code in C. – M.: Triumph, 2002.
76. Schneier B. Applied Cryptography. Protocols, algorithms, source texts in C = Applied Cryptography. Protocols, Algorithms and Source Code in C. – M.: Triumph, 2002.

КРИПТОГРАФИЧЕСКИЕ СИСТЕМЫ

ВВЕДЕНИЕ

Вопросы безопасной передачи секретной информации стали актуальными для человечества, наверное, со времен появления первых племенных объединений и государств. Развитие и массовое внедрение средств передачи информации в XVIII-XIX веках применительно к государству и бизнесу только обострили данную проблему обеспечения тайны передачи важной информации. Появление же в XX веке социальных сетей, механизмов электронного бизнеса и территориального распределения хранилищ и потребителей информации сделало эту проблему сверхактуальной для современного цифрового общества – сейчас не только государственные структуры и бизнес, но и частные лица вынуждены решать проблемы обеспечения собственной информационной безопасности и защиты своих данных от несанкционированного использования.

Учебное пособие «Криптографические системы» способно представить читателю, занимающемуся решением проблем обеспечения безопасности передачи и хранения информации как основ построения современных криптографических систем, так и помочь в использовании криптографических протоколов при построении собственных систем защиты информации. В пособии предложены к рассмотрению два основных подхода к построению криптозащищенных систем: симметричное шифрование и шифрование с открытым (публичным) ключом. Для данных подходов рассмотрены основные механизмы и алгоритмы реализации, приведен анализ сильных и слабых сторон. Для обеспечения системного подхода в обеспечении безопасности рассмотрены вопросы управления ключами. Также рассмотрены криптографические протоколы, которые открыли новую веху в использовании методов защиты информации в различных компьютерных сетях.

Авторы выражают глубокую благодарность рецензентам: Генеральному директору Международной Академии Информатизации ***профессору Цеховому Алексею Филипповичу***, доктору технических наук, профессору Костанайского социально-технического университета имени академика З. Алдамжар ***Баймухамедову Малику Файзулловичу***, кандидату технических наук, и.о. профессора Костанайского инженерно-экономического университета имени М. Дулатова ***Баганову Николаю Анатольевичу***, заведующему кафедрой вычислительной техники и программного обеспечения Костанайского Государственного

Университета имени Ахмета Байтурсынова доценту ***Салыковой Ольге Сергеевне*** и доктору технических наук, профессору Рудненского индустриального института ***Олейнику Александру Ивановичу***.

Авторы внимательно и с благодарностью рассмотрят все критические замечания и предложения, связанные с дальнейшим улучшением данного учебного пособия. Все замечания и предложения прошу отправлять по адресу: 111500, Республика Казахстан, г. Рудный, ул. 50 лет Октября, 38 или по электронной почте zarubin_mu@mail.ru или gali17@mail.ru.

ГЛАВА 1. ИСТОРИЯ КРИПТОГРАФИИ

Ключевые слова: примитивная криптография, формальная криптография, научная криптография, компьютерная криптография, подстановка, перестановка, сцитала, квадрат Полибия, диск и линейка Энея, атбаш, эддуба, «черный» кабинет, перлюстрация, шифр Вернама, роторная шифровальная машина, цилиндр Джефферсона, Энигма, Turing Bombe, блочный шифр, шифрование с публичным ключом, манускрипт Войнича, фестский диск, Кодекс Рохонци, кипу, вероятностное шифрование, квантовая криптография, криптография на решетках, медовое шифрование, функциональное шифрование, гомоморфное шифрование, ДНК-шифрование.

1.1 Введение в историю криптографии

«Кто владеет информацией, тот владеет миром!» – знаменитая фраза Натана Ротшильда как никогда точно отражает важность информации и ее сохранности от несанкционированного использования. Поэтому не удивителен интерес человечества к методам защиты информации от посторонних.

Одной из наиболее динамично и успешно развивающихся наук, обеспечивающих защиту информации от несанкционированного доступа, стала *криптография.*

Слово «криптография» произошло от сочетания древнегреческих слов «κρυπτός» («скрытый») и «γράφω» («пишу»).

В настоящем понимании «криптография» – это наука о методах обеспечения конфиденциальности (невозможности прочтения информации посторонним), целостности данных (невозможности незаметного изменения информации), аутентификации (проверки подлинности авторства или иных свойств объекта), а также невозможности отказа от авторства.

Мы привыкли использовать методы и средства криптографии при аутентификации для электронной почты, чатов, компьютерных игр, электронных платежных системам, защите наших данных при хранении и передаче в линиях вычислительных систем, подтверждении наших действий посредством электронно-цифровой подписи. И, зачастую, мы даже не задумываемся, как это все реализовано.

В данном разделе будет представлен материал, как появились и развивались криптографические системы от древнего мира до сверхсовременных решений.

Известная история криптографии насчитывает порядка 4 тысяч лет[1]:

– ***первый период*** (приблизительно с III-го тысячелетия до н.э.). Этот период также получил названия *наивной криптографии*. Он характеризуется господством простейших моноалфавитных шифров;

– ***второй период*** (с IX века на Ближнем Востоке и с XV века в Европе –
до начала XX века) ознаменовался введением в обиход систем полиалфавитных шифров и получил название *формальной криптографии*. Период связан с появлением формализованных и относительно стойких к ручному криптоанализу шифров. В европейских странах это произошло в эпоху Возрождения, когда развитие науки и торговли вызвало спрос на надежные способы защиты информации. В этот период в XIX веке голландец Керкгоффс сформулировал главное требование к криптографическим системам, которое остается актуальным и поныне: секретность шифров должна быть основана на секретности ключа, а не секретности алгоритма;

– ***третий период*** (30-е 60-е годы XX века) характеризуется использованием строгого математического аппарата для построения криптосистем и внедрением электромеханических устройств в работу шифровальщиков. К началу 30-х годов окончательно сформировались разделы математики, являющиеся научной основой криптологии: теория вероятностей и математическая статистика, общая алгебра, теория чисел, начали активно развиваться теория алгоритмов, теория информации, кибернетика. Своеобразным водоразделом стала работа Клода Шеннона «Теория связи в секретных системах», где сформулированы теоретические принципы криптографической защиты информации. Шеннон ввел понятия «рассеивание» и «перемешивание», обосновал возможность создания сколь угодно стойких криптосистем. В 60-х годах ведущие криптографические школы подошли к созданию блочных шифров, еще более стойких по сравнению с роторными криптосистемами, однако допускающие практическую реализацию только в виде цифровых электронных устройств;

– ***четвёртый период*** начался с середины 70-х годов XX века – это период перехода к компьютерной криптографии. Первым классом криптосистем, практическое применение которых стало возможно

[1] В качестве основного критерия периодизации криптографии современные историки используют технологические характеристики используемых методов шифрования.

из-за появления вычислительных средств, стали блочные шифры. В 70-е годы был разработан американский стандарт шифрования DES. Один из его авторов, Хорст Фейстель, предложил подходы построения блочных шифров, на основе которых в дальнейшем были построены другие, более стойкие симметричные криптосистемы.

С появлением DES обогатился и криптоанализ, для атак на данный криптоалгоритм было создано несколько новых видов криптоанализа (линейный, дифференциальный и т.д.), практическая реализация которых опять же была возможна только с появлением мощных вычислительных систем.

В середине 70-х годов произошел настоящий прорыв в современной криптографии – появление криптосистем, основанных на использовании публичного ключа. Здесь отправной точкой принято считать работу, опубликованную Уитфилдом Диффи и Мартином Хеллманом в 1976 году под названием «Новые направления в современной криптографии». В ней впервые сформулированы принципы обмена шифрованной информацией без обмена секретным ключом. Несколькими годами позже Рон Ривест, Ади Шамир и Леонард Адлеман предложили миру криптосистему RSA, основанную на использовании публичного и секретного ключей. Асимметричная криптография открыла сразу несколько новых прикладных направлений, в частности: системы электронной цифровой подписи (ЭЦП) и электронных денег. Актуальной для этого периода остается и задача совершенствования симметричных криптосистем. В 80-90-х годах разработаны ГОСТ 28147-89, нефейстелевские шифры (SAFER, RC6 и др.), а в 2000 году после открытого международного конкурса принят новый национальный стандарт шифрования США – AES.

В последние годы появились совершенно новые направления криптографии. Например:

- вероятностное шифрование Шафи Голдвассера;
- квантовая криптография[2] Стивена Визнера;
- криптография на решетках, сформулированная Сецилией Бочини;
- гомоморфное шифрование, предложенное в 1978 году Рональдом Ривестом, Леонардом Алдеманом и Майклом Дертузосом;
- медовое шифрование, представленное на конференции Eurocrypt в Копенгагене в 2015 году Ари Джулсом и Томасом Ристенпартом;

[2] Современные криптографы уже поднимают вопросы создания постквантовой криптографии.

– функциональное шифрование – сформулированная в начале 90-х годов XX века Уитфилдом Диффи и Мартином Хеллманом концепция которого считается одной из наиболее перспективных для шифрования с публичным ключом;
– ДНК-шифрование Леонарда Адлемана.

Как бы фантастически не звучали данные направления, скорее всего, они (а может быть и совершенно другие решения) будут в основе криптографии XXI века. Так, в 1989 году Беннет и Брассар в Исследовательском центре IBM построили первую работающую квантово-криптографическую систему. Осознание практической ценности этих научных изысканий, мы предполагаем, позволит выделить пятый этап развития криптографии.

1.2 Наивная криптография

История криптографии насчитывает уже более четырех тысяч лет и появилась параллельно с появлением письменности. Наверное, самыми древними центрами появления криптографических преобразований были Месопотамия, Индия, Китай. Значительно более поздним периодом характеризуются лучше изученные и освещенные методы шифрования Древней Греции.

Первым известным применением криптографии принято считать начало использования специальных иероглифов около 4000 лет назад в Древнем Египте. Элементы криптографии обнаружены уже в надписях Старого и Среднего царств (периоды III-VI и XI-XII династий фараонов), полностью криптографические тексты известны с периода XVIII династии египетских фараонов. Иероглифическое письмо произошло от пиктографии, в нем используются идеограммы (письменный знак или условное изображение, рисунок, соответствующий определённой идее автора) и, в результате отсутствия огласовки, дало возможность создавать фонограммы по принципу ребусов. Криптография египтян использовалась, скорее всего, не с целью затруднить чтение, а вероятнее, со стремлением писцов превзойти друг друга в остроумии и изобретательности, а также, с помощью необычности и загадочности, привлечь внимание к своим текстам. Одним из показательных примеров таких «криптограмм» являются тексты прославления «начальника Востока» Хнумхотепа II (XIX в. до н. э.), найденные в местности Бени-Хасан.

В древнеиндийских рукописях приводится более шестидесяти способов письма, среди которых есть и такие, которые можно рассматривать как криптографические. Имеется описание системы замены гласных букв согласными, и наоборот.

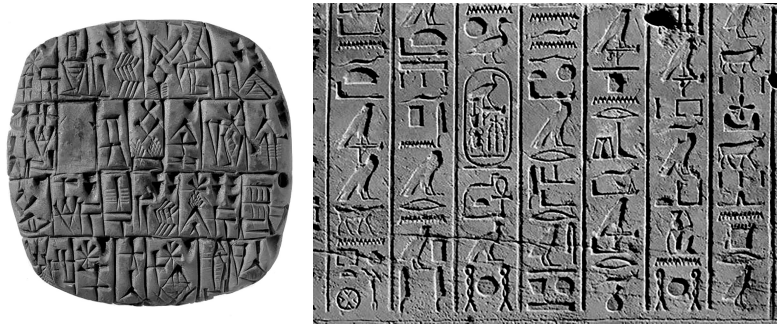


Рисунок 1.1 – Месопотамская глиняная табличка эддуба и табличка с египетскими иероглифами

Одним из древнеиндийских документов по криптографии является … Камасутра. Составленная в четвертом веке до нашей эры, она содержит в себе описание 64 искусств (йог), которыми должна овладеть каждая женщина. Среди них есть такие привычные нам умения, как приготовление блюд и напитков, искусство выбора наряда, приготовления ароматов, а также навык делать массаж. Но в ее главе 3 под номером 44 указано особое искусство «Mlecchita vikalpa», которое описывается как «искусство понимания письма в шифре, и написание слов особым способом».

В Месопотамии[3] неизвестный автор таблички с рецептом для изготовления глазури для гончарных изделий из использовал редкие обозначения, пропускал буквы, а имена заменял на цифры, чтобы скрыть написанное.

Аналогичные алгоритмы шифрования применялись и в Древнем Китае.

Однако клинопись, рисуночное и иероглифическое письмо само по себе было крайне сложно и требовало длительного обучения, так что

[3] Государства в междуречье Тигра и Ефрата (Месопотамия) нам более известны как шумеры, Вавилон и Ассирия (период с 3200 года вплоть до 100 года до нашей эры). Первые школы для подготовки писцов в Междуречье назывались «домами табличек» (по-шумерски «эдубба»). Название им дали таблички из глины, на которые наносилась клинопись.

вопрос о шифровании сообщений часто попросту не поднимался, так как количество грамотных людей было минимально. Нельзя судить и о широте распространения различных криптографических систем и тайнописи того периода, так как число дошедших до нас артефактов и записей очень невелико.

С появлением фонетического письма в II-I тысячелетии до нашей эры письменность стала значительно проще, что сделало ее более доступной. Соответственно, возросло и значение криптографии.

Одним из таких центров считают государства древней Греции и Рима. В этих государствах применялись одни из самых известных криптографических преобразований и устройств: сцитала, диск и линейка Энея, квадрат Полибия и, чуть позже, код Юлия Цезаря.

Одним из древнейших дошедших до нас криптографическим устройством является сцитала[4] (от греческого σκυτάλη «жезл»). Сцитала представляет собой жезл (цилиндр) и узкую полоску папируса или пергамента (кожи), обматывавшуюся вокруг него по спирали. После этого на нее наносился текст сообщения. При сматывании полоски с жезла буквы сообщения теряли свой порядок – происходила простейшая перестановка. Получатель сообщения для расшифрования должен быть иметь жезл такого же диаметра.

Рисунок 1.2 – Сцитала и фрагмент фрески ее применения спартанцами

По-видимому, изначально сциталу греки использовали для удобства письма (потому что в ранних упоминаниях на ней писали, в том числе, и стихи), а где-то с IV века до нашей эры стали использовать как инструмент для тайнописи.

[4] Перестановочный шифр, реализуемый посредством сциталы, также называют Спартанским шифром.

Афинянам, а точнее Аристотелю, приписывают изобретение метода расшифровки текстов, записанных с помощью сциталы. Полоску перехваченного пергамента с секретным сообщением достаточно было обернуть вокруг достаточно длинного конуса у его основания, а затем постепенно сдвигать к вершине конуса. Там, где диаметр конуса совпадал с диаметром сциталы, буквы на пергаменте сочетались в слоги и слова.

С именем Энея Тактика, полководца IV века до н. э., связывают несколько техник шифрования и тайнописи. Это два устройства – диск и линейка Энея – и книжный шифр Энея.

Диск Энея представлял собой деревянный или медный диск диаметром 10-15 сантиметров с отверстиями по числу букв алфавита. Каждому отверстию ставилась в соответствие конкретная буква. В центре диска находилась катушка с намотанной на неё бечевой. Для записи сообщения бечева протягивалась через отверстия в диске, соответствующим буквам сообщения. При чтении получатель вытягивал бечеву, и получал буквы, правда, в обратном порядке. Хотя недоброжелатель мог прочитать сообщение, если перехватит диск, Эней предусмотрел и способ быстрого уничтожения сообщения - для этого было достаточно выдернуть нить.

Первым действительно криптографическим инструментом можно назвать линейку Энея, реализующей шифр замены. Вместо диска использовалась линейка с отверстиями по числу букв алфавита, катушкой и прорезью. Для шифрования нить протягивалась через прорезь и отверстие, после чего на нити завязывался очередной узел. Для дешифрования необходимо было иметь саму нить и линейку с аналогичным расположением отверстий. Таким образом, даже зная алгоритм шифрования, но, не имея ключа (линейки), прочитать сообщение было невозможно.

Также с именем Энея Тактика связано использование малозаметных пометок в тексте документа (например, проколов иглой, поставленных рядом с буквой). Такие пометки позволяют вычленить из общего текста значимые символы – скрытый текст. Такое сокрытие информации получило имя книжный шифр Энея[5].

[5] Первое упоминание о использовании книжного шифра встречается в сочинении Энея Тактика, «О перенесении осады», где он предлагает данный метод для тайнописи. Много позже, аналогичный шифр использовали германские шпионы в Первой мировой войне.

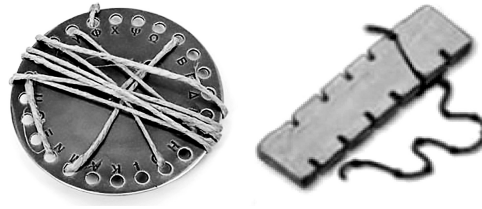


Рисунок 1.3 – Диск и линейка Энея

Одним из дошедших до нас древнегреческих подстановочных шифров был алгоритм, предложенный философом и полководцем Полибием, жившим во втором веке до нашей эры, получивший название «квадрат Полибия». Ключ алгоритма представлял из себя квадрат 5x5 клеток, в которые в произвольном порядке записывались буквы греческого алфавита. В шифрсообщении, обычно передаваемом гелиографом, передавались последовательно номера строк и номера столбцов букв. При получении сообщения для его расшифрования требовался точно такой же квадрат – ключ.

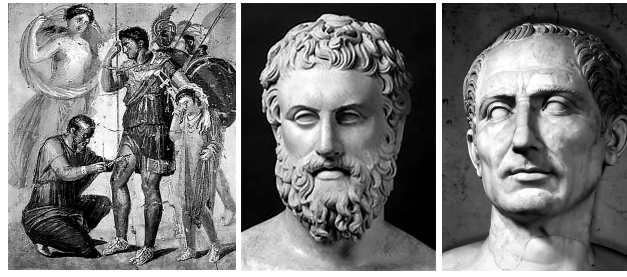


Рисунок 1.4 – Эней Тактик (IV век до н.э.), Полибий (206-124 годы до н.э.), Гай Юлий Цезарь (100-44 годы до н.э.)

Наверное, самым известным стал криптоалгоритм, используемый римским патрицием и великим понтификом Гаем Юлием Цезарем, для шифрования своей переписки со своими генералами и друзьями. Юлий Цезарь, согласно «Жизни двенадцати цезарей» Светония, использовал его со сдвигом 3. Хотя Цезарь был первым зафиксированным человеком, использующим эту схему, другие шифры подстановки, как известно, использовались и ранее. Также сохранились доказательства, что Цезарь использовал и более сложные шифры.

Отдельно стоит упомянуть и шифр «Атбаш». Предполагается, что шифр был изобретен Ессеями, иудейской сектой повстанцев, с целью защиты от раскрытия властями своих последователей и избежания казней. Знания этих кодов и шифров потом перенял орден Тамплиньеров. Таким образом, шифр атбаш был использован на протяжении многих сотен лет (от около 500 до н.э. до 1300 года н.э.).

Уже тогда шифрованная переписка использовалась не только государственными деятелями и полководцами, но и церковью, и учеными. Жрецы шифровали тексты прорицателей, а ученые – свои труды. Например, у Э.Шюре в книге «Великие посвященные» встречается фраза о том, что «с великим трудом и большой ценой добыл Платон один из манускриптов Пифагора, который никогда не записывал свое учение иначе, как тайными знаками и под различными символами».

К сожалению, письменность большинства американских и африканских народов того времени не дошла до наших дней, поэтому вопросы использования криптографических методов в этих странах сейчас можно отнести к загадкам истории.

Подводя итог, можно сказать, что для всех алгоритмов наивной криптографии (до начала XVI века) характерно использование любых (обычно примитивных) способов запутывания противника относительно содержания шифруемых текстов. На начальном этапе для защиты информации использовались методы кодирования и стеганографии, большинство же используемых алгоритмов шифрования сводились к перестановке или моноалфавитной подстановке (смотрите главу 2).

1.3 Формальная криптография

Формальная криптография связана в большей степени с историей Европы эпохи Возрождения[6]. В этот период активно развивается как государственная, так и частная переписка. Естественно, это порождает и бурное развитие всевозможных криптографических методов защиты этой переписки от посторонних глаз.

Наиболее характерными для данного периода являются опять же моноалфавитные шифры подстановки и шифры перестановки. В этот период появляются и первые научные труды по криптографии.

---

[6] До эпохи Возрождения в христианской Европе криптография считалась «темным» искусством и смешивалась с Каббалой.

Первой известной в настоящее время европейской книгой, описывающей использование криптографии, считается труд Роджера Бэкона «Послание монаха Роджера Бэкона о тайных действиях искусства и природы и ничтожестве магии», описывающий, в числе прочего, применение 7 методов скрытия текста. Ему же приписывают авторство и таинственного манускрипта Войнича.

В XIV веке сотрудник тайной канцелярии папской курии Чикко Симонети пишет книгу о системах тайнописи, а в XV веке секретарь папы Климентия XII Габриэль де Левинда, заканчивает работу над «Трактатом о шифрах».

Очередной известный результат принадлежит перу германского аббата Иоганна Тритемия (или Тритемуса), которого многие историки считают вторым отцом современной криптологии. Он становится автором первой печатной книги по криптографии. В пятой книге серии «Polygraphia», изданной в 1518 году, он описал шифр, в котором каждая следующая буква шифруется своим собственным шифром сдвига. Его подход был улучшен Джованом Баттистой Белласо. Кроме этого, Тритемий первым заметил, что шифровать можно и по две буквы за раз – биграммами (смотрите главу 2).

Следующий шаг в развитии криптографии сделан Джованни Порта, известным итальянским естествоиспытателем. В 1563 году он написал книгу «О тайной переписке», в которой приводится описание всех известных систем шифров. В ней дается и описание биграммного шифра, в котором осуществляется замена пар букв. Порта предвосхитил то, что называют «методом вероятного слова» и приводит примеры списков вероятных слов из различных областей.

В этот же период появляется первая организация, посвятившая себя целиком криптографии. Она была создана в Венеции в 1452 году. Три секретаря этой организации занимались взломом и созданием шифров по заданиям правительства.

В 1626 году, при осаде города Реальмон во время восстания гугенотов во Франции, а позже и в 1628 году при осаде Ла-Рошели, Антуан Россиньоль (1600-1682) расшифровал перехваченные сообщения и тем самым помог победить королю. После победы правительство Франции несколько раз привлекало его к расшифровке шифров. После смерти Россиньоля его сын, Бонавентур, а позже и внук, Антуан-Бонавентур Россиньоль, продолжили его дело, которое в дальнейшем вылилось в создание специальной службы перлюстрации и криптоанализа во

Франции – так называемых чёрных кабинетов[7]. В Англии также был свой «черный кабинет». В его работе в XVII в. заметное место занимал Джон Валлис, известный как крупнейший английский математик до Исаака Ньютона. В Германии начальником первого дешифровального отделения был граф Гронсфельд, создавший один из вариантов усовершенствования шифра Виженера. В Российской империи в это время создается для этих же целей цифирная палата.

Рисунок 1.5 – Роджер Бэкон (1220-1292), Антуан Россиньоль (1600-1682), Джон Валлис (1616-1703)

Более развитой и известной для периода формальной криптографии является арабская криптография. Уровень развития математики и других наук в этот период значительно опережал знания народов Европы. Поэтому период развития и сложность шифров арабского мира до X-XII веков значительно опережал Европу. Даже слово шифр – арабского происхождения.

Наиболее древней считается научная работа арабского ученого Абу Бакр Ахмед бен-Али бен-Вахшия ан-Набати датируемая 855 годом. В ней упоминаются различные системы шифрования основанные на символах древних народов. Данные шифры применялись вплоть до начала XIX века для шифрования секретной переписки, донесений шпионов, трактатов по черной магии.

[7] В 1911 году энциклопедия Британника писала, что «чёрные кабинеты» больше не существуют, однако фактически, в той или иной форме службы перлюстрации и дешифровки переписки существовали и в тот момент, и позже, несмотря на существующие законы о тайне переписки.

Познания арабов в области криптографии были изложены Шехабехом Калкашанди, автором энциклопедии, написанной в 1412 году. Он включил целый раздел, посвященный использованию систем шифрования, основанных как на перестановках, так и на замене (в том числе и множественной) символов. Кроме того, большое внимание было уделено вскрытию зашифрованных посланий[8].

Рисунок 1.6 – Кабинет перлюстрации при почтовом отделении в Российской империи (XIX век), аналог французских «черных кабинетов».

В Европе и Азии в этот период получают широкое распространение шифры, называемые *номенклаторами*. Они объединяют в себе шифры простой замены и кодирование. В простейших номенклаторах код состоял из нескольких десятков слов или фраз с соответствующими им двухбуквенными кодовыми обозначениями.

Также на смену моноалфавитным шифрам появляются полиграммые и полиалфавитные подстановочные шифры (смотрите главу 2), которые впервые предложил использовать Леон Баттиста Альберти. Также Альберти предложил устройство из двух скреплённых в центре дисков, каждый из которых имел алфавит, написанный по краю, и мог поворачиваться относительного другого диска. Пока диски не двигаются, они позволяют шифровать с использованием шифра Цезаря, однако через несколько слов диски поворачиваются, и меняется ключ сдвига.

[8] Предложенные методы основывались на статистических и лингвистических свойствах языка. На основании текста Корана в энциклопедии был приведена статистика всех символов арабского языка и пример вскрытия чужого сообщения.

В 1883 году криптология получила новые идеи, изложенные в труде под названием «Военная криптография» Огюста Керкгоффса. Опираясь на свои знания в области лингвистики и математики, Кергоффс проводит сравнительный анализ шифров, на основе которого формулирует требования к шифрам и делает вывод, что практический интерес представляют только те шифры, которые остаются стойкими даже при интенсивной переписке.

Керкгоффс сформулировал принцип ставший основой современной криптологии – стойкость криптографической системы должен зависеть не от секретности алгоритма шифрования, а от криптостойкости и секретности используемого ключа. Этот принцип не потерял своей актуальности и сегодня.

Не менее ценна мысль Керкгоффса о том, что надежность шифра должны оценивать дешифровальщики. Разумеется, об этом догадывались и до него, но после закрытия «черных кабинетов» как-то позабыли. Во всяком случае, изобретатели новых шифров, вместо того чтобы вынести их на суд криптоаналитиков, стремились оценить их стойкость самостоятельно, подсчитывая число веков, необходимых для последовательного перебора всех возможных ключей, или старались доказать невозможность «пробить» какой-либо из элементов шифра.

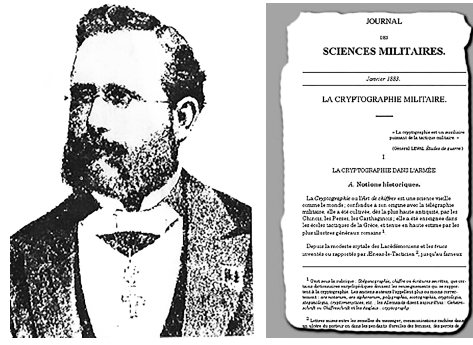

JOURNAL
DES
SCIENCES MILITAIRES.
Janvier 1883.
LA CRYPTOGRAPHIE MILITAIRE.
I
LA CRYPTOGRAPHIE DANS L'ARMÉE
A. Notions historiques.

Рисунок 1.7 – Жан Вильгельм Губерт Виктор Франсуа Александр Огюст Керкгоффс фон Ниувенгоф (1835-1903) и «Военная криптология»

Керкгоффс писал: «Я поражен тем, что наши ученые и профессора преподают и рекомендуют для применения в военное время системы,

ключи к которым, несомненно, менее чем за час откроет самый неопытный криптоаналитик... Можно также полагать, что отсутствие серьезных работ по искусству прочтения тайнописи способствовало распространению самых ошибочных идей о стойкости наших шифрсистем».

Благодаря работам Кергоффса, во всех ведущих государствах мира уже в 80-х годах XIX века криптографию признают наукой и в обязательном порядке начинают преподавать в военных академиях.

Конец XIX и начало XX века характеризуется массовым появлением сначала телеграфа, а затем и радио для передачи информации. Естественно, возросли и возможности перехвата информации. Поэтому данный период развития характеризуется и появлением шифров и систем, ориентированных для передачи информации посредством электрической и радиосвязи.

Для шифрования телеграфных сообщений разрабатывается целая плеяда кодов и шифроалгоритмов, апогеем которых становится появление абсолютно надежного шифра – шифра Вернама.

Кроме этого, для шифрования телетайпных сообщений Вернам предложил заранее готовить «гамму» – перфоленту со случайными знаками – и затем электромеханически складывать ее импульсы с импульсами знаков открытого текста. Полученная сумма представляла собой шифртекст. На приемном конце импульсы, полученные по каналу связи, складывались с импульсами той же самой «гаммы», в результате чего восстанавливались исходные импульсы сообщения. А если сообщение перехватывалось, то без «гаммы» расшифровать его было невозможно, противник видел только ничего не значащую последовательность «плюсов» и «минусов».

Во время Первой мировой войны главным (и зачастую единственным) средством шифрования были коды. Несмотря на то, что все участники боевых действий постоянно разрабатывали новые коды и улучшали старые, обеспечить их сохранность удавалось далеко не всегда, поэтому противники зачастую были полностью осведомлены обо всем, что содержалось в чужой секретной переписке. С применением шифров связан ряд трагических событий, из которых упомянем лишь разгром двух русских армий – генералов Ранненкампа и Самсонова в Восточной Пруссии в августе 1914 года. Причиной разгрома стала в том числе и плохая организация закрытой связи, в результате чего переговоры по радио велись вообще без всякого шифрования.

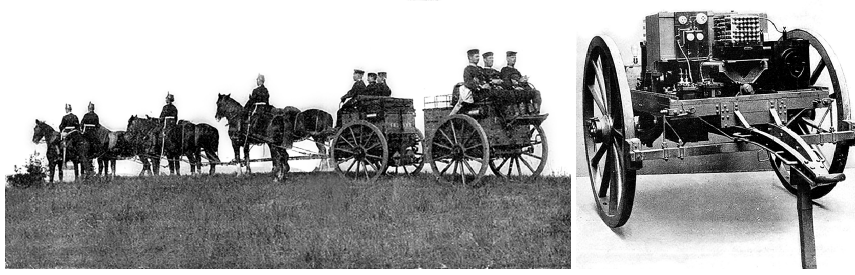


Рисунок 1.8 – Конная малая (полевая) военная радиостанция с боевым расчетом начала XX века

Вторая мировая война еще больше подняла требования к скорости передачи и секретности информации. Ручные алгоритмы преобразования и коды, еще кое-как решали проблемы нелегальной разведки, но абсолютно были неприменимы для воюющих армий и флота. Требовались повышения криптостойкости шифров и автоматизация (точнее механизация) процессов шифрования.

Одной из первых подобных систем стала изобретенная еще в 1790 году Томасом Джефферсоном, будущим президентом США, механическая машина –
цилиндр Джефферсона.

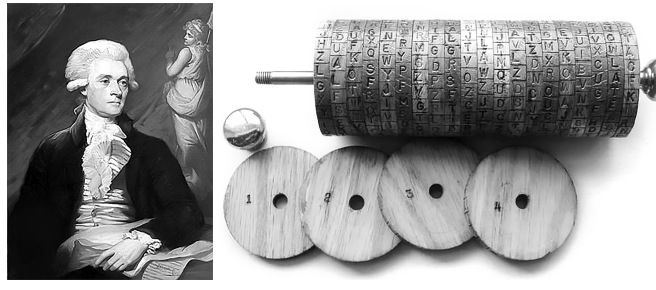


Рисунок 1.9 – Томас Джефферсон (1743-1826) и цилиндр Джефферсона

Джефферсон назвал свою систему шифрования «дисковым шифром». Однако сам он не был уверен в надежности своего изобретения, поэтому относился к нему с осторожностью и, будучи президентом США, не использовал его, а продолжил применять традиционные коды и шифры.

При использовании цилиндра Джефферсона достаточно повернуть диски так, чтобы на гранях дисков в строчке появился нужный исходный текст, а записать строку с любой другой грани. После получения шифросообщения и набора его на цилиндре Джефферсона на одной из граней появлялся исходный текст. Естественно, цилиндры отправителя и получателя должны быть идентичными.

Однако практическое распространение механические шифраторы получили только в начале XX века. Одной из первых практически используемых машин, стала роторная машина, разработанная в 1917 году Эдвардом Хеберном. А 23 февраля 1918 года немецкому инженеру Артуру Шербиусу был выдан патент на шифровальную машину «Энигму», которая и стала легендой роторных шифровальных машин.

«Энигма» вначале представляла собой четыре вращающихся на одной оси барабана, что обеспечивало более миллиона вариантов шифра простой замены. На каждой стороне барабана по окружности располагались 25 электрических контактов (по количеству букв). Контакты с обеих сторон барабана соединялись попарно случайным образом 25 проводами, формировавшими замену символов. Колеса складывались вместе и их контакты, касаясь друг друга, обеспечивали прохождение электроимпульсов сквозь весь набор колес. Перед началом работы барабаны поворачивались так, чтобы устанавливалось заданное кодовое слово. При нажатии клавиши и кодировании очередного символа правый барабан поворачивался на один шаг. После того, как барабан делал полный оборот, на один шаг поворачивался следующий барабан. Таким образом, получался ключ заведомо гораздо более длинный, чем текст сообщения.

Рисунок 1.10 – Эдвард Хуг Хеберн (1869-1952), Артур Шребиус (1878-1929) и роторная шифровальная машина «Энигма»

Впервые шифр «Энигмы» удалось дешифровать в польском Бюро шифров в декабре 1932 года. Мариан Реевский, Ежи Ружицкий, Генрих Зыгальский и Мариан Реевский, с помощью данных французской разведки, математической теории и методов обратной разработки смогли разработать специальное устройство для дешифровки закодированных сообщений, которое назвали криптологической бомбой. После этого немецкие инженеры усложнили устройство «Энигмы» и в 1938 году выпустили обновленную версию, для дешифровки которой требовалось построить более сложные механизмы.

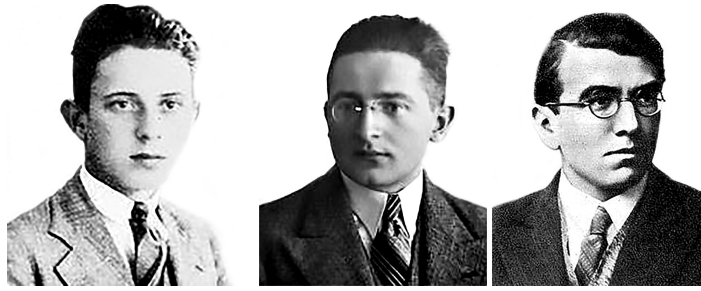


Рисунок 1.11 – Ежи Ружицкий (1909-1942), Мариан Адам Реевский (1905-1980), Генрик Зыгальский (1908-1978)

Во время Второй мировой войны в Англии для расшифровки сообщений «Энигма», была создана машина с кодовым названием «Turing Bombe», оказавшая значительную помощь антигитлеровской коалиции.

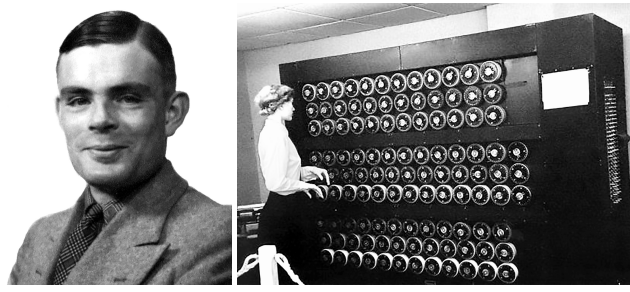


Рисунок 1.12 – Алан Тьюринг (1912-1954) и электронный вычислитель «Turings Bomb»

Роторные машины активно использовались во время второй мировой войны и другими государствами. Помимо немецкой машины использовались также устройства «Sigaba» (США), «Турех» (Великобритания), «91-shiki ohbun-injiki» (Япония) и многие другие. Роторные системы стали вершиной формальной криптографии, так как относительно просто реализовывали достаточно криптостойкие шифры.

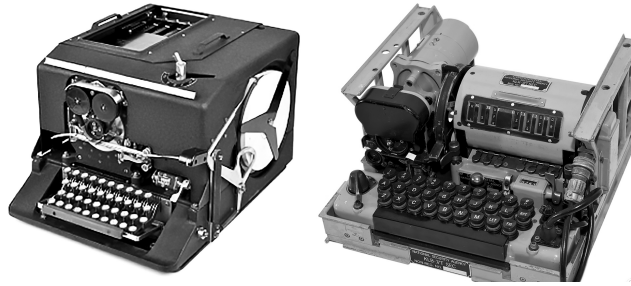


Рисунок 1.13 – Американские роторные шифровальные машины SIGABA (ECM MARK II) и ее наследница KL-7[9]

С зарубежными машинами, которые работали по принципу механически программируемых дисковых шифраторов, техника СССР имела мало чего общего. Шифровальная машина, разработанная в 1934 году, М-100 «Спектр» и ее потомки работали по принципу наложения гаммы на открытый текст. Она позволяла шифровать со скоростью до 300 символов в минуту. Только шифровальная машина К-37 «Кристалл», разработанная в СССР в 1939 году, была построена по аналогии с другими роторными машинами.

При этом случаев расшифровки сообщений, зашифрованных М-100 и М-101, не зафиксировано[10]. Сообщения же, зашифрованные К-37,

[9] KL-7 оставалась в строю до 1970 годов. В некоторых странах в течение многих лет машины KL-7 служили в качестве резервных устройств, пока окончательно не были изъяты из службы в 1983 году.

[10] Известно, что в период с 1941 по 1947 год на базе уфимского ГСПЭИ 56 и ряда других заводов в общей сложности выпустили 2024 шифратора речи. За годы войны восьмым (шифровальным) управлением Генштаба СССН было разослано около 3,2 миллиона комплектов шифров. Было передано в общей сложности свыше полутора миллионов шифрованных телеграмм и кодограмм. Как пишет автор нескольких книг по криптографии Дмитрий Ларин, «нагрузка на каналы связи порой достигала 1500 телеграмм в сутки».

США на регулярной основе «читали» с апреля 1946-го по 1947-й год, автоматизировав этот процесс с помощью машины аналога Sauterne Mark I.

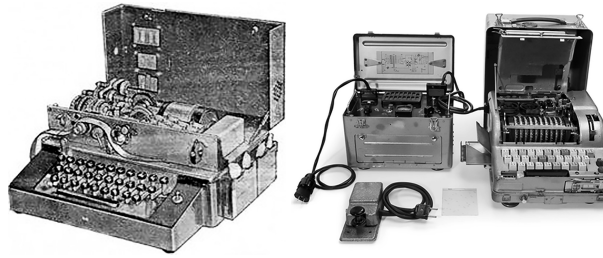


Рисунок 1.14 – Советская шифровальная машина М-100 «Спектр» и ее потомок – кодировочная машина «Фиалка-125»

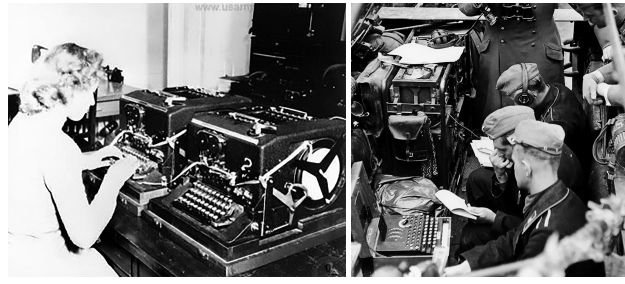


Рисунок 1.15 – Работа шифровальщицы в США на роторной машине Sigaba и шифровальщиков вермахта на машине «Энигма» в окопах

1.4 Научная криптография

После Первой мировой войны правительства практически всех ведущих стран засекретили все работы в области криптографии. К началу 1930-х годов окончательно сформировались разделы математики, являющиеся основой для будущей науки: общая алгебра, теория чисел, теория вероятностей и математическая статистика. К концу 1940-х годов построены первые программируемые счётные машины, заложены основы теории алгоритмов, кибернетики. Тем не менее, в период после Первой мировой войны и до конца 1940-х годов в открытой печати было опубликовано минимум работ и монографий, но даже они отражали далеко не самое актуальное состояние дел. Наибольший прогресс в криптографии достигался в военных ведомствах.

Ключевой вехой в развитии научной криптографии является фундаментальный труд Клода Шеннона «Теория связи в секретных системах» (англ. Communication Theory of Secrecy Systems) – секретный доклад, представленный автором в 1945 г., и опубликованный им в «Bell System Technical Journal» в 1949 г. В этой работе, по мнению многих современных криптографов, был впервые показан подход к криптографии в целом как к математической науке.

Клод Шеннон ввел понятия «рассеивание» и «перемешивание», обосновал возможность создания практически сколь угодно стойких криптосистем.

Шеннон доказал, что предложенный Вернамом в 1917 году метод шифрования – абсолютно стойкая система шифрования. Естественно, при условии, что длина ключа равняется или больше длины сообщения.

Этот период также характеризуется постепенным выводом электромеханических роторных шифровальных машин из эксплуатации правительственными и армейскими структурами и переход на электронные шифровальные машины, например, такие как KW-26, KW-37, KL-51 (RACE) и Aroflex в США и странах блока НАТО.

Рисунок 1.16 – Электронная шифровальная машина Aroflex (США) и портативная электронная шифромашина HG-530/535 семейства CRIPTOMATIC 500 Бориса Хагелина (Швейцария)

В 1960-х годах начали появляться различные блочные шифры, которые обладали большей криптостойкостью по сравнению с результатом работы роторных машин. Однако они предполагали обязательное использование цифровых электронных устройств – ручные или полумеханические способы шифрования уже не использовались.

1.5 Компьютерная криптография

Практически все современные применяемые криптосистемы являются разумно стойкими, то есть стойкость этой криптосистемы сегодня оценивается объемами вычислений, которые требуются для ее вскрытия. Считается, что ключ шифрования достаточно стоек, если все известные способы его поиска настолько сложны, что требуют больше времени, чем простой перебор всех возможных ключей. А период его нахождения превышает срок жизни защищаемой информации (или затраты больше стоимости получения данной информации другими путями).

Первым классом компьютерных криптосистем, практическое применение которых стало возможно с появлением компактных вычислительных средств, стали блочные шифры.

В 70-е годы XX века сделано множество работ по стандартизации шифров. Наверное, одними из первых в этом направлении были криптографы США. Сотрудниками фирмы INTEL разработан алгоритм, ставший впоследствии американским стандартом шифрования DES. Один из его авторов, Хорст Фейстель, описал модель блочных шифров, на основе которой в дальнейшем были построены другие, более стойкие симметричные криптосистемы, в том числе советский и российский стандарт шифрования ГОСТ 28147–89 и современный стандарт AES.

С появлением DES обогатился и криптоанализ, для атак на американский алгоритм был создано несколько новых видов криптоанализа (линейный, дифференциальный и т.д.), практическая реализация которых опять же была возможна только с появлением мощных вычислительных систем.

Также особо следует отметить, что в середине 70-х годов XX столетия произошел настоящий прорыв в современной криптографии – появление криптосистем с двумя ключами – секретным и публичным. С их появлением стала менее значимой проблема распространения ключей шифрования. Такие системы также получили название *асимметричных криптосистем.*

Отправной точкой в ассиметричной криптографии принято считать работу, опубликованную Уитфилдом Диффи и Мартином Хеллманом в 1976 году под названием «Новые направления в современной криптографии». В ней впервые сформулированы принципы обмена шиф-

рованной информацией без обмена секретным ключом. Независимо к идее асимметричных криптосистем подошел Ральф Меркли.

Несколькими годами позже Рон Ривест, Ади Шамир и Леонард Адлеман разработали алгоритм RSA, первую практическую асимметричную криптосистему, стойкость которой была основана на проблеме факторизации больших простых чисел. Асимметричная криптография открыла сразу несколько новых прикладных направлений, в частности, *системы электронной цифровой подписи* (ЭЦП) и *электронных денег*.

Рисунок 1.17 – Ральф Меркли и Уилфрид Диффи с Мартином Хеллманом после вручения премии Тьюринга за фундаментальный вклад в развитие криптографии в 2015 году

Рисунок 1.18 – Разрвботчики алгоритма RSA Рон Ривест, Ади Шамир и Леонард Адлеман[11]

[11] Интересный факт: криптографический алгоритм RSA, созданный Рональдом Ривестом, Ади Шамиром и Леонардом Адлеманом, как уже отмечалось, базируется на работе Уитфилда Диффи и Мартина Хеллмана «Новые направления в криптографии». При этом премию Тьюринга Ривест, Шамир и Алдеман за алгоритм RSA получили в 2002 году, а Диффи и Хеллман – только в 2015 году.

Актуальной задачей этого периода остается и задача совершенствования симметричных криптосистем. В этот же период разработаны уже нефейстелевские шифры (SAFER, RC6 и др.), а в 2000 году после открытого международного конкурса принят новый национальный стандарт шифрования США – AES.

1.6 Неизвестная криптография

Несмотря на фантастические успехи в криптографии и криптоанализе, основанные на применении вычислительной техники и систем искусственного интеллекта, человечеству известен ряд артефактов, которые ломают все аксиомы и вроде бы успешно доказанные гипотезы.

По нашему мнению, повествование о достижении криптографии и криптоанализа будут неполными без информации о таинственных загадках прошлого и нерасшифрованных криптограммах.

Наверное, самой известной из таких загадок является «Манускрипт Войнича» – иллюстрированная рукопись датирована XV веком и названа так по имени польско-литовского библиофила и антиквара Михаила Леонардовича Войнича. Необычную 240-страничную книгу он купил на вилле Мондрагоне близ Рима в 1912 году во время секретной распродажи архива библиотеки иезуитского колледжа.

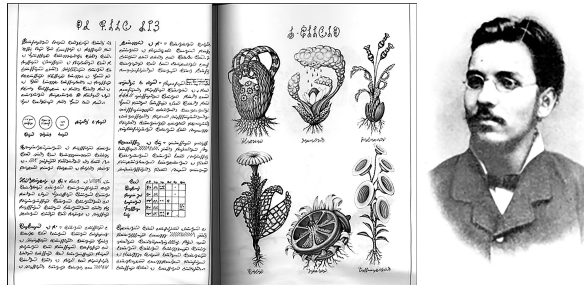


Рисунок 1.19 – Страница «манускрипта Войнича» и сам Вильфред Войнич (1865-1930)

Книга написана на неизвестном языке неизвестными символами (алфавит содержит более 50 уникальных символов) и содержит диаграммы и рисунки неизвестных животных и растений. Проведенный в Аризонском университете радиоуглеродный анализ позволяет

утверждать, что манускрипт написан на пергаменте между 1404 и 1438 годами, а анализ ее чернил, выполненный в McCrone Research Institute в Чикаго, подтвердил датировку. Семантический и частотный анализ позволяет говорить о том, что язык, на котором написан манускрипт, имеет сходные характеристики с латынью и европейскими языками. Неоднократно проведенные исследования позволяют утверждать и то, что книга содержит осмысленную информацию.

Общее впечатление, которое создают страницы манускрипта, позволяет исследователям предполагать, что он предназначался для того, чтобы служить фармакологическим или медицинским справочником или энциклопедией. Однако сбивающие с толку детали иллюстраций питают множество теорий о происхождении книги, содержании её текста и цели, для которой она была написана.

С большой долей уверенности можно сказать, что первая часть книги посвящена травам, но попытки сравнить их с реальными образцами трав и со стилизованными рисунками трав того времени в целом провалились[12]. Остальные разделы условно называют астрономическим, биологическим, космологическим, фармацевтическим и рецептным.

Существует множество гипотез о происхождении манускрипта – от того, что книга написана на украинском языке как справочник по алхимии, до того, что книга является копией документа из другого мира.

Практически весь XX век ученые различных направлений (начиная с лингвистов и историков и заканчивая физиками и криптоаналитиками) пытались разгадать загадку манускрипта. Большинство исследователей предполагает, что он представляет собой некую шифровку, и сегодня десятки научных групп пытаются расшифровать его, используя как системы искусственного интеллекта, так и методы исторических наук[13].

С 1969 года манускрипт хранится в библиотеке редких книг Бейнеке Йельского университета. Книга полностью оцифрована, поэтому любой

[12] К 2014 году ученые смогли предположительно идентифицировать 37 из 303 изображённых в рукописи растений.

[13] По данным на январь 2018 года ученым из Альбертского университета в Канаде при использовании искусственных нейроподобных сетей удалось расшифровать одну из фраз книги. Исследователи обнаружили, что автор рукописи для своего шифра изменил порядок букв в каждом слове, а гласные отбросил. Фраза звучит как «Она дала советы священнику, хозяину дома, мне и людям», хотя сейчас этот перевод оспаривается.

желающий может попробовать расшифровать загадочные графические элементы и буквы.

Менее известны, чем манускрипт Войнича, но не менее таинственны такие документы как, например, Кодекс Рохонци и Волшебные скрижали из Касселя.

Кодекс Рохони представляет собой книжку «карманного формата», содержащую 448 страниц, испещренных некими письменами-символами. Количество уникальных знаков, используемых в Кодексе, примерно в десять раз больше, чем в любом известном алфавите. Кое-где на страницах имеются иллюстрации, содержащие не только религиозные, но и вполне бытовые сюжеты.

Изучение бумаги Кодекса Рохонци показало, что она, вероятнее всего, была изготовлена в Венеции в начале XVI века. Расшифровать Кодекс пока никому не удалось, есть только некоторые версии.

Рисунок 1.20 – Пара страниц Кодекса Рохонци

Следующим классом неразгаданных древнейших шифрограмм является Фестский диск, линейное письмо и ронго-ронго.

В 1908 году итальянский археолог Луиджи Пернье обнаружил небольшой глиняный диск при раскопках на месте древнего Критского города Фест. На диске изображены 242 символа. Специалисты сумели различить 45 видов символов, но из них лишь несколько опознаны как иероглифы, которые использовались в додворцовом периоде древней истории Крита.

На данный момент, несмотря на множество попыток, никому не удалось разгадать тайну Фестского диска[14].

[14] Некоторые ученые полагают, что фестский диск является астрономическим календарем, другие считают, что он родом из легендарного затонувшего города Атлантиды.

Линейное письмо[15] также было найдено на Крите и названо в честь британского археолога Артура Эванса. В 1952 году Майкл Вентрис частично расшифровал линейное письмо, которое использовалось для шифровки микенского языка.

Рисунок 1.21 – Фестский диск и линейное письмо

Ронго-ронго – система таинственных записей, которые обнаружили на острове Пасхи в XIX веке. Считается, что Ронго-ронго представляет собой утраченную систему протописьма. Многочисленные попытки расшифрования Ронго-ронго оказались безуспешными. Возможно, это дало бы ответ на главную загадку острова – предназначение гигантских статуй острова Пасхи.

Таинственным и только фрагментарно расшифрованным является и узелковое письмо инков[16].

---

[15] Использовалось в Древней Греции в XIX – XV веках до н.э.

[16] В древности во многих регионах Земли процветало так называемое узелковое письмо, которое по легендам принесли на Землю белые Боги. Узелковое письмо было распространено у разных народов: в государствах инков и майя, в Китае, Австралии, Тибете, Калифорнии, Западной и Центральной Африке, на островах Рюкю, Палау, Хайнань.

Китайская узелковая письменность упоминается в трактате Дао дэ цзин («Книге пути и достоинства»), написанном древнекитайским философом Лао-Цзы в VI-V вв. до н.э. В качестве носителя информации выступают связанные между собой шнуры, а саму информацию несут узелки и цвета шнурков.

Также узелковое письмо или Узелковая Вязь является отображением славянского алфавита «глаголица».

Имеется ряд легенд о громадных золотых кладах «сыновей Солнца», которые инки спрятали от испанских конкистадоров[17], а информация об их местонахождении сохранилась в перехваченных кипу инков.

Также к древним нерасшифрованным криптограммам можно отнести надпись Шагборо. Будоражит криптографов и лингвистов, перехваченный из космоса радиосигнал, получивший название «Wow». До сих пор не раскрыты уголовные дела, где с телами умерших насильственной смертью людей найдены криптограммы – дело «Таман Шуд» и записки Рикки Маккормика.

К сожалению, формат учебного пособия не позволяет даже осветить наиболее известные из таких криптографических загадок. История оставляет будущим криптоаналитикам множество интереснейших задач, ответы на которые приоткроют тайну на таинственные события прошлого.

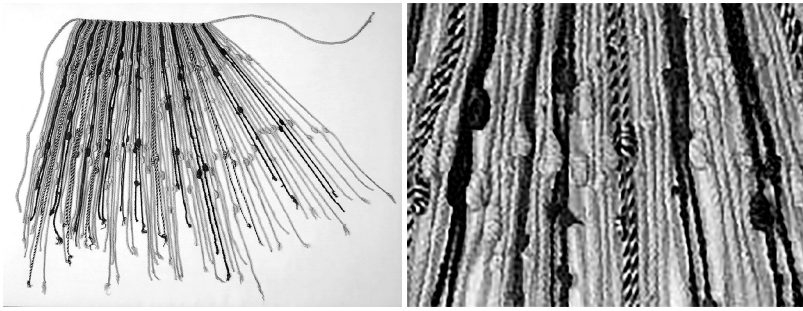


Рисунок 1.22 – Кипу инков и ее увеличенный фрагмент

[17] Например, в 1533 году отряд конкистадоров под предводительством Франко Писарро вторгся в империю Великого Инки. Захватив одного из наследников Великого Инки – Атауальпу – Писарро потребовал выкуп – заполнить огромную комнату золотом до уровня поднятой руки. Золото везли со всей империи инков, однако оговоренной высоты до окончания установленного срока достигнуть не успел. Инка попросил подождать еще немного, но вечером 26 августа 1553 года на площади в Куско Атауальпа был повешен. Перед смертью Инка передал верным людям кипу. На нем было завязано тринадцать узелков, кроме них, к шнуру Атауальпа привязал брусочек золота. Считают, что именно в этом послании было зашифровано распоряжение спрятать сокровища инков в каком-то тайном месте. В тот же день из храмов исчезли все сокровища, в том числе золотая цепь длинной в 350 шагов, со звеньями толщиной в руку, весившая столько, что поднять ее могли только 200 человек. Пропали и одиннадцать тысяч лам, груженных золотом и шедших в столицу империи инков с выкупом для Атауальпы.

Вопросы для самоконтроля

1. На какие исторические периоды можно разделить историю криптографии?
2. Какое устройство относят к первым европейским устройствам для криптографии?
3. Какой алгоритм шифрования предложил Гай Юлий Цезарь?
4. Что такое «атбаш»?
5. Кем были предложены принципы полиграммной подстановки?
6. Что такое «черный кабинет»?
7. На каких принципах построена шифровальная машина «Енигма»?
8. Для чего использовалось устройство М100 «Кристалл»?
9. Когда были заложены принципы ассиметричной криптографии?
10. О чем манускрипт Войнича?

Рекомендуемая литература

1. Амиров А.Ж., Султанова Б.К., Шаханов Д.Ж. История развития криптологии. Этапы. – Молодой ученый. 2016. – №1.
2. Амиров А.Ж., Султанова Б.К., Шаханов Д.Ж. История развития криптологии. Этапы. – Молодой ученый. 2016.
3. Спивак С.И., Вильданов А.Н., Зарипова Л.И. Достижения и приложения современной информатики, математики и физики: материалы III Всероссийской научно-практической заочной конференции (г. Нефтекамск, 20-22 октября 2014 г.). – Уфа: РИЦ БашГУ, 2014.
4. Luciano D., Prichett G. Cryptology: From Caesar Ciphers to Public-Key Cryptosystems. – The College Mathematics Journal. – Mathematical Association of America, 1987. – Vol. 18, Iss. 1.
5. Сингх С., Книга шифров. Тайная история шифров и их расшифровки. – М.: Астрель, 2007. – 448 с.
6. Соболева Т.А. История шифровального дела в России. – М.: ОЛМА-ПРЕСС Образование, 2002.
7. Бабаш А.В., Шанкин Г.П. Криптография (аспекты защиты). – М .: СОЛОН-ПРЕСС, 2007. – 512 с.
8. Чмора А. Л. Современная прикладная криптография. – М.: «Гелиос АРВ», 2001.
9. Gardner M. A new kind of cipher that would take millions of years to break. – Mathematical Games, Scientific American, 1978. – 237(2).

10. Rivest R.L., Shamir A., Adleman L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. – Comm. ACM, 1978. – 21(2).

ГЛАВА 2. СИММЕТРИЧНЫЕ КРИПТОСИСТЕМЫ

Ключевые слова: симметричная криптосистема, подстановка, перестановка, гаммирование, алгоритм шифрования, ключ, раунд, лавинный эффект, ключ прогона, расписание ключей, сеть Фейстеля, криптографическая стойкость, блочный шифр, потоковый шифр.

2.1 Основные классы симметричных криптосистем

Симметричные криптосистемы[1] – способ шифрования, в котором для шифрования и расшифровывания применяется один и тот же криптографический ключ. Ключ алгоритма должен сохраняться в секрете обеими сторонами. Алгоритм шифрования выбирается сторонами до начала обмена сообщениями.

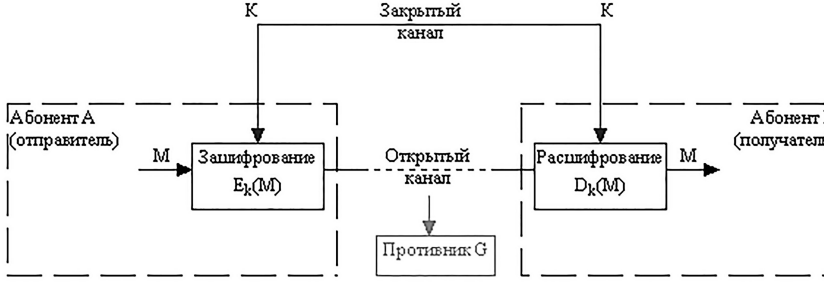


Рисунок 2.1 – Общая схема работы симметричных криптосистем

В настоящее время современные симметричные шифры делят на блочные шифры и потоковые:

- ***блочные шифры*** – обрабатывают информацию блоками определённой длины (обычно 32, 64, 128 или 256 бит), применяя к блоку ключ в установленном порядке, как правило, несколькими циклами перемешивания и подстановки, называемыми *раундами*. Результатом повторения раундов является лавинный эффект – нарастающая потеря соответствия битов между блоками открытых и зашифрованных данных;
- ***потоковые или поточные шифры*** – шифрование проводится над каждым битом либо байтом исходного (открытого) текста с использованием гаммирования.

[1] Такие системы также могут называться «симметричное шифрование», «симметричные шифры» или на английском «symmetric-key algorithm».

В основе симметричных криптосистем лежат три базовые операции преобразования сообщения или открытого текста:

– подстановки;
– перестановки;
– гаммирование.

Большинство современных симметричных шифров используют сложную комбинацию большого количества подстановок и перестановок. Многие такие шифры исполняются в несколько (иногда до 80) проходов, используя на каждом проходе «ключ прохода». Множество «ключей прохода» для всех проходов называется «расписанием ключей» (key schedule). Как правило, оно создается из общего ключа выполнением над ним неких операций, в том числе перестановок и подстановок.

Типичными способами построения алгоритмов симметричного шифрования являются подстановочно-перестановочная сеть[2] и сеть Фейстеля.

Шифр на основе *подстановочно-перестановочной сети* получает на вход блок и ключ и совершает несколько чередующихся раундов, состоящих из чередующихся стадий подстановки (англ. substitution stage) и стадий перестановки (англ. permutation stage).

Для достижения безопасности достаточно одного S-блока, но такой блок будет требовать большого объёма памяти. Поэтому используются маленькие S-блоки, смешанные с P-блоками.

Нелинейная стадия подстановки перемешивает биты ключа с битами открытого текста, создавая конфузию Шеннона. Линейная стадия перестановки распределяет избыточность по всей структуре данных, порождая диффузию.

S-блок (англ. substitution box or S-box) замещает маленький блок входных бит на другой блок выходных бит. Эта замена должна быть взаимно однозначной, чтобы гарантировать обратимость. Назначение S-блока заключается в нелинейном преобразовании, что препятствует проведению линейного криптоанализа. Одним из свойств S-блока является лавинный эффект, то есть изменение одного бита на входе приводит к изменению всех бит на выходе.

P-блок (англ. permutation box or P-box) – перестановка всех бит: блок получает на вход вывод S-блока, меняет местами все биты и подает

[2] Подстановочно-перестановочная сеть или SP-сеть также является разработкой Хорста Фейстеля.

результат S-блоку следующего раунда. Важным качеством P-блока является возможность распределить вывод одного S-блока между входами как можно больших S-блоков.

Для каждого раунда используется свой, получаемый из первоначального, ключ. Подобный ключ называется *раундовым*. Он может быть получен как делением первоначального ключа на равные части, так и каким-либо преобразованием всего ключа.

В сети Фейстеля алгоритм шифрования строит схему шифрования на основе функции F(D, K), где D – порция данных размером вдвое меньше блока шифрования, а K – «ключ прохода» для данного прохода. От функции не требуется обратимость – обратная ей функция может быть неизвестна. Достоинства сети Фейстеля – почти полное совпадение дешифровки с шифрованием (единственное отличие – обратный порядок «ключей прохода» в расписании), что значительно облегчает аппаратную реализацию.

Операция перестановки перемешивает биты сообщения по некоему закону. В аппаратных реализациях она тривиально реализуется как перепутывание проводников. Именно операции перестановки дают возможность достижения «эффекта лавины»[3]. Операция перестановки линейна – f(a) xor f(b) == f(a xor b).

Операции подстановки выполняются как замена значения некоей части сообщения (часто в 4, 6 или 8 бит) на стандартное, жестко встроенное в алгоритм иное число путём обращения к константному массиву. Операция подстановки привносит в алгоритм нелинейность.

Полная утрата всех статистических закономерностей исходного сообщения является важным требованием к симметричному шифру. Как отмечалось ранее, для этого шифр должен иметь «эффект лавины».

---

[3] Лавинный эффект или эффект лавины (англ. Avalanche effect) - понятие в криптографии, обычно применяемое к блочным шифрам и криптографическим хэш-функциям. Важное криптографическое свойство для шифрования, которое означает, что изменение значения малого количества битов во входном тексте или в ключе ведет к «лавинному» изменению значений выходных битов шифротекста. Другими словами, это зависимость всех выходных битов от каждого входного бита.

Термин «лавинный эффект» впервые введён Фейстелем в статье Cryptography and Computer Privacy, опубликованной в журнале Scientific American в мае 1973 года, хотя концептуальное понятие использовалось ещё Шенноном.

Пример лавинного эффекта для алгоритма шифрования Ek(K,M):

- Ek(ключ = «аааа», открытый текст = «аааа») = «5188»;
- Ek(ключ = «аааа», открытый текст = «ааса») = «f7e5».

Также важным требованием является отсутствие линейности (то есть условия f(a) xor f(b) == f(a xor b)).

Зачастую стойкость алгоритма, особенно к дифференциальному криптоанализу, зависит от выбора значений в таблицах подстановки (S–блоках). Как минимум, считается нежелательным наличие неподвижных элементов S(x) = x, а также отсутствие влияния какого–то бита входного байта на какой-то бит результата – то есть случаи, когда бит результата одинаков для всех пар входных слов, отличающихся только в данном бите.

В настоящее время существует множество (как минимум не менее двух десятков) используемых алгоритмов симметричных шифров, существенными параметрами которых являются:

- криптографическая стойкость;
- длина ключа;
- число раундов;
- длина обрабатываемого блока;
- сложность аппаратной/программной реализации;
- сложность преобразования.

Достоинствами симметричных криптосистем являются:

- скорость шифрования/расшифровывания;
- простота реализации (за счёт более простых операций);
- меньшая требуемая длина ключа для сопоставимой стойкости (по сравнению с системами с публичным ключом);
- лучшая изученность (за счёт большего возраста) опять же по сравнению с системами с публичным ключом.

Недостатками симметричных криптосистем являются:

- сложность управления ключами в большой сети;
- сложность обмена ключами. Для применения алгоритмов необходимо решить проблему надёжной передачи ключей каждому абоненту, так как нужен секретный канал для передачи каждого ключа обеим сторонам.

Для компенсации недостатков симметричного шифрования в настоящее время широко применяется комбинированная (гибридная) криптографическая схема, где с помощью асимметричного шифрования передаётся сеансовый ключ, используемый сторонами для обмена данными с помощью симметричного шифрования.

Важным недостатком симметричных шифров является и невозможность их использования в механизмах формирования электронной цифровой подписи и сертификатов, так как ключ известен каждой стороне.

2.2 Подстановки, перестановки и гаммирование

2.2.1 Шифрование с помощью перестановок

Шифр перестановки – это метод симметричного шифрования, в котором символы шифруемого открытого текста меняются местами. В качестве элемента шифруемого текста, как правило, выбирают один символ, однако используются и более крупные конструкции – группы символов.

Классическим примером перестановок являются *анноrаммы*[4].

Исторически перестановки являются одними из первых древнейших методов тайнописи. Неизвестно, когда появились первые перестановки. Возможно, писцы древности использовали анаграммы или перестановки букв в имени своего царя ради того, чтобы скрыть его подлинное имя или в ритуальных целях.

Первые устройства и алгоритмы, работающие на принципах перестановок, известны с V века до нашей эры. К ним можно отнести, например ту же сциталу.

В классической криптографии перестановки разделяются на два крупных класса:

– шифры одинарной (простой) перестановки – при шифровании символы открытого текста перемещаются с исходных позиций в новые один раз;

---

[4] Анаграмма (греч. ανα – «снова» и γράμμα – «запись») – литературный приём, состоящий в перестановке букв или звуков слова (или словосочетания), что в результате даёт другое слово или словосочетание. Например: апельсин – спаниель, полковник – клоповник, горилка – рогалик, лепесток – телескоп.

Прародителем анаграммы считают древнегреческого поэта и грамматика Ликофрона, жившего в III веке до н. э. По сохранившимся записям византийца Иоанна Цеца, из имени царя Птоломея Ликофрон составил первую из известных нам анаграмм: Ptolemaios – Apo Melitos, что в переводе означает «из мёда», а из имени царицы Арсинои: Arsinoe – Ion Eras («фиалка Геры»).

В XVII – XIX вв. среди естествоиспытателей было принято зашифровывать свои открытия в виде анаграмм, что служило двум нуждам: скрыть гипотезу до её окончательной проверки и утвердить авторство на открытие, когда оно будет подтверждено. Например, в 1610 г. Галилео Галилей для закрепления авторства на открытие спутников Сатурна зашифровал латинскую фразу «Altissimun planetam tergeminum observavi» («Высочайшую планету тройную наблюдал») следующим образом: «Smaismrmilmepo etaleumibunenugttauiras» (буквы «v» и «u» в латинских текстах часто считались взаимозаменяемыми). Дальнейшее развитие и популяризация анаграмм была связана с развитием христианства и латыни.

– шифры множественной (сложной) перестановки – при шифровании символы открытого текста перемещаются с исходных позиций в новые несколько раз.

Шифры с простой перестановкой нашли свое применение, как правило, только в ручных методах шифрования. В них часто используются таблицы, которые дают простые шифрующие процедуры перестановки букв в сообщении. Ключом в них служат размер таблицы, фраза, задающая перестановку или специальная особенность таблицы.

Простая перестановка без ключа – один из самых простых методов шифрования, родственный шифру сцитала. Для шифрования исходный открытый текст записывается в таблицу, например, по столбцам. Считывание для получения зашифрованного текста производится по строкам.

Например, сообщение «НЕЯСНОЕ СТАНОВИТСЯ ЕЩЕ БОЛЕЕ НЕПОНЯТНЫМ» записывается в таблицу по столбцам. Для таблицы из 5 строк и 7 столбцов это выглядит так:

Таблица 2.1 – Таблица перестановки

Н	О	Н	С	Б	Н	Я
Е	Е	О	Я	О	Е	Т
Я	С	В	Е	Л	П	Н
С	Т	И	Щ	Е	О	Ы
Н	А	Т	Е	Е	Н	М

После того, как открытый текст записан колонками, для образования шифровки он считывается по строкам. Если его записывать группами по 5 букв, то получится: «НОНСБ НЯЕЕО ЯОЕТЯ СВЕЛП НСТИЩ ЕОЫНА ТЕЕНМ».

Ключом является размер таблицы и алгоритмы записи/чтения. Объединение букв в группы не входит в ключ шифра и используется лишь для удобства записи несмыслового текста.

Более практичный метод шифрования, называемый одиночной перестановкой по ключу очень похож на предыдущий. Он отличается лишь тем, что колонки таблицы переставляются по ключевому слову, фразе или набору чисел длиной в строку таблицы.

Например, используя в качестве ключа слово «ЛУНАТИК», получим такую таблицу:

Таблица 2.2 – Результаты одиночной перестановки

до перестановки	после перестановки
Л У Н А Т И К 4 7 5 1 6 2 3 Н О Н С Б Н Я Е Е О Я О Е Т Я С В Е Л П Н С Т И Щ Е О Ы Н А Т Е Е Н М	А И К Л Н Т У **1 2 3 4 5 6 7** С Н Я Н Н Б О Я Е Т Е О О Е Е П Н Я В Л С Щ О Ы С И Е Т Е Н М Н Т Е А

В верхней строке ее записан ключ, а номера под ключом определенный по естественному порядку соответствующих букв ключа в алфавите. Если в ключе встретились бы одинаковые буквы, они бы нумеровались слева направо.

Получается шифровка: «СНЯНН БОЯЕТ ЕООЕЕ ПНЯВЛ СЩОЫС ИЕТЕН МНТЕА».

Также широкое распространение получил метод маршрутной перестановки. В них ключом является некая геометрическая фигура. Преобразование осуществляется за счет того, что запись текста идет по одной траектории, а чтение – по другой. Опять же, наиболее известным примером можно считать сциталу.

Один из способов маршрутной перестановки носит название «перекресток». В приведенном ниже примере рисуют крестообразные фигуры в количестве, достаточном, чтобы разместить в них все буквы сообщения. Открытый текст записывают вокруг этих фигур заранее оговоренным способом – в нашем случае по часовой стрелке. Буквы берутся построчно. Вначале берется оговоренное количество букв (N) из первой строки, затем удвоенное количество букв (2N) из второй и снова N букв из третьей строки.

Например, сообщение «ПРИЕЗЖАЮ ШЕСТОГО» может выглядеть следующим образом:

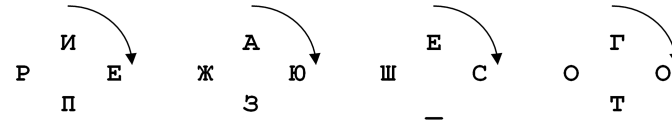


Рисунок 2.2 – Пример размещения открытого текста в шифре «Перекресток»

Например, при N = 2 шифрограмма будет выглядеть «ИА РЕЖЮ ПЗ ЕГ ШСОО _ Т»[5].

Также для маршрутных перестановок могут быть использованы и другие геометрические фигуры, например, треугольники и трапеции. Открытый текст вписывается в эти фигуры в соответствии с количеством слов и формой выбранной фигуры, которая может быть растянута или сжата, чтобы в ней поместилось сообщение.

Для первой фигуры, треугольника, открытый текст записывается пос-трочно от вершины до основания. Ниже записывается ключевое слово.

Если основание треугольника широкое и больше длины ключевого слова, то ключевое слово повторяется. Буквы строки с ключевым словом нумеруются последовательно согласно их алфавитному порядку. Зашифрованное сообщение выписывается по столбцам согласно выполненной нумерации.

Открытый текст				П			
			Р	И	Е		
		З	Ж	А	Ю		
	Ш	Е	С	Т	О	_	О
Ключ	Л	У	Н	А	Т	И	К
Алгоритм перестановки столбцов	4	7	5	1	6	2	3

Рисунок 2.3 – Пример использования шифра перестановки при вписывании в треугольник

[5] Пробелы в зашифрованный текст вставлены для иллюстрации понимания работы алгоритма и при получении зашифрованного текста. В реальном шифровании они, естественно, присутствовать не будут.

Например, для открытого текста «ПРИЕЗЖАЮ _ ШЕСТОГО» и ключевого слова «ЛУНАТИК» шифрограмма полученная на основании треугольника будет выглядеть как «ПИАТ _ ГОШЗЕ-ЕЮОЗЕ».

В 1550 г. итальянский математик Джероламо Кардано предложил новую технику шифрования – решётку Кардано.

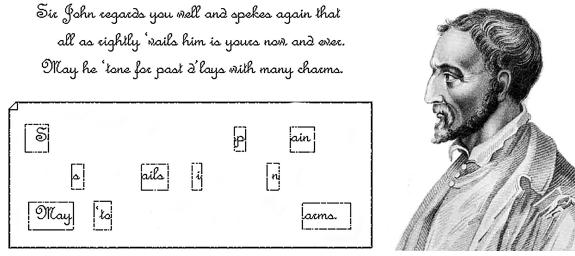


Рисунок 2.4 – Решетка Кардано[6] и ее создатель, Джероламо Кардано (1501-1576)

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

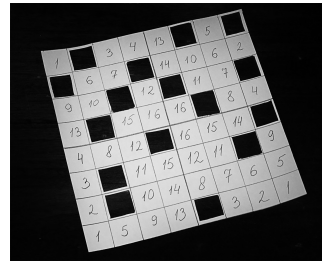


Рисунок 2.5 – Поворотная решетка

[6] Текст записки: «Сэр Джон высоко ценит Вас и снова повторяет, что все, что доступно ему, теперь ваше, навсегда. Может ли он заслужить прощение за свои прежние промедления посредством своего обаяния».

Шифрованное послание: «В мае Испания направит свои корабли на войну».

Позднее был предложен шифр «поворотная решетка» или, как его еще называют, «решетка для вьющихся растений», поскольку она напоминала отверстия в деревянных решетках садовых строений. Этот шифр считают первым транспозиционным (геометрическим) шифром. Естественно, для поворотной решетки удобнее и проще использовать не прямоугольник, а квадрат. Пример квадратной решетки приведен на рисунке 2.5.

Несмотря на то, что между изначальным предложением Кардано и шифром «поворотная решетка» большая разница, методы сокрытия информации, основанные на использовании трафаретов, принято называть «решетками Кардано»[7].

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

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

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

Еще одной разновидностью перестановок являются магические квадраты. Магическими квадратами называются квадратные таблицы со вписанными в их клетки последовательными натуральными числами начиная с 1, которые в сумме по каждому столбцу, каждой строке и каждой диагонали дают одно и то же число.

---

[7] Известно, что кардинал Ришельё был приверженцем решётки Кардано и активно использовал её в личной и деловой переписке.

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

Впервые эти квадраты появились в Китае, где им и была приписана некоторая «магическая сила»[8].

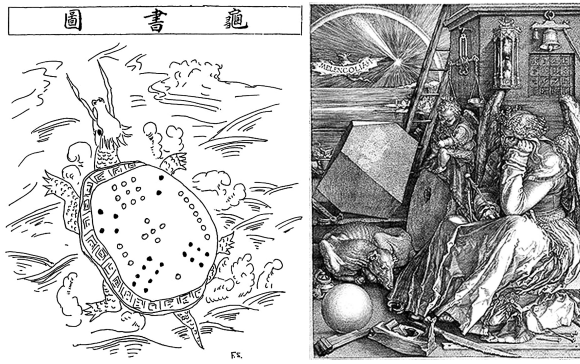


Рисунок 2.6 – Магический квадрат Ло Шу и магический квадрат на гравюре Альберта Дюрера «Меланхолия»

Магические квадраты широко применялись для передачи секретной информации. При шифровании исходное сообщение вписывалось в квадрат по приведенной в них нумерации, после чего шифрограмма выписывалась по строкам. Количество возможных магических квадратов (ключей) быстро возрастает с увеличением их размера. Так, существует лишь один магический квадрат размером 3x3, если не принимать во внимание его повороты. Магических квадратов 4x4 насчитывается уже 880, а число магических квадратов размером 5x5 около 250000. Поэтому магические квадраты больших размеров могли быть хорошей основой для надежной системы шифрования того времени, потому что ручной перебор всех вариантов ключа для этого шифра был практически невыполним.

Для примера, рассмотрим квадрат размером 4x4. В него вписываются числа от 1 до 16. Его магия состоит в том, что сумма чисел по

---

[8] По преданию, описанному в одной из пяти канонических книг Древнего Китая – Шу-Цзин (Книге записанных преданий), в 2200 году до н.э. из реки Ло вышла огромная черепаха (по другой версии – дракон), символ вечности. На ее панцире были видны пятна, образовывавшие удивительный рисунок.

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

строкам, столбцам и полным диагоналям равняется одному и тому же числу – 34.

Шифрование по магическому квадрату производилось следующим образом. Буквы этой фразы вписываются последовательно в квадрат согласно записанным в них числам: позиция буквы в предложении соответствует порядковому числу. В пустые клетки ставится точка или любая буква.

Например, требуется зашифровать фразу: «ПРИЕЗЖАЮ _ ШЕСТОГО» с помощью магического квадрата 4х4.

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

Рисунок 2.7 – Один из возможных 880 вариантов – магический квадрат 4х4

16 О	3 И	2 Р	13 Т
5 З	10 Ш	11 Е	8 Ю
9 _	6 Ж	7 А	12 С
4 Е	15 Г	14 О	1 П

Рисунок 2.8 – Пример шифрования с помощью магического квадрата

Зашифрованный текст записывается в строку (считывание производится слева-направо сверху-вниз, построчно) – «ОЗРТЗШЕЮ _ ЖАСЕГОП».

Шифры сложной перестановки основаны на идее повторного шифрования методами перестановок уже зашифрованного с текста, то есть многократной перестановки.

Наиболее известным методом сложной перестановки является метод двойной перестановки. При этом перестановки определяются отдельно для столбцов и отдельно для строк. В таблицу вписывается открытый текст. После этого переставляют столбцы, а потом строки. При расшифровке порядок перестановок обратный.

Например, произведем шифрование сообщения «ПРИЕЗЖАЮ ШЕСТОГО». Ключ «2413» и «4123».

Таблица 2.3 – Результаты двойной перестановки

исходная таблица					перестановка столбцов					перестановка строк				
	2	4	1	3		1	2	3	4		1	2	3	4
4	П	Р	И	Е	4	И	П	Е	Р	4	А	З	Ю	Ж
1	З	Ж	А	Ю	1	А	З	Ю	Ж	1	Е		С	Ш
2		Ш	Е	С	2	Е		С	Ш	2	Г	Т	О	О
3	Т	О	Г	О	3	Г	Т	О	О	3	И	П	Е	Р

Получается шифровка «АЗЮЖЕ СШГТООИПЕР».

Число вариантов двойной перестановки тоже велико: для таблицы 3х3 их 36, для 4х4 их 576, а для 5х5 их уже 14400. Однако двойная перестановка очень слабый вид шифра, легко читаемый при любом размере таблицы шифрования.

2.2.2 Шифрование с помощью подстановок

Подстановка – это метод симметричного шифрования, основанный на замене символов исходного алфавита на другие символы по определенному правилу. Также как и при подстановках в качестве символа открытого текста может быть использована одиночная буква, пара или группа букв или цифр.

В классической криптографии различают четыре типа шифра подстановки:

- одноалфавитный шифр подстановки (шифр простой замены) – шифр, при котором каждый символ открытого текста заменяется на некоторый, фиксированный при данном ключе символ того же алфавита. Примеры: код Юлия Цезаря, квадрат Полибия, Атбаш;
- однозвучный шифр подстановки похож на одноалфавитный, но в нем символ открытого текста может быть заменен одним из нескольких возможных символов;
- полиграммный шифр подстановки заменяет не один символ, а целую группу. Примеры: шифр Плейфера, шифр Хилла;

– полиалфавитный шифр подстановки состоит из нескольких шифров простой замены. Примеры: шифр Виженера, шифр Бофора, одноразовый блокнот.

При многоалфавитной подстановке закон преобразования меняется от символа к символу.

Одноалфавитный шифр[9] – это класс методов шифрования, которые сводятся к созданию по определённому правилу таблицы шифрования, в которой для каждой буквы открытого текста существует единственная сопоставленная ей буква (символ) зашифрованного текста. Само шифрование заключается в замене букв согласно данной таблице. Для расшифрования используют ту же таблицу.

К шифрам простой замены относятся многие способы шифрования, возникшие в древности или средневековье, как, например, Атбаш, шифр Цезаря, квадрат Полибия, метод пляшущих человечков[10] и многие другие.

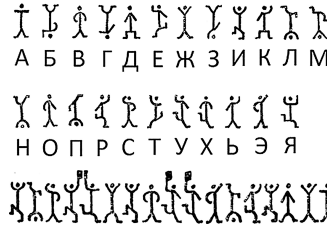


Рисунок 2.9 – Таблица шифрования для одноалфавитного шифра «Пляшущие человечки» и одна из шифрограмм из произведения «Возвращение Шерлока Холмса»

Криптостойкой данного метода шифрования определяется емкостью используемого алфавита.

Система криптографической защиты на основе одноалфавитного шифра легко уязвима. Если противник имеет шифрованный и соответствующий исходный текст или шифрованный текст выбранного противником исходного текста, то определение ключа и дешифрование исходного текста – тривиальная задача.

---

[9] Другие распространенные названия – «шифр простой замены», «простой подстановочный шифр», «моноалфавитный шифр».

[10] «Пляшущие человечки» - один из 56 рассказов английского писателя Артура Конана Дойля о сыщике Шерлоке Холмсе и докторе Ватсоне, включённый писателем в сборник 13 рассказов «Возвращение Шерлока Холмса».

В данном шифре применяются числа, заменяющие буквы. Никакой логики в этих числах нет. Такой простой шифр можно расшифровать, имея таблицу шифров.

В настоящее время шифры простой замены легко взламываются и при наличии только зашифрованного текста. В любом языке различные буквы и комбинации из двух, трех или большего количества букв имеют характерные частоты повторений в текстах.

Например, для русского языка (да, как и любого другого языка) имеются таблицы частот символов.

Таблица 2.4 – Частоты символов в русскоязычном тексте

Буква	Частота	Буква	Частота	Буква	Частота
а	0,075	к	0,034	ф	0,002
б	0,017	л	0,042	х	0,011
в	0,046	м	0,031	ц	0,005
г	0,016	и	0,065	ч	0,015
д	0,030	о	0,110	ш	0,007
е, ё	0,087	II	0,028	щ	0,004
ж	0,009	р	0,048	ь, ъ	0,017
	0,018	с	0,055	ы	0,019
и	0,075	т	0,065	э	0,003
и	0,012	у	0,025	ю	0,022
				я	0,022

Например, частоты символов для художественного произведения, научной статьи либо технической документации, написанных на одном языке будут практически одинаковыми.

Соответственно, если произвести частотный анализ зашифрованного методом простой замены текста, определить частоты символов, то произвести замену шифросимволов на соответствующие буквы языка проблем не представляет.

Следует отметить, что шифр простой замены не всегда подразумевает замену буквы на какую-то другую букву. Допускается использовать замену буквы на число. Например, в квадрате Полибия замена букв происходит на некий шифр-алфавит.

Однозвучные шифры подстановки полностью схожи с одноалфавитными шифрами, за исключением того факта, что в процессе зашифрования символ открытого текста может быть заменен одним из нескольких вариантов, каждый из которых однозначно соответствует исходному. Однозвучный шифр подстановки, в отличие от одноалфавитного шифра, не может быть взломан с помощью частотного криптоанализа, так как он маскирует частотную характеристику текста, хотя и не скрывают всех статистических свойств.

Примерами таких однозвучных шифров являются номенклатор, великий шифр Россиньоля и книжный шифр.

Полиграммные шифры основаны на том, что для повышения криптостойкости шифра замена символов производится не по одному символу, а по нескольким сразу (замена производится граммами)[11]. Примерами таких шифров могут быть биграммные шифры Порты и Плейфера, шифр Хилла и другие подобные.

Шифр Порты является, наверное, первым известным биграммным шифром. Его алгоритм в 1563 году был опубликован в книге Джованни Порты «О тайной переписке»[12].

Порта предложил использовать квадратную таблицу с периодически сдвигаемым смешанным алфавитом и паролем. Он советовал выбирать длинный ключ. Впервые им был предложен шифр простой биграммной замены, в котором пары букв представлялись одним специальным графическим символом. Они заполняли квадратную таблицу размеров 20x20 (в нем не было букв J, K, U, W, X и Z), строки и столбцы которой занумерованы буквами алфавита.

В ячейках таблицы в принципе могли быть записаны любые числа, буквы или символы – сам Джованни Порта пользовался символами – при условии, что содержимое ни одной из ячеек не повторялось.

Несмотря на то, что за этот шифр Порты позднее стали называть отцом современной криптографии, в то время его система не была признана итальянскими криптографами и не нашла широкого приме-

---

[11] Считается, что «отец» биграммных шифров – это немецкий аббат Иоганн Трисемус, который ещё в 1508 г. в своей работе «Полиграфия», впервые отметил возможность шифрования биграммами, то есть, двухбуквенными сочетаниями. Их устойчивость к вскрытию оказалась намного выше, чем у других предшественников, поэтому некоторые биграммные шифры сохранили свою актуальность вплоть до Второй мировой войны.

[12] Порта можно сказать, что предвосхитил то, что называют «методом вероятного слова» и приводит примеры списков вероятных слов из различных областей. По сути, эта книга являлась учебником по криптографии, содержащим криптографические познания того времени.

нения. Причиной этого были сложность шифрования и необходимость постоянно иметь при себе всю таблицу шифра.

	А	Б	В	Г	Д	Е (Ё)	Ж	З	И (Й)	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
А	001	002	003	004	005	006	007	008	009	010	011	012	013	014	015	016	017	018	019	020	021	022	023	024	025	026	027	028	029	030	031
Б	032	033	034	035	036	037	038	039	040	041	042	043	044	045	046	047	048	049	050	051	052	053	054	055	056	057	058	059	060	061	062
В	063	064	065	066	067	068	069	070	071	072	073	074	075	076	077	078	079	080	081	082	083	084	085	086	087	088	089	090	091	092	093
Г	094	095	096	097	098	099	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124
Д	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155
Е (Ё)	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186
Ж	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217
З	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248
И (Й)	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279
К	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310
Л	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341
М	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372
Н	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403
О	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434
П	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465
Р	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496
С	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527
Т	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558
У	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589
Ф	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620
Х	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651
Ц	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682
Ч	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713
Ш	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744
Щ	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775
Ъ	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806
Ы	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837
Ь	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868
Э	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899
Ю	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930
Я	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961

Рисунок 2.10 – Пример таблицы шифрозамен для шифра Порты (русский алфавит)

Одним из наиболее известных полиграммных шифров является шифр Плейфера. Данный ручной алгоритм биграммной подстановки изобретен в 1854 году английским физиком Чарльзом Уитстоном, но назван именем лорда Лайона Плейфера. Его первое описание было

зарегистрировано в документе, подписанном Уитстоном 26 марта 1854 года[13].

Рисунок 2.11 – Чарльз Уитстон (1802-1875) и лорд Лайон Плейфер (1818-1898)

Биграммный шифр Плейфейра разрабатывался для шифрования сообщений парами букв (биграммами).

Основой этого шифра является таблица, ключом служит число строк и столбцов (размер таблицы) и ключевое слово.

Процесс шифрования начинается с этапа подготовки открытого текста, который должен соответствовать следующим требованиям:

- иметь четное число букв, в случае исходного сообщения нечетной длины необходимо добавлять незначимый символ (например, пробел или точку) в конец сообщения;
- после разбиения на пары букв не должно быть биграмм, содержащих две одинаковые буквы. Повторяющиеся подряд две буквы встречаются довольно часто в любом языке, поэтому необходимо сделать так, чтобы они находились в разных биграммах, напри-

[13] Шифр использовался в тактических целях британскими вооруженными силами во Второй Англо-Бурской войне и в Первой мировой войне, а также австралийцами и немцами во время Второй мировой войны. Причиной использования шифра Плейфера была его достаточная простота в применении и отсутствие необходимости в дополнительном специальном оборудовании. Основной целью использования этой системы шифрования была защита важной, но несекретной информации во время ведения боя. К тому времени, когда вражеские криптоаналитики взламывали сообщение, информация уже была бесполезна для них.

Использование шифра Плейфера в настоящее время является нецелесообразным, поскольку современные компьютеры могут легко взломать шифр в течение нескольких секунд. Первый изданный алгоритм взлома шифра Плейфера был описан в 1914 году Джозефом О. Моуборном.

мер, в слове «ДИАГРАММА» при разделении на биграммы четвертая биграмма состоит из двух одинаковых букв «ММ» («ДИ АГ РА ММ А_»). Чтобы исправить данную ситуацию лучше всего добавить пробел в начало слова. Тогда идущие подряд буквы попадут в разные биграммы: «_Д ИА ГР АМ МА».

На заключительном этапе шифрования разделяют открытый текст на пары букв, которые последовательно преобразуются с помощью шифрующей таблицы в биграммы шифртекста по следующим правилам:

- если обе буквы биграммы исходного текста не лежат в одной строке или в одном столбце, тогда находят буквы в углах прямоугольника, определяемого данной парой букв. Первой буквой биграммы шифртекста становится буква, расположенная в той же строке, что и первая буква исходной биграммы, и в том же столбце, что и вторая буква открытого текст, вторая буква биграммы шифртекста находится на пересечении строки, содержащей вторую букву, и столбца, содержащего первую букву открытого текста;
- если обе буквы биграммы открытого текста принадлежат одной строке таблицы, то первой и второй буквами биграммы шифртекста считаются буквы, лежащие справа, соответственно, от первой и второй букв биграммы открытого текста;
- если обе буквы биграммы открытого текста принадлежат одному столбцу таблицы, то первой и второй буквами биграммы шифртекста считаются буквы, лежащие, соответственно, под первой и под второй буквами биграммы открытого текста;
- считается, что таблица циклически замкнута по строкам, то есть конец любой строки связан с ее началом, поэтому если буквы биграммы расположены в одной строке и одна из них находится в последнем столбце таблицы, то для шифртекста берется буква из первого столбца этой строки или если есть конец любого столбца замыкается на его начале. Поэтому если буквы биграммы расположены в одном столбце и одна из них находится в последней строке таблицы, то для шифртекста берется буква из первой строки этого столбца.

Например, зашифруем сообщение «ВО ВРЕМЯ ПЕРВОЙ МИРОВОЙ ВОЙНЫ ИСПОЛЬЗОВАЛИСЬ БИГРАММНЫЕ ШИФРЫ» биграммным шифром Плейфейра.

На этапе подготовки текста учитываем, что в исходном сообщении 61 символ (нечетное число) и одна из биграмм (51 и 52 символы) содержит одинаковые буквы «ММ».

Чтобы увеличить число символов сообщения до четного числа и разделить повторяющиеся буквы по разным биграммам, добавим один пробел перед словом «ИСПОЛЬЗОВАЛИСЬ». Добавление пробела перед словом «БИГРАММНЫЕ» привело бы к ситуации, когда в одной биграмме находятся два пробела.

Разделив текст на биграммы, получим: «ВО», « _ В», «РЕ», «МЯ», « _ П», «ЕР», «ВО», «Й _ », «МИ», «РО», «ВО», «Й _ », «ВО», «ЙН», «Ы _ », « _ И», «СП», «ОЛ», «ЬЗ», «ОВ», «АЛ», «ИС», «Ь _ », «БИ», «ГР», «АМ», «МН», «ЫЕ», « _ Ш», «ИФ», «-РЫ».

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

Рисунок 2.12 – Пример реализации шифра Плейфейра

Для первой биграммы «ВО» используем первое правило шифрования. Она заменяется на биграмму шифртекста «ГМ». Далее, по тому же правилу « _ В» заменяем на «ЭЕ».

Аналогично, применяя алгоритм, производятся замены остальных биграмм.

В результате шифрования исходного сообщения методом Плейфейра с использованием ключа «ЛУНАТИК» получим следующие биграммы шифртекста:

«ГМ», «ЭЕ», «ЩР», «ПЭ», «ЯР», «РЩ», «ГМ», «Т.», «СУ», «СП», «ГМ», «Т.», «ГМ», «КА», «Щ,», «.Т», «ФР», «НЗ», «ЛХ», «МГ», «ТУ», «ЁЪ», «Э.», «ЁЛ», «ЕО», «ПУ», «ОУ», «ЩЖ», «ЯЩ», «КС», «ФЩ».

Шифр Хилла[14] является первым полностью полиграммным шифром подстановки, который позволил на практике (хотя и с трудом)

[14] Впервые шифр Хилла был описан в статье «Cryptography in an Algebraic Alphabet», опубликованной в журнале «The American Mathematical Monthly» в июне-июле 1929 года. В августе того же года Хилл расширил тему и выступил с речью о криптографии перед Американским математическим обществом.

одновременно оперировать более чем с тремя символами (граммами). Шифр основан на линейной алгебре и модульной арифметике. Изобретён американским математиком Лестером Хиллом в 1929 году.

При шифровании букве, во-первых, сопоставляется число. Для латинского алфавита часто используется простейшая схема: A = 0, B = 1, ..., Z = 25. Блок из n букв рассматривается как n-мерный вектор и умножается на n × n матрицу по модулю 26[15]. Матрица целиком является ключом шифра. Матрица должна быть обратима в Z_{26}^{n}, чтобы была возможна операция расшифрования.

Для n = 3 шифр Хилла может быть описан в матричной форме:

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} (\mathrm{mod}\, 26) \qquad (2.1)$$

или

$$C = KP \ (\mathrm{mod}\ 26) \qquad (2.2)$$

где P и C – векторы-столбцы высоты 3, представляющие открытый и зашифрованный текст соответственно;

K – матрица 3 × 3, представляющая ключ шифрования. Операции выполняются по модулю 26.

Для того, чтобы расшифровать сообщение, требуется получить обратную матрицу[16] ключа K^{-1}.

Для того, чтобы расшифровать сообщение, необходимо обратить шифротекст обратно в вектор и затем просто умножить на обратную матрицу ключа:

$$P = K^{-1}C \ (\mathrm{mod}\ 26) = K^{-1}[KP \ (\mathrm{mod}\ 26)] \ (\mathrm{mod}\ 26) = P \bmod 26 \qquad (2.3)$$

при P меньшем 26.

---

[15] Если в качестве основания модуля используется число большее 26, то можно использовать другую числовую схему для сопоставления буквам чисел и добавить пробелы и знаки пунктуации.

[16] Существуют стандартные методы вычисления обратных матриц, однако не все матрицы имеют обратную матрицу. Матрица будет иметь обратную в том и только в том случае, когда её детерминант не равен нулю и не имеет общих делителей с основанием модуля. Если детерминант матрицы равен нулю или имеет общие делители с основанием модуля, то такая матрица не может использоваться в шифре Хилла, и должна быть выбрана другая матрица (в противном случае шифротекст будет невозможно расшифровать).

При работе с двумя символами за раз шифр Хилла не предоставляет никаких конкретных преимуществ перед шифром Плэйфера и даже уступает ему по криптостойкости и простоте вычислений на бумаге. По мере увеличения размерности ключа шифр быстро становится недоступным для расчётов на бумаге человеком. Шифр Хилла размерности 6 (шестиграммный) был реализован механически. Хилл с партнёром получили патент на устройство, которое выполняло умножение матрицы 6×6 по модулю 26 при помощи системы шестерёнок и цепей. Расположение шестерёнок (а значит, и ключ) нельзя было изменять для конкретного устройства, поэтому в целях безопасности рекомендовалось тройное шифрование. Такая комбинация обладала очень высокой криптостойкостью для 1929 года.

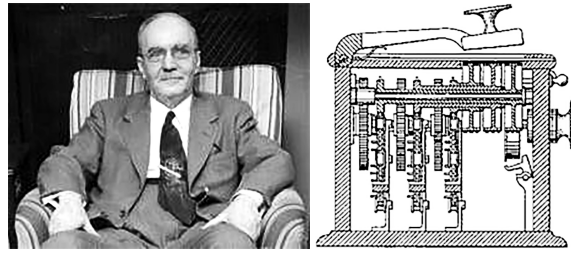


Рисунок 2.13 – Лестер Хилл (1890-1961) и его шифровальная машина

Шифр Хилла, хотя и являлся полиграммным шифром, но не нашёл практического применения в криптографии из-за слабой устойчивости[17] ко взлому и отсутствия описания алгоритмов генерации прямых и обратных матриц большого размера[18].

---

[17] Стандартный шифр Хилла уязвим к атаке по выбранному открытому тексту, потому что в нём используются линейные операции. Криптоаналитик, который перехватит n2 пар символов сообщения/символов шифротекста сможет составить систему линейных уравнений, которую обычно несложно решить. Если окажется, что система не решаема, то необходимо добавить ещё несколько пар символов сообщения/символов шифротекста.

Такого рода расчёты средствами обычных алгоритмов линейной алгебры не требуют значительных затрат времени. В связи с этим для увеличения криптостойкости в шифр Хилла должны быть добавлены какие-либо нелинейные операции. Комбинирование линейных операций, как в шифре Хилла, и нелинейных шагов привело к созданию подстановочно-перестановочной сети (например, сеть Фейстеля). Поэтому современные блочные шифры можно рассматривать как вид полиграммных шифров.

[18] Во Второй мировой войне машины, реализующие шифр Хилла, были использованы только для шифрования трёхсимвольного кода радиосигналов.

С развитием «Черных кабинетов»[19] в странах Европы в XVIII веке все шифры моноалфавитной и полиграммной замен потеряли всякую надежность. Этот факт способствовал вынужденному переходу к использованию полиалфавитных шифров. Другой причиной популяризации более сложного вида шифрования стало развитие телеграфа и возникшая необходимость в защите сообщений от перехвата.

Полиалфавитный шифр подстановки состоит из нескольких шифров простой замены. Примеры: шифр Виженера, шифр Бофора, одноразовый блокнот.

Полиалфавитные подстановочные шифры были изобретены Лионом Баттистой (Lean Battista) в 1568 году. Основная идея многоалфавитных систем состоит в том, что на протяжении всего текста одна и та же буква может быть зашифрована по–разному. То есть замены для буквы выбираются из многих алфавитов в зависимости от положения буквы в тексте. Это является хорошей защитой от простого подсчета частот, так как не существует единой (замены) маскировки для каждой буквы в криптотексте. В данных шифрах используются множественные однобуквенные ключи, каждый из которых используется для шифрования одного символа открытого текста. Первым ключом шифруется первый символ открытого текста, вторым – второй, и т.д. После использования всех ключей они повторяются циклически.

Шифр Вижинера[20] относится к одному из первых полиалфавитных шифров. Отличительной особенностью шифра Вижинера является то, что он прост для понимания и реализации, а также является недоступным для простых методов криптоанализа.

---

[19] Чёрный кабинет – орган, занимающийся перлюстрацией и дешифрованием корреспонденции, и помещение, служащее для этих целей, обычно тайная комната в почтовом отделении. Название берёт начало от соответствующей французской службы Cabinet Noir.

[20] Первое точное документированное описание данного полиалфавитного шифра было сформулировано Леоном Баттиста Альберти в 1467 году. В алгоритме шифрования для переключения между алфавитами использовался металлический шифровальный диск. Система Альберти переключает алфавиты после нескольких зашифрованных слов. Позднее, в 1518 году, Иоганн Трисемус в своей работе «Полиграфия» изобрел tabula recta – центральный компонент шифра Виженера.

То, что принято называть шифром Виженера, впервые описано Джованни Батиста Беллазо, котовый ввел понятие «ключа» для переключения между алфавитами после каждой буквы. Блез Виженер представил своё описание данного шифра перед комиссией Генриха III во Франции в 1586 году.

Рисунок 2.14 – Леон Баттиста Альберти (1404-1472), Иоганн Трисемус (Тритемий) (1462-1516), Джованни Баттиста Беллассо (1505-…), Блез де Вижинер (1523-1596).

Шифр Виженера имел репутацию исключительно стойкого шифра к «ручному» взлому[21].

Шифр Виженера достаточно прост для использования в полевых условиях, особенно если применяются шифровальные диски или шифровальные линейки (линейки Сен-Сира). Например, командование армии Конфедераии во время Гражданской войны в США использовали медный шифровальный диск для шифра Виженера.

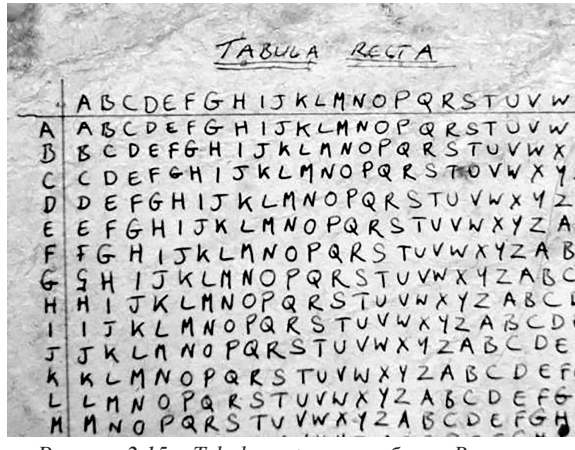


Рисунок 2.15 – Tabula recta или таблица Виженера

[21] Официально принято считать, что шифр Вижинера был признан криптографически нестойким после публикации в XIX веке алгоритма Касиски, хотя известны случаи взлома этого шифра некоторыми криптоаналитиками ещё в XVI веке.

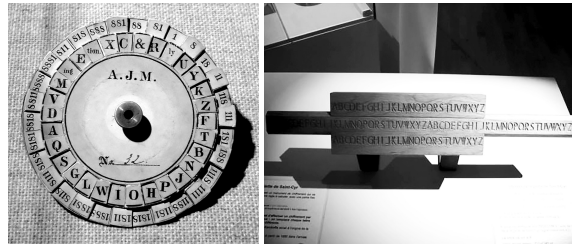


Рисунок 2.16 – Шифровальный диск армии Конфедерации и линейка Сен-Сира[22]

Алгоритм шифрования не содержит сложных преобразований. Во-первых, для зашифровывания создается таблица Виженера. Применительно к латинскому алфавиту таблица Виженера составляется из строк по 26 символов, причём каждая следующая строка сдвигается на несколько позиций[23].

Во-вторых, выбирается ключевая фраза. Если длинна данной фразы (ключа) меньше длинны шифруемого текста, то она повторяется несколько раз: до достижения длинны ключа равному длине шифруемого текста.

В-третьих, для шифрования символа шифруемого текста выбирается строка таблицы Вижинера, соответствующая символу ключа. Таким образом, на каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова.

Например, для открытого текста «ПРИЕЗЖАЮ _ ШЕСТОГО» и ключевого слова «ЛУНАТИК», применим алгоритм Виженера. Для этого, во-первых, создадим таблицу Виженера.

Пример таблицы Виженера представлен на рисунке 2.17.

Циклически записываем ключ «ЛУНАТИК» до тех пор, пока его длина не будет соответствовать длине исходного текста:

Текст – ПРИЕЗЖАЮ _ ШЕСТОГО

Ключ – ЛУНАТИКЛУНАТИКЛУ

[22] В военной академии Сен-Сир придумали простое устройство, состоящее из двух частей – алфавитной линейки и подвижного бегунка с написанным алфавитом и прорезью. В линейке Сен-Сира был реализован шифр замещения с переменным сдвигом, так называемый шифр Блеза де Виженера.

[23] Можно сказать, что в таблице Виженера получается 26 различных шифров Цезаря.

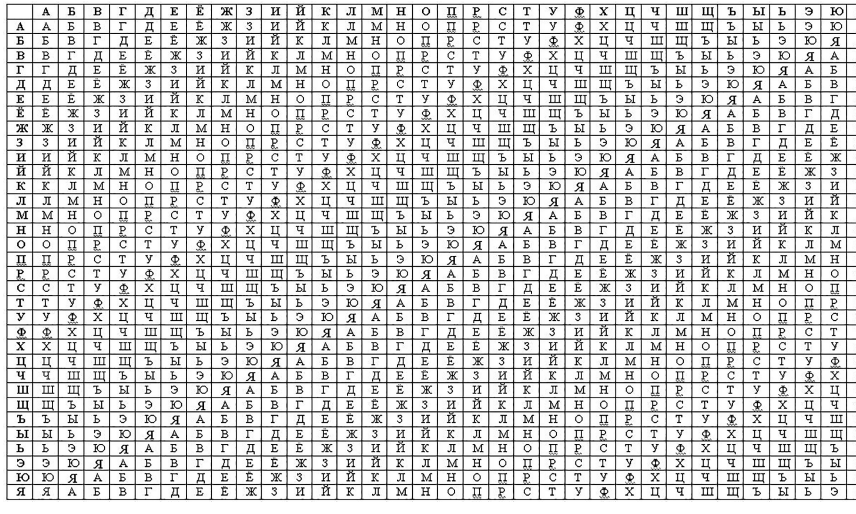


Рисунок 2.17 – Таблица Виженера для кириллического алфавита для примера

Первый символ исходного текста «П» зашифрован последовательностью «Л», которая является первым символом ключа. Первый символ «Ы» зашифрованного текста находится на пересечении строки «Л» и столбца «П» в таблице Виженера. Точно так же для второго символа исходного текста используется второй символ ключа; то есть второй символ шифрованного текста «Д» получается на пересечении строки «У» и столбца «Р». Остальная часть исходного текста шифруется подобным способом.

```
Исходный текст      –   ПРИЕЗЖАЮ _ ШЕСТОГО
Ключ                –   ЛУНАТИКЛУНАТИКЛУ
Зашифрованный текст –   ЫДЦЕЪПКЙ _ ЁЕДЫЩОВ
```

Расшифровывание производится следующим образом: находим в таблице Виженера строку, соответствующую первому символу ключевого слова; в данной строке находим первый символ зашифрованного текста. Столбец, в котором находится данный символ, соответствует первому символу исходного текста. Следующие символы зашифрованного текста расшифровываются подобным образом.

Кажется, что если таблица будет более сложной, чем циклическое смещение строк, то шифр станет надежнее. Это действительно так,

если ее менять чаще, например, от слова к слову. Но составление таких таблиц, где любая буква встречается в строке или столбце один раз, трудоемко и без использования ЭВМ практически не решаемая задача. Для ручного же многоалфавитного шифра полагаются лишь на длину и сложность ключа, используя приведенную таблицу, которую можно не держать в тайне, а это упрощает шифрование и расшифровывание.

Шифр Виженера «размывает» характеристики частот появления символов в тексте, однако некоторые особенности появления символов в тексте остаются. Главный недостаток шифра Виженера состоит в том, что его ключ повторяется и его длину достаточно легко вычислить[24]. После этого «взломать» шифр Виженера не представляет проблему.

В 1863 году Фридрих Касиски (Friedrich Wilhelm Kasiski) нашел способ вскрытия шифра Виженера с коротким кодовым словом, использование которого было наиболее распространено. В случае, когда шифрование производилось с помощью соразмерного с открытым текстом ключа, кодовая фраза может быть подобрана при условии, что она состоит из осмысленных слов. Попытки изобрести новый стойкий к взлому шифр долгое время не приводили к успеху, поэтому криптографы придумывали такие реализации существующих полиалфавитных шифров, чтобы избежать их вскрытия с помощью метода Касиски и метода подбора кодового слова.

Дальнейшей модификацией системы Виженера является система шифров с автоключом. Идея автоключа приписывается математику XVI века Дж. Кардано. Шифрование начинается с помощью «первичного ключа» (который является настоящим ключом в нашем смысле) и продолжается с помощью сообщения или криптограммы, смещенной на длину первичного ключа, затем производится сложение по модулю, равному мощности алфавита.

К примеру:

Сообщение	**ПРИЕЗЖАЮ _ ШЕСТОГО**
Первичный ключ	**ЛУНАТИК**
Автоключ	**ПРИЕЗЖАЮШ**
Шифротекст	**ЫДЦЕЪПКН _ БЙЩЩОБЖ**

[24] Тесты Фридмана и Касиски позволяют определить длину ключа шифра Виженера.

Легальное расшифрование сообщения по известному ключу не представляет труда: по первичному ключу получается начало сообщения, после чего найденная часть исходного сообщения используется в качестве ключа.

Гилберт Вернам (Gilbert Vernam) из AT&T (American Telephone & Telegraph) попытался улучшить криптостойкость взломанного шифра Вижинера (алгоритм в дальнейшем получил название шифр Вернама–Виженера в 1918 году или просто шифр Вернама).

В классическом понимании шифр Вернама является преобразованием открытого текста по большой неповторяющейся последовательностью символов ключа.

Отправитель использовал каждый символ ключа для шифрования только одного символа открытого текста. Шифрование представляет собой сложение по модулю n (мощность алфавита) символа открытого текста и символа ключа из одноразового блокнота[25]. Каждый символ ключа используется только один раз и для единственного сообщения, иначе даже если использовать блокнот большого объема, при получении криптоаналитиком нескольких текстов с перекрывающимися ключами он сможет восстановить исходный текст. Криптоаналитик сдвинет каждую пару шифротекстов относительно друг друга и подсчитает число совпадений в каждой позиции. Если шифротексты смещены правильно, соотношение совпадений резко возрастет. С этой точки зрения криптоанализ не составит труда. Если же ключ не повторяется и случаен, то криптоаналитик, перехватывает тексты или нет, всегда имеет одинаковые знания. Случайная ключевая последовательность, сложенная с неслучайным открытым текстом, дает совершенно случайный криптотекст, и никакие вычислительные мощности не смогут это изменить.

Криптосистема Вернама была предложена для шифрования телеграфных сообщений, которые представляли собой бинарные тексты, в которых открытый текст представляется в коде Бодо (в виде пятизначных «импульсных комбинаций»). В этом коде, например, буква «А» имела вид (11000). На бумажной ленте цифре «1» соответствовало отверстие, а цифре «0» – его отсутствие. Секретный ключ должен был представлять собой хаотичный набор букв того же самого алфавита и реализовывался как одноразовая лента для телетайпов. Для получения

[25] Вернам не использовал понятие «Исключающее ИЛИ» в патенте, но реализовал именно эту операцию в релейной логике. Каждый символ в сообщении преобразовывался побитовым XOR (исключающее ИЛИ) с ключом бумажной ленты.

шифротекста открытый текст объединяется операцией «исключающее ИЛИ» с секретным ключом.

Так, например, при применении ключа (11101) на букву «А» (1 1 0 0 0) получаем зашифрованное сообщение (00101): (11000) ⊕ (11101) = (00101). Зная, что для принимаемого сообщения имеем ключ (11101), легко получить исходное сообщение той же операцией: (00101) ⊕ (11101) = (11000).

Гилберт Вернам создал устройство производящее указанные операции автоматически, без участия шифровальщика и 1919 году получил на него патент. Аппарат Вернама содержал магниты, реле и токосъёмные пластины. С учётом того, что процедуры шифрования и дешифровки математически одинаковы, этот спецаппарат можно было использовать и в последнем случае. Шифрование осуществлялось поступлением импульсов в модуль суммирования из 2-х панелей считывания: одна считывала «гамму», а другая – открытое сообщение. Получаемые на выходном блоке комбинации значений «+» и «–» передавались в линию как обычные телетайпные данные. На приёмном пункте второй такой же спецаппарат производил добавление импульсов, считываемых с перфоленты с идентичной «гаммой», и восстановление первоначальных импульсов исходного сообщения.

Этим аппаратом Гилберт Вернам положил начало так называемому «*линейному шифрованию*», когда процессы шифрования и передачи сообщения происходят одновременно. До той поры шифрование было предварительным, поэтому линейное шифрование существенно повышало оперативность связи.

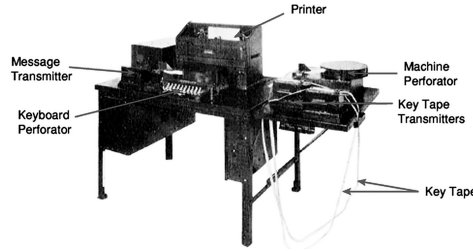


Рисунок 2.17 – Одноленточная электромеханическая шифрмашина создана в США приблизительно в 1933 году Western Union Telegraph Company[26].

[26] В основе таких машин был шифр Вернама.

Также хорошо известен так называемый шифр Вернама по модулю m, в котором знаки открытого текста, шифрованного текста и ключа принимают значения из кольца вычетов Zm. Шифр является обобщением оригинального шифра Вернама, где m = 2.

Например, кодирование шифром Вернама по модулю m=26 (A=0,B=1,…, Z=25):

```
Ключ:              EVTIQWXQVVOPMCXREPYZ
Открытый текст:    ALLSWELLTHATENDSWELL
                   All's well that ends well)
Шифротекст:        EGEAMAIBOCOIQPAJATJK
```

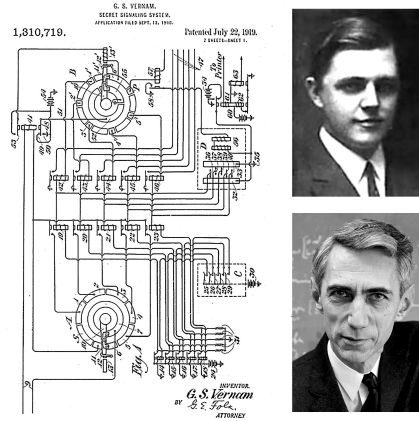


Рисунок 2.18 – Патент на изобретение Вернама, Гилберт Стефорд Вернам (1890-1960) и Клод Элвурд Шенон (1916-2001)

Майор Джозеф Моборн занялся дальнейшим усовершенствованием метода Вермана. Он объединил хаотичность гаммы с правилом одноразового шифроблокнота. Теперь для алгоритма шифрования было введено три ограничения:

- шифроблокнот реализовался как шифрующая гамма, по длине равная или превышающая шифруемое сообщение;
- знаки гаммы были полностью случайными или равновероятными;
- каждая гамма использовалась один и только один раз, после чего уничтожалась передающим или принимающим корреспондентом.

Было еще и дополнительное правило: изготавливались только две копии шифроключа, одна копия для передающего, вторая копия для принимающего корреспондента.

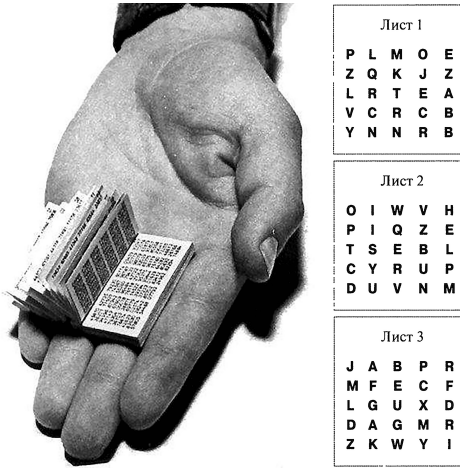


Рисунок 2.19 – Фотография шифроблокнота и три листа его, каждый из которых является возможным ключом для шифра.

При этом следует отметить, что некоторые криптографы полагали, что они могут создать огромное количество случайных ключей, наобум, например, печатая на печатной машинке. Однако при этом машинистка (или оператор печатающего устройства) всякий раз стремился печатать буквы следующим образом: одну букву левой рукой, следующую – правой и так далее, поочередно ударяя по клавишам то на одной, то на другой стороне. Таким способом и в самом деле можно было быстро создать ключ, но получающаяся при этом последовательность обладала структурой и вследствие этого более не являлась случайной – если машинист ударял по клавише с буквой D, находящейся на левой части клавиатуры, то следующей буквой, скорее всего, будет буква, находящаяся на правой части клавиатуры. Если же криптографический ключ одноразового использования действительно случаен, то примерно в половине всех случаев за буквой с левой части клавиатуры должна следовать другая буква с левой же части клавиатуры.

С появлением одноразового блокнота пришло понимание, что лучшие случайные ключи создаются на основе естественных физических процессов, например, радиоактивности. Криптограф может взять крупный кусок радиоактивной руды и измерять излучение с помощью счетчика Гейгера. Иногда ионизирующие частицы излучения испускаются одна за одной очень быстро, иногда между отдельными актами испускания проходит довольно длительное время, поэтому время между этими актами есть величина непредсказуемая и случайная. В таком случае на генераторе случайных чисел в циклическом режиме быстро, но с постоянной скоростью пробегает алфавит, моментально останавливающийся при срабатывании счетчика.

Какой бы ни была буква генератора, она может использоваться в качестве очередной буквы случайного ключа. После этого опять начинается пролистывание алфавита в циклическом режиме до следующего срабатывания счетчика, которое происходит в результате попадания в него ионизирующей частицы.

Такое устройство гарантированно создавало бы действительно случайный ключ, но оно непригодно для повседневной криптографии.

Даже если бы смогли создать достаточно случайные ключи, то возникла бы еще одна проблема: сложность их распределения. Представьте себе район боевых действий, где сотни радистов составляют единую коммуникационную сеть. Для начала все они должны иметь идентичные экземпляры одноразового шифрблокнота. Затем, когда подготовлены новые шифрблокноты, их необходимо одновременно передать всем. Наконец, все должны быть уверены, что нужный лист одноразового шифрблокнота используется в нужное время. Более того, если противник захватит хотя бы один комплект ключей, то надежность всей коммуникационной системы будет разрушена.

Представляется соблазнительным сократить усилия на подготовку и распределение ключей путем повторного использования одноразовых шифрблокнотов, но повторное использование одноразового шифрблокнота позволяет криптоаналитику противника легко дешифровать сообщения.

Практические недостатки теоретически совершенного одноразового шифрблокнота означали, что идею Моборна никогда не удастся широко применять на практике.

В 1945 году Клод Шеннон написал работу «Математическая теория криптографии», в которой доказал абсолютную криптостой-

кость шифра Вернама-Моборна, получившего название «одноразовый блокнот»[27].

С точки зрения криптографии, невозможно придумать систему безопаснее одноразового блокнота или в дальнейшем просто шифра Вернама. Однако и требования к реализации подобной схемы шифрования достаточно нетривиальны, поскольку необходимо обеспечить наложение уникальной гаммы, равной длине сообщения[28], с последующим её гарантированным уничтожением. В связи с этим коммерческое применение шифра Вернама не так распространено в отличие от схем с открытым ключом и он использовался, в основном, для передачи сообщений особой важности государственными структурами.

В настоящее время, как одноразовый блокнот, так и шифр Вернама используются крайне редко. В большой степени это вызвано существенным размером ключа, длина которого должна совпадать с длиной сообщения. Тем не менее, совершенно стойкие шифры типа Вернама всё же нашли практическое применение для защиты особо важных линий связи с относительно небольшим объёмом информации. Так, например, англичане и американцы использовали шифры типа Вернама во время Второй мировой войны. Шифр Вернама по модулю 2 использовался на правительственной «горячей линии» между Вашингтоном и Москвой, где ключевые материалы представляли собой бумажные ленты, на которые знаки ключевой последовательности наносились с помощью перфорации.

На практике можно один раз физически передать носитель информации с длинным истинно случайным ключом, а потом по мере необходимости пересылать сообщения. На этом основана идея шифроблокнотов: шифровальщик по дипломатической почте или при личной

---

[27] Как не удивительно, но класс шифров Вернама – единственный класс шифров, для которого может быть доказана (и была доказана Шенноном) невскрываемость в абсолютном смысле этого термина.

[28] Чтобы обойти проблему предварительной передачи секретного ключа большого объема, инженеры и изобретатели придумали много остроумных схем генерации очень длинных потоков псевдослучайных цифр из нескольких коротких потоков в соответствии с некоторым алгоритмом. Получателя шифрованного сообщения при этом необходимо снабдить точно таким же генератором, как и у отправителя. Но такие алгоритмы добавляющих регулярности в шифротекст, обнаружение которых может помочь аналитику дешифровать сообщение. Другой способ – указание местонахождения ключа как места в книге. Все символы, входящие в алфавит, начиная с указного места книги используются как одноразовый ключ для какого-либо сообщения. Но в данном случае ключ не будет случайным и может быть использована информация о частотах распределения букв.

встрече снабжается блокнотом, каждая страница которого содержит ключи. Такой же блокнот есть и у принимающей стороны. Использованные страницы после однократного использования сразу уничтожаются[29].

С развитием ЭВМ все полиалфавитные шифры перестали быть столь устойчивыми к криптоатакам, и, так же, как в своё время и моноалфавитные шифры, отошли на задний план, став частью истории криптографии.

2.2.3 Гаммирование

Метод гаммирования состоит в том, что на символы шифруемого открытого текста последовательно «накладываются» символы некоторой специальной последовательности, называемой *гаммой* или *гамма-последовательностью.* Суммирование, обычно, выполняется в каком-либо конечном поле.

Например, в поле Галуа GF(2) суммирование принимает вид операции «исключающее ИЛИ (xor)».

При использовании функции «исключающее или» зашифрование производится следующим образом:

$$c_i = m_i \oplus k_i \text{ для } i = 1,2,3... \quad (2.4)$$

где ci – знак шифротекста;

mi – знак открытого текста;

ki – знак ключевой гамма–последовательности;

$\oplus$ – сложение по модулю 2.

Поскольку повторное применение операции XOR восстанавливает первоначальное значение, расшифрование производится повторным наложением гаммы:

$$m_i = c_i \oplus k_i \text{ для } i = 1,2,3... \quad (2.5)$$

---

[29] Бывали случаи, когда одна и та же страница блокнота по различным причинам применялась дважды. Например, среди всего объёма советской шифрованной переписки, перехваченной разведкой США в 40-х годах прошлого века, были обнаружены сообщения, закрытые дважды использованной гаммой. Период этот длился не очень долго, потому что уже после первых успехов американских криптоаналитиков в конце 1940-х годов в спецслужбах СССР узнали о серьёзных проблемах с надёжностью своей шифропереписки. Такие сообщения были расшифрованы в течение 40 последующих лет в рамках секретного проекта «Venona», документы которого были не так давно рассекречены и выложены на сайте АНБ.

Принцип шифрования гаммированием заключается в генерации бесконечного ключа (гаммы шифра) с помощью генераторов псевдослучайных чисел (ПСЧ) и наложении полученной гаммы на исходные данные обратимым образом.

Процесс расшифрования данных сводится к повторной генерации гаммы шифра при известном ключе и наложении такой гаммы на зашифрованные данные.

Если период гаммы превышает длину всего зашифрованного текста и неизвестна никакая часть исходного текста, то мы получаем одноразовый блокнот и зашифрованный текст можно считать, что невозможно раскрыть.

При шифровании методами гаммирования особые требования предъявляют к гамма-последовательности:

- для формирования гаммы (последовательности псевдослучайных чисел) нужно использовать аппаратные генераторы случайных чисел, основанные на физических процессах. Если гамма не будет случайной, для получения открытого текста потребуется подобрать только начальное состояние генератора псевдослучайных чисел;
- длина гаммы должна быть не меньше длины защищаемого сообщения (открытого текста). В противном случае для получения открытого текста потребуется подобрать длину гаммы, проанализировать блоки шифротекста угаданной длины, подобрать биты гаммы.

Однако на практике это требование выполняется только в серьезных государственных структурах и для генерации гамм используют программные методы.

2.3 Генераторы ключей

В 1949 г. Клод Шеннон опубликовал работу, в которой выделил три требования к ключу как гама-последовательности:

- гамма-последовательность должна быть истинно случайной;
- гамма-последовательность должна совпадать по размеру или быть больше заданного открытого текста[30];

[30] Для современных симметричных алгоритмов (AES, CAST5, IDEA, Blowfish, Twofish, ГОСТ28147-89) основной характеристикой криптостойкости является длина ключа. Шифрование с ключами длиной 128 бит и выше считается сильным, так как для расшифровки информации без ключа требуются годы работы мощных суперкомпьютеров.

– гамма-последовательность должна применяться только один раз.

В качестве такой гаммы может быть использована любая последовательность случайных символов. На практике используют длинные случайные или псевдослучайные ключи, сгенерированные с помощью специальных технических устройств или программно-аппаратных комплексов:

1) на основе применения устройств, основанных на физических процессах, например регистрирующее распад ядер, белый шум, естественный радиационный фон, космическое излучение и т.д.

2) на основе применения детерминированных алгоритмов генерации псевдослучайных чисел с помощью функций Random, хеш-функций или рекуррентных формул. При этом следует помнить, что никакой детерминированный алгоритм не может генерировать полностью случайные числа[31]. Любой генератор ПСЧ с ограниченными ресурсами рано или поздно зацикливается – начинает повторять одну и ту же последовательность чисел.

Примером такого «неудачного» алгоритма является печально известный алгоритм RANDU (один из вариантов линейного конгруэнтного генератора псевдослучайных чисел)[32], десятилетиями использовавшийся на мейнфреймах. Он определяется рекуррентным соотношением:

$$V_{i+1} = (65539 \cdot V_i) \bmod 2^{31}, \qquad (2.6)$$

где V_0 нечётное число.

Пример псевдослучайной последовательности, порождаемой алгоритмом RANDU при начальном значении $V_0 = 1$:

```
1
65539
393225
1769499
7077969
26542323
...
388843697
238606867
```

[31] Как сказал Джон фон Нейман, «всякий, кто питает слабость к арифметическим методам получения случайных чисел, грешен вне всяких сомнений».

[32] Согласно принципу Керхгоффса, надёжность криптографической системы должна определяться сокрытием секретных ключей, но не сокрытием используемых алгоритмов или их особенностей.

79531577
477211307
1 (повтор для элемента № 536 870 913).

В общем случае линейный конгруэнтный генератор псевдослучайных чисел задается выражением

$$X_{i+1} = (a\ X_i + b) \bmod m, \tag{2.7}$$

где a, b и m – некоторые коэффициенты.

К сожалению, линейные конгруэнтные генераторы нельзя использовать в криптографии, так как они предсказуемы. Впервые линейные конгруэнтные генераторы были взломаны Джимом Ридсом (Jim Reeds), а затем Джоан Бояр (Joan Boyar). Ей удалось также вскрыть квадратичные генераторы

$$X_{i+1} = (a\ X_i^2 + b\ X_i + c) \bmod m \tag{2.8}$$

и кубические генераторы

$$X_{i+1} = (a\ X_i^3 + b\ X_i^2 + c\ X_i + d) \bmod m. \tag{2.9}$$

Криптографически стойким генератором ПСЧ является алгоритм Блюм – Блюм – Шуба (англ. Algorithm Blum – Blum – Shub, BBS), предложенный в 1986 году Ленор Блюм, Мануэлем Блюмом и Майклом Шубом.

Рисунок 2.20 – Джоан Бояр, Мануэль и Ленор Блюм

Рекуррентная формула BBS выглядит следующим образом:

$$X_{i+1} = X_i^2 \bmod m, \tag{2.10}$$

где $m = p \cdot q$ – является произведением двух больших простых p и q, сравнимых с 3 по модулю 4.

На каждом шаге алгоритма выходные данные получаются из X_i путём взятия либо паритетного бита, либо одного или более младших бит X_i.

Пример с использованием двух малых простых чисел.
p = 7 (7 mod 4 = 3) и q = 19 (19 mod 4 = 3).
m = 7 * 19 = 133.
X_0 = 53.

№ п/п	X		Четный паритетный бит	Младший бит	2 младших бита
	Dec-код	Bin-код			
0	53	110101	0	1	01
1	16	10000	1	0	00
2	123	1111011	0	1	11
3	100	1100100	1	0	00
4	25	11001	1	1	01
...	...	...	...	...	...
Гамма			01011...	10101...	0100110001...

Рисунок 2.21 – Пример генерации гаммы по алгоритму BBS

Особенностью алгоритма BBS является то, что для получения X_n необязательно вычислять все n–1 предыдущих чисел, если известно начальное состояние генератора X_0 и числа p и q, то n–ное значение может быть вычислено «напрямую» по формуле:

$$X_n = X_0^{2^n \bmod((p-1)(q-1))} \bmod m. \qquad (2.11)$$

Также для генерации гамм нашли применение, так называемые, М-последовательности. М-последовательность или последовательность максимальной длины (Maximum length sequence, MLS) – псевдослучайная двоичная последовательность, порожденная регистром сдвига с линейной обратной связью и имеющая максимальный период.

Использование М-последовательностей как псевдослучайных чисел обусловлено, прежде всего, тем, что М-последовательности имеют очень хорошие периодические коррелирующие функции (ПКФ) и генерируются с помощью простой схемы: m – разрядного регистра, охва-

ченного обратной связью через сумматор по модулю 2. Причем длина последовательности, определяемая как

$$N=2^m - 1, \quad (2.12)$$

практически не ограничена: известны М-последовательности длиной до $(2^{34} - 1) = 17\ 179\ 869\ 183$. Из всех двоичных последовательностей М-после-довательности наиболее полно изучены.

М–последовательности называют также последовательностями сдвигового регистра, линейными рекуррентными последовательностями.

Таким образом, можно сделать вывод: для генерации гамма-последовательностей становится понятно, что использование обыкновенных генераторов ПСЧ не обеспечивает необходимой криптостойкости. Необходимо использование специализированных, так называемых, криптографически стойких генераторов ПСЧ. Требуемое «качество» случайности генерируемого криптографически стойкого генератора ПСЧ меняется от задачи к задаче[33].

В идеале, генерация случайных чисел в криптографически стойком генераторе ПСЧ использует высоконадёжный источник энтропии.

Требования к обычному генератору псевдослучайных чисел выполняются и криптографически стойким генераторам ПСЧ, хотя обратное неверно. Требования к криптографически стойким генераторам ПСЧ можно разделить на две группы: во–первых, они должны проходить статистические тесты на случайность; а во–вторых, они должны сохранять непредсказуемость, даже если часть их исходного или текущего состояния становится известна криптоаналитику – скомпроентировано. А именно:

- криптографически стойкий генератор ПСЧ должен удовлетворять «тесту на следующий бит»[34];
- криптографически стойкий генератор ПСЧ должен оставаться надёжным даже в случае, когда часть или все его состояния стали известны (или были корректно вычислены). Это значит, что

[33] Например, генерация одного случайного числа в некоторых протоколах требует только уникальности, тогда как генерация мастер-ключа или одноразового шифроблокнота требует высокой энтропии.

[34] Смысл теста в следующем: не должно существовать полиномиального алгоритма, который, зная первые k битов случайной последовательности, сможет предсказать (k+1)-ый бит с вероятностью более 50%. Эндрю Яо доказал в 1982 году, что генератор, прошедший «тест на следующий бит», пройдёт и любые другие статистические тесты на случайность, выполнимые за полиномиальное время.

не должно быть возможности получить случайную последовательность, созданную генератором, предшествующую получению этого знания криптоаналитиком. Кроме того, если во время работы используется дополнительная энтропия, попытка использовать знание о входных данных должна быть вычислительно невозможна.

В настоящее время получившими распространение следующие криптографически стойкие генераторы ПСЧ[35]:

- алгоритм Ярроу (Yarrow algorithm)[36];
- алгоритм Fortuna, который является наследником алгоритма Ярроу;
- алгоритм Блюма-Микали (Blum-Micali algorithm);
- метод Фибоначчи с запаздываниями;
- Microsoft CryptoAPI;
- Java SecureRandom;
- вихрь Мерсенна;
- SAAC (Indirection, Shift, Accumulate, Add and Count) – разработан в 1996 году Робертом Дж. Дженкинсом младшим, как развитие разработанных им же алгоритмов IA и IBAA[37] и т.д.

Некоторые из перечисленных генераторах ПСЧ для повышения энтропии применяют хеш-функции (SHA-1 и MD5).

2.4 Блочные шифры

Большинство современных программно–реализованных алгоритмов симметричного шифрования делятся на два класса: блочные и потоковые шифры.

Блочный шифр – это разновидность симметричного шифра, оперирующего группами бит фиксированной длины – блоками, характерный размер которых меняется в пределах 32–256 бит. Если исходный текст (или его остаток) меньше размера блока, перед шифрованием его дополняют. Фактически, блочный шифр представляет собой подстановку на алфавите блоков, которая, как следствие, может быть моно-

[35] Ранее рассмотренный алгоритм BBS обладает существенным недостатком – он очень «медленный».

[36] Алгоритм Ярроу используется в FreeBSD, OpenBSD и Mac OS X.

[37] Этот генератор относят к разряду криптостойких генераторов псевдослучайных чисел, хотя полное и строгое доказательство проведено не было.

или полиалфавитной. Блочный шифр является важной компонентой многих криптографических протоколов и широко используется для защиты данных, передаваемых по сети.

В отличие от одноразового блокнота, где длина ключа равна длине сообщения, блочный шифр способен зашифровать одним ключом одно или несколько сообщений суммарной длиной больше, чем длина ключа. Передача малого по сравнению с сообщением ключа по зашифрованному каналу – задача значительно более простая и быстрая, чем передача самого сообщения или ключа такой же длины, что делает возможным его повседневное использование. Однако, при этом шифр перестает быть невзламываемым.

Современная модель блочных шифров основана на идее итеративных блочных шифров, предложенной в публикации 1949 года Клода Шеннона. Данная концепция позволяет достичь определённого уровня безопасности комбинированием простых в исполнении операций подстановки и перестановки.

Первым в области блочных шифров стал шифр «Люцифер»[38], разработанный в 1970 году компанией IBM в рамках исследовательского проекта и основанный на SP-сети. В проекте участвовали ставшие позднее известными криптографами Хорст Фейстель (Horst Feistel) и Дон Копперсмит (Don Coppersmith).

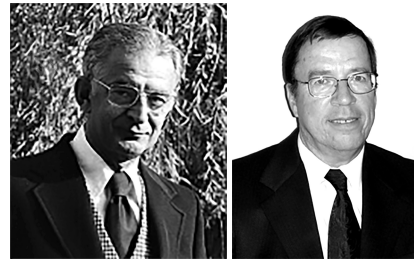


Рисунок 2.22 – Хорст Фейстель (1915-1990) и Дон Копперсмит (1950)

Структура алгоритма Люцифер образца июня 1971 года представляет собой SP-сеть (или подстановочно-перестановочную сеть) – «сэндвич» из слоёв двух типов, используемых по очереди. Первый

[38] Первая версия алгоритма от 1971 года использовала блоки и ключи длиной по 48 бит и основывалась на SP-сетях. Последующая модификация алгоритма была запатентована в ноябре 1971 года (U.S. Patent 3 796 830). В этом шифре использовались 64-разрядные ключи и 32-битные блоки. Последняя, третья, версия предложенная в 1973 году оперировала с 128-битными блоками и ключами.

тип слоя – P-блок разрядности 128 бит, за ним идёт второй слой, представляющий собой 32 модуля, каждый из которых состоит их двух 4-битных S-блоков, чьи соответствующие входы закорочены и на них подаётся одно и то же значение с выхода предыдущего слоя. Но сами блоки подстановок различны (отличаются таблицами замен). На выход модуля подаются значения только с одного из S-блоков, какого конкретно – определяется одним из битов в ключе, номер которого соответствовал номеру S-блока в структуре.

Упрощённая схема алгоритма меньшей разрядности и неполным числом раундов приведена на рисунке 2.24. В ней используется 16 модулей выбора S-блоков (всего 32 S-блока), таким образом, такая схема использует 16-битный ключ.

На рисунке 2.25 показано как меняется шифротекст в приведённом алгоритме при изменении всего одного бита. Для простоты взяты таблицы замен S-блоков такими, что если на вход S-блока подаются все нули, то и на выходе будут все нули. В реальных системах такие таблицы замен не используются, так как они сильно упрощают работу криптоаналитика, но в примере они наглядно иллюстрируют сильную межсимвольную взаимосвязь при изменении одного бита шифруемого сообщения.

На рисунке 2.24 видно, что благодаря первому P-блоку единственная единица сдвигается в центр блока, затем следующий нелинейный S-блок «размножает» её, и уже две единицы за счёт следующего P-блока изменяют своё положение и т.д.

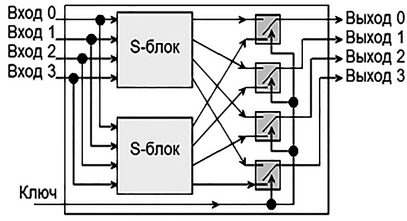


Рисунок 2.23 – Модуль, выбирающий используемую таблицу подстановок по битовому ключу

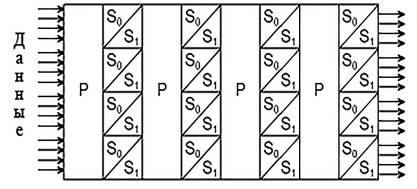


Рисунок 2.24 – Упрощённая схема S- и P-слоёв в алгоритме «Люцифер» (июнь 1971)

В конце устройства шифрования благодаря сильной межсимвольной связи выходные биты стали сложной функцией от входных и от используемого ключа. В среднем, на выходе половина бит будет равна «0» и половина – «1».

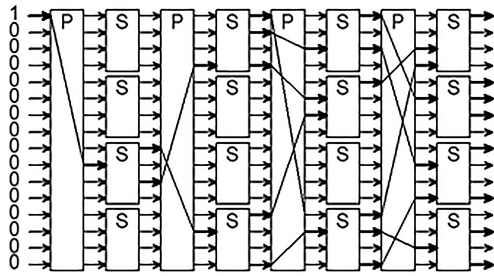


Рисунок 2.25 – Схема генерации и распространения единиц

Идея шифра «Люцифер» заключалась в использовании комбинаций простых, а, следовательно, и быстро вычисляемых как аппаратно, так и программно операций. Однако, алгоритм получился неудачным: был слишком громоздким, что привело к низкой скорости шифрования как и в программной реализации (около 8 кбайт/с), так и в аппаратной (97 кбайт/с).

В начале 70-х годов стали появляться опасения, связанные со стойкостью данного алгоритма. Тем не менее, принципы, выработанные при построении «Люцифера», (SP-сеть и сеть Фейстеля, названная так в честь одного из разработчиков), легли в основу конструирования блочных шифров.

В 1973 году Национальный институт стандартов и технологий (NIST) объявил конкурс с целью разработать стандарт шифрования данных, победителем которого в 1974 году стал шифр DES (Data Encryption Standard), являющийся, фактически, улучшенной версией «Люцифер»[39].

Размер блока для DES равен 64 битам. В основе алгоритма лежит сеть Фейстеля с 16 циклами (раундами) и ключом, имеющим длину 56 бит. 8 бит использовались для формирования имитовставок.

Схема шифрования алгоритма DES указана на рисунке 2.26.

Исходный текст разбит на блоки по 64 бит.

Процесс шифрования состоит из начальной перестановки, 16 циклов шифрования и конечной перестановки.

[39] Публикация шифра DES в 1977 году была основополагающей в общественном понимании современной модели блочного шифра.

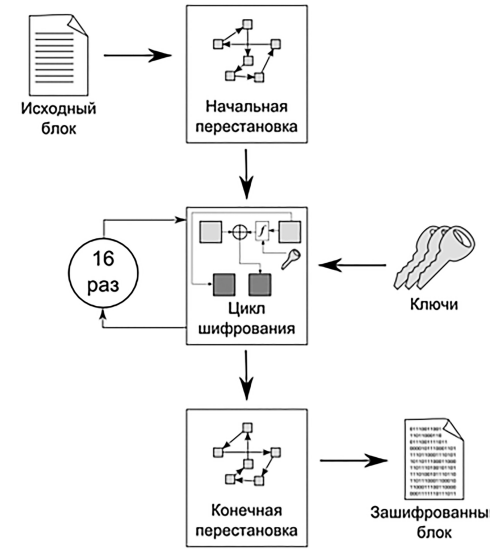


Рисунок 2.26 – Схема шифрования алгоритма DES

В начальной перестановке исходный текст T (блок по 64 бит) преобразуется с помощью перестановки, которая определяется таблицей 2.5.

Таблица 2.5 – Начальная перестановка алгоритма DES ID

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

Согласно таблице 2.5 начальной перестановки первые 3 бита результирующего блока после начальной перестановки являются битами 58, 50, 42 блока исходного текста, а его, например, 3 последние бита являются битами 23, 15, 7 входного блока.

Полученный после начальной перестановки 64-битовый блок текста участвует в 16 циклах преобразования Фейстеля.

Для этого в циклах преобразования происходит изначально разбиение входного блока на две части L_0 и R_0, где L_0 и R_0 – соответственно 32 старших битов и 32 младших битов блока текста или $ID(T_0) = L_0R_0$.

Результат i–ой итерации цикла Ti определяется как:

$$L_i = R_{i-1}, \qquad (2.13)$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, k_i).$$

где – функция Фейстеля, которая в этих циклах играет роль шифрования.

Аргументами функции $f(R_{i-1}, k_i)$ являются 32-битовый вектор R_{i-1} и 48-битовый ключ k_i, который является результатом преобразования 56-битового исходного ключа шифра K.

Для вычисления функции $f(R_{i-1}, k_i)$ последовательно используются:

1. функция расширения E;
2. операция сложения по модулю 2 с ключом $E(R_{i-1}) \oplus k_i$;
3. преобразование S, состоящее из 8 преобразований S-блоков S_1, S_2, S_3, …, S_8:
4. перестановка P.

Функция E расширяет 32-битовый вектор R_{i-1} до 48-битового вектора $E(R_{i-1})$ путём дублирования некоторых битов из R_{i-1}; порядок битов вектора $E(R_{i-1})$ указан в таблице 2.6.

Таблица 2.6 – Функция расширения E

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

По таблице 2.6 видно, что биты 1, 4, 5, 8, 9, 12, 13, 16, 17, 20, 21, 24, 25, 28, 29, 32 дублируются. Полученный после перестановки блок $E(R_{i-1})$ складывается по модулю 2 с ключами k_i и затем представляется в виде восьми последовательных блоков B_1, B_2, B_3, …, B_8.

Таблица 2.7 – Преобразования S_j, i = 1…8

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	S1
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0	
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	S2
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	S3
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	S4
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	S5
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11	S6
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	S7
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	S8
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

$$B_1, B_2, B_3, \ldots, B_8 = E(R_{i-1}) \oplus k_i. \qquad (2.14)$$

Каждый B_j является 6-битовым блоком. Далее каждый из блоков B_j трансформируется в 4-битовый блок B_j' с помощью преобразований S. Преобразования S_j определяются таблицей 2.7.

Предположим, что B_3 = 101111, и мы хотим найти B_3'. Первый и последний разряды B_3 являются двоичной записью числа a, 0<=a<=3, средние 4 разряда представляют число b, 0<=b<=15. Строки таблицы S_3 нумеруются от 0 до 3, столбцы таблицы S_3 нумеруются от 0 до 15. Пара чисел (a, b) определяет число, находящееся в пересечении строки a и столбца b. Двоичное представление этого числа дает B_3'. В нашем случае $a = 11_2 = 3$, $b = 0111_2 = 7$, а число, определяемое парой (3,7), равно 7. Его двоичное представление $B_3' = 0111_2$.

Значение функции $f(R_{i-1}, k_i)$ (32 бит) получается перестановкой P, применяемой к 32-битовому блоку B_1, B_2, B_3, …, B_8. Перестановка P задана таблицей 2.8.

Таблица 2.8 – Перестановка P

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

Преобразование P претставлено формулой

$$f(R_{i-1}, k_i) = P(B_1', B_2', B_3', \ldots, B_8') \qquad (2.15)$$

Согласно таблице 2.8, первые четыре бита результирующего вектора после действия функции f – это биты 16, 7, 20, 21 вектора B_1', B_2', B_3', …, B_8'.

Ключи k_i алгоритма DES получаются из начального ключа K длинной 56 бит следующим образом:

– добавляются биты в позиции 8, 16, 24, 32, 40, 48, 56, 64 ключа K таким образом, чтобы каждый байт содержал нечетное число единиц. Это используется для обнаружения ошибок при обмене и хранении ключей.
– делают перестановку для расширенного ключа (кроме добавляемых битов 8, 16, 24, 32, 40, 48, 56, 64). Такая перестановка определена в таблице 2.9.

Таблица 2.9 – Первая перестановка для генерации ключа

57	49	41	33	25	17	9	1	58	50	42	34	26	18	C_0
10	2	59	51	43	35	27	19	11	3	60	52	44	36	
63	55	47	39	31	23	15	7	62	54	46	38	30	22	D_0
14	6	61	53	45	37	29	21	13	5	28	20	12	4	

Эта перестановка определяется двумя блоками C_0 и D_0 по 32 бит каждый. Первые 3 бита C_0 есть биты 57, 49, 41 расширенного ключа. А первые три бита D_0 есть биты 63, 55, 47 расширенного ключа. C_i,D_i для i = 1,2,3… получаются из C_{i-1},D_{i-1} одним или двумя левыми циклическими сдвигами согласно таблице 6.

Таблица 2.10 – Число сдивга C_{i-1},D_{i-1} для i–ой итерации генерации ключа шифрования

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

Ключ k_i, i = 1,…16 состоит из 48 бит, выбранных из битов вектора C_i,D_i, содержащего 64 бит, согласно таблице 2.11. Первый и второй биты k_i есть биты 14, 17 вектора C_i,D_i.

Таблица 2.11 – Таблица выборки ключа k_i

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

Конечная перестановка при шифровании текста T_0, получившая название ID^{-1}, действует на T_{16} и является обратной к первоначальной перестановке ID. Конечная перестановка определяется таблицей 2.12.

Таблица 2.12 – Обратная перестановка ID^{-1}

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

Подробная схема шифрования алгоритма DES приведена на рисунке 2.27.

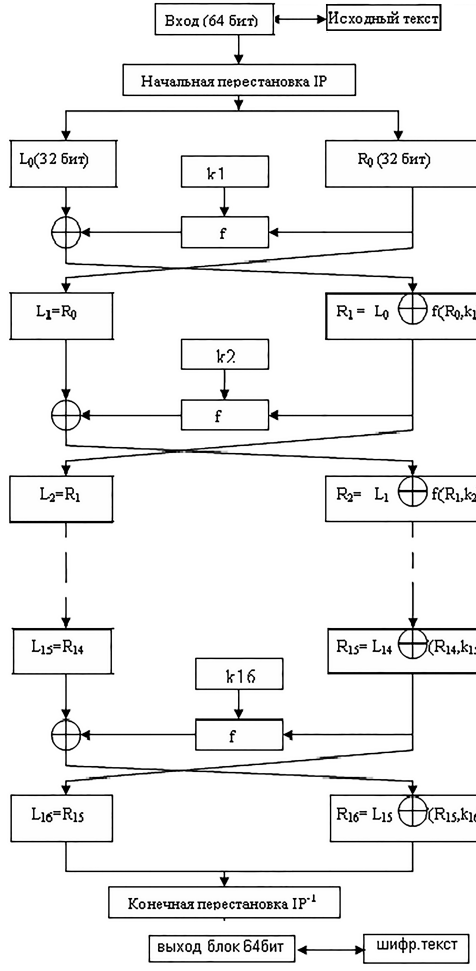


Рисунок 2.27 – Подробная схема шифрования алгоритма DES

При расшифровании данных все действия выполняются в обратном порядке. В 16 циклах расшифрования используется обратное преобразование сетью Фейстеля.

$$R_i = L_{i-1}, \qquad (2.16)$$
$$L_i = R_{i-1} \oplus f(L_{i-1}, k_i).$$

Схема расшифрования указана на рисунке 2.28.

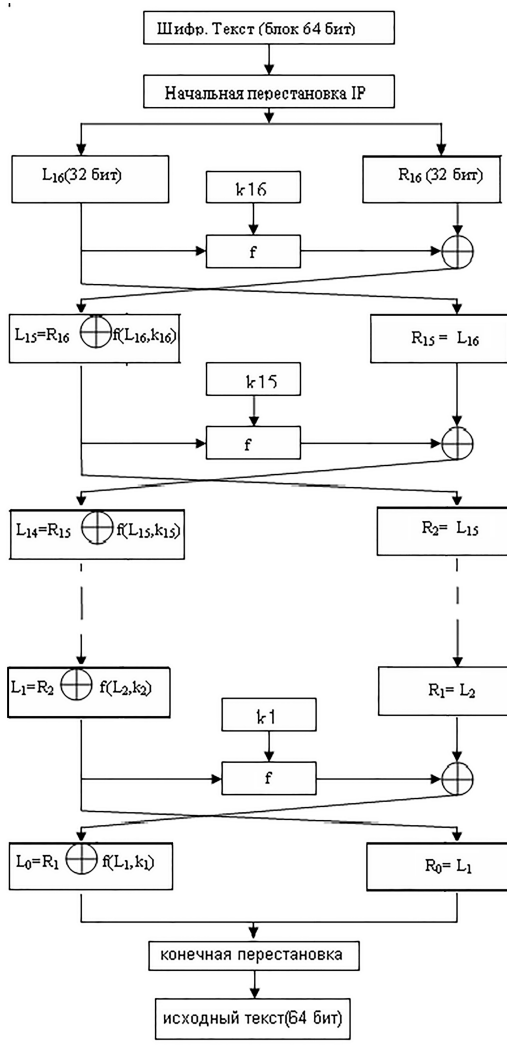


Рисунок 2.28 – Схема расшифрования алгоритма DES

Ключ K_i, i = 16,…,1, функция f, перестановка IP и ID^{-1} такие же, как и в процессе шифрования.

Если в реализации ранее сгенерированные ключи не сохранялись, нужные нам ключи (C_i,D_i, i = 1,2,3…) получаются из C_{i-1},D_{i-1} правыми циклическими сдвигами согласно таблице 2.13.

Таблица 2.13 – Число сдвига C_{i-1},D_{i-1}
для генерации ключей расшифрования

Итерация i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Число сдвига	0	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

После одобрения Американским национальным институтом стандартов в 1981 году, алгоритм долгое время использовался в гражданском секторе и даже вышел за пределы США.

Для DES было рекомендовано несколько режимов использования:

– ECB (**e**lectronic **c**ode **b**ook) – режим «электронной кодовой книги» (для современных блочных шифров получил название «простая замена»). В этом режиме каждый 64 битный блок текста шифруется независимо друг от друга;
– CBC (**c**ipher **b**lock **c**haining) – режим сцепления блоков – один из режимов шифрования для симметричного блочного шифра с использованием механизма обратной связи. Каждый блок открытого текста (кроме первого) побитово складывается по модулю 2 (операция XOR) с предыдущим результатом шифрования;
– CFB (**c**ipher **f**eed **b**ack) – режим обратной связи по шифротексту. В этом режиме для шифрования следующего блока открытого текста он складывается по модулю 2 с перешифрованным (блочным шифром) результатом шифрования предыдущего блока;
– OFB (**o**utput **f**eed **b**ack) – режим обратной связи по выходу. Особенностью данного режима является то, что в качестве входных данных для алгоритма блочного шифрования не используется само сообщение. Вместо этого блочный шифр используется для генерации псевдослучайного потока байтов, который с помощью операции XOR складывается с блоками открытого текста. Подобная схема шифрования называется потоковым шифром;
– Counter Mode (CTR) – режим счётчика. Данный режим предполагает возврат на вход соответствующего алгоритма блочного шифрования значения некоторого счётчика, накопленного с мо-

мента старта. Режим делает из блочного шифра также потоковый.

Несмотря на достаточно широкое распространение алгоритма DES шифр имел существенный недостаток – маленькую длину ключа, породившую множество связанных с параллельным перебором атак. Для DES сложность перебора соответствует 2^{56} или приблизительно 10^{17} операциям, что кластером из нескольких десятков параллельно работающих ЭВМ или одним суперкомпьютером может быть «взломано» за достаточно короткий срок[40].

Также алгоритм DES «страдал» проблемой наличия слабых и частично слабых ключей.

Отсутствие достойной защиты от атак шифра DES породило множество алгоритмов, являющихся как более сложной версией DES (2DES – открытый текст шифровался последовательно по двум различным ключам (ключ 112 бит), 3DES – открытый текст шифровался последовательно по трем различным ключам (168бит), DESX, G–DES), так и совершенно иных схем (NewDES, FEAL, IDEA).

Алгоритм DES был национальным стандартом США в 1977-1980 годы, но в настоящее время DES используется (с ключом длины 56 бит) только для устаревших систем, чаще всего используют его более криптоустойчивый вид (3DES, DESX)[41].

Алгоритм IDEA (Intrnational Data Encryption Algorithm), разработанный и запатентованный швейцарской фирмой Ascom, задумывался как замена стандарта шифрования DES и не стал таковым по причине его запатентованности и необходимости лицензирования для коммерческих приложений. Конечная редакция алгоритма была опубликована в 1992 году.

---

[40] В 1990 году алгоритм DES удалось «взломать» за 39 дней с помощью огромной сети, состоящей из десятков тысяч компьютеров.

Общественная организация «EFF», занимающаяся проблемами информационной безопасности и личной тайны в сети Internet, инициировала исследование «DES Challenge II» с целью выявления проблем DES. В рамках исследования сотрудники фирмы «RSA Laboratory» построили суперкомпьютер стоимостью 250 тыс. долл. В 1998 году суперкомпьютер выполнил расшифровку данных, закодированных методом DES с использованием 56-битного ключа, менее чем за три дня. Суперкомпьютер получил название «EFF DES Cracker».

[41] Алгоритм DES все еще широко применяется для защиты финансовой информации: так, модуль THALES (Racal) HSM RG7000 полностью поддерживает операции TripleDES для эмиссии и обработки кредитных карт VISA, EuroPay и прочих. Канальные шифраторы THALES (Racal) DataDryptor 2000 используют TripleDES для прозрачного шифрования потоков информации. Также алгоритм DES используется во многих других устройствах и решениях THALES-eSECURITY.

Алгоритм IDEA также как и DES оперирует 64-битовыми блоками открытого текста. Алгоритм IDEA обладает рядом преимуществ перед алгоритмом DES. Он значительно безопаснее алгоритма DES, поскольку 128-битовый ключ алгоритма IDEA вдвое больше ключа DES. Внутренняя структура алгоритма IDEA обеспечивает лучшую устойчивость к криптоанализу. Существующие программные реализации примерно вдвое быстрее реализаций алгоритма DES. Недостатком алгоритма является его ориентированность на 16-разрядную архитектуру, что снижает эффективность использования на 32 и 64 битных вычислительных средствах.

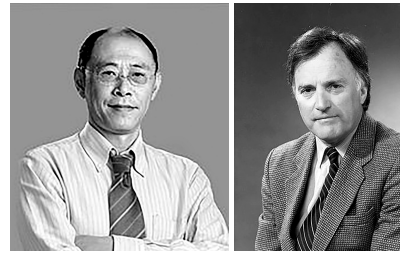


Рисунок 2.29 – Лай Сюэцзя (Xuejia Lai) и Джеймс Мэсси (James Massey), разработчики алгоритма IDEA

В 1997 году стал годом начала программы правительства США по принятию AES (Advanced Encryption Standard – Расширенный Стандарт Шифро-вания)[42], разработанный бельгийцами Винсентом Рэйменом (V. Rijmen) и Йоаном Дайменом (J. Daemen).

[42] 2 января 1997 года NIST (National Institute of Standards and Technology) объявил о намерении выбрать алгоритм шифрования для замены DES и в сентябре 1997 года были представлены официальные требования к алгоритмам. В этих требованиях указывалось, что целью NIST является разработка неклассифицированного, хорошо проанализированного алгоритма шифрования, доступного для широкого применения. Алгоритм должен быть симметричным, блочным, поддерживать длину блока 128 бит и длину ключа 128, 192 и 256 бит. В августе 1998 года NIST анонсировал пятнадцать кандидатов на алгоритм AES на первой конференции по кандидатам AES. Данные алгоритмы были разработаны промышленными и академическими кругами двенадцати стран мира. Вторая конференция по кандидатам AES была проведена в марте 1999 года с целью обсуждения результатов анализа предложенных алгоритмов. В августе 1999 года были представлены выбранные NIST пять финалистов. Ими стали MARS, RC6™, Rijndael, Serpent и Twofish. 2 октября 2000 года было объявлено, что победителем конкурса стал алгоритм Rijndael, и началась процедура его стандартизации. 28 февраля 2001 года был опубликован проект, а 26 ноября 2001 года AES был принят как FIPS 197 (Federal Information Processing Standard).

Алгоритм AES иногда называют еще Rijndael, по названию алгоритма легшего в его основу. Строго говоря, AES и Rijndael – не совсем одно и то же, поскольку AES имеет фиксированный размер блока в 128 бит и размеры ключей в 128, 192 и 256 бит, в то время как для Rijndael могут быть заданы любые размеры блока и ключа, от минимума в 32 бит до максимума в 256 бит с шагом в 32 бита.

Рисунок 2.30 – Йоан Даймен и Винсент Рэймен

Алгоритм представляет каждый блок кодируемых данных в виде двумерного массива байт размером 4x4, 4x6 или 4x8 в зависимости от установленной длины блока. Далее на соответствующих этапах преобразования производятся либо над независимыми столбцами, либо над независимыми строками, либо вообще над отдельными байтами в таблице.

Все преобразования в шифре имеют строгое математическое обоснование. Сама структура и последовательность операций позволяют выполнять данный алгоритм эффективно как на 8-битных так и на 32-битных процессорах. В структуре алгоритма заложена возможность параллельного исполнения некоторых операций, что на многопроцессорных рабочих станциях может еще поднять скорость шифрования в 4 раза.

В таблице 2.14 приведены основные обозначения функций и параметров критопреобразования AES.

Размер ключа в алгоритме AES равен 128 битам, исходя из этого, обычно ключ представляют как матрицу размером 4×4 байта.

В начале процесса шифрования, входные данные открытого текста разбиваются на блоки размером 16 байт или 128 бит. Если полный размер данных не кратен 16 байтам – данные дополняются до размера кратного 16 байтам. Блок данных в алгоритме AES называется state и обычно представляется в виде матрицы 4×4 байта.

Таблица 2.14 – Параметры алгоритма, символы и функции

Обозначение	Смысл обозначения
AddRoundKey()	Преобразование в Шифровании и Дешифровании, в котором Раундовый ключ добавляется к state, используя операцию XOR.
InvMixColumns()	Преобразование в Дешифровании, которое является инверсией MixColumns.
InvShiftRows()	Преобразование в Дешифровании, которое является инверсией ShiftRows.
InvBytesSub ()	Преобразование в Дешифровании, которое является инверсией SubBytes.
K	Ключ шифрования – массив из 128 бит или 16 байт.
MixColumns()	Преобразование в процессе Шифрования, которое берет все столбцы state и смешивает их данные (независимо друг от друга), чтобы получить новые столбцы.
Rcon()	Массив постоянных раундовых Слов.
RotWord()	Функция используемая в операции Расширения ключа, она берет четырехбайтовое слово и выполняет циклическую перестановку.
ShiftRows()	Преобразование в процессе Шифрования, которое выполняется над state, циклически сдвигая последние 3 строки state на различные значения.
BytesSub ()	Преобразование в процессе Шифрования, которое выполняется над state и состоит в замене каждого байта, используя таблицу замены (S-box).
N_k	Длина ключа в словах, для AES 4 слова.
N_b	Длина блока в словах, для AES 4 слова.
N_r	Число раундов при шифровании – 10.

Операция шифрования каждого блока данных проводится независимо от содержимого других блоков. По окончанию шифрования блока – матрица заполняется следующей порцией данных и процесс повторяется. Как уже отмечалось, в силу независимости шифрования одного блока от другого процесс шифрования хорошо поддается распараллеливанию.

S_{00}	S_{01}	S_{02}	S_{03}
S_{10}	S_{11}	S_{12}	S_{13}
S_{20}	S_{21}	S_{22}	S_{23}
S_{30}	S_{31}	S_{32}	S_{33}

Рисунок 2.31 – Представление блока данных AES – state

Каждый блок шифруется в несколько этапов – раундов[43]. Схема криптопреобразования может быть записана следующим описанием:

1. производится расширение ключа KeyExpansion;
2. производится начальная операция – AddRoundKey – суммирования с основным ключом;
3. выполняется 9 раундов из четырех шагов каждый:
   3.1. выполняется процедура BytesSub – замена байтов state по таблице замен;
   3.2. выполняется процедура ShiftRows – циклический сдвиг строк state;
   3.3. выполняется процедура MixColumns – перестановка столбцов state;
   3.4. выполняется процедура AddRoundKey – суммирование с раундовым ключом;
4. выполняется заключительный 10-й раунд:
   4.1. выполняется процедура BytesSub – замена байтов state по таблице замен;
   4.2. выполняется процедура ShiftRows – циклический сдвиг строк state;
   4.3. выполняется процедура AddRoundKey – суммирование с раундовым ключом.

Преобразование BytesSub – это нелинейная замена байт, проводящаяся над каждым байтом state, используя таблицу замены S-box (таблица 2.12).

[43] Количество раундов – от 10 до 14 – зависит от выбранных размера блока и длины ключа

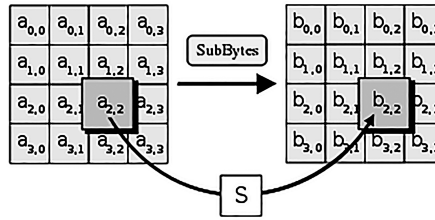


Рисунок 2.32 – ByteSub – табличная подстановка 8x8 бит

Цель применения таблицы замен BytesSub – затруднить линейный и дифференциальный криптоанализ. Таблица замены в алгоритме AES фиксированная. В таблице 2.15 числа представлены в шестнадцатеричной системе счисления, в этой системе счисления любое значение байта представимо не более чем двумя шестнадцатеричными разрядами.

Таблица 2.15 – Таблица замены байт S-box алгоритма AES

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	Fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Замена байта по таблице S-box производится по следующему алгоритму:

– байт Z преобразуется в шестнадцатеричную систему счисления, в формате *XY*h, X – старший разряд, Y – младший разряд. Если старшего разряда нет – он заменяется нулем.
– в S-box выбирается строка X и столбец Y.
– значение Z' на пересечении строки X и столбца Y таблицы S-box используется как замена Z.

Полный процесс SubBytes состоит в замене всех 16 байт матрицы *state*.

В преобразовании ShiftRows байты в последних трех строках state циклически смещаются влево на различное число байт. Строка 1 (нумерация строк с нуля, смещается на один байт, строка 2 – на два байта, строка 3 – на три байта. На рисунке 2.33 проиллюстрировано применение преобразования ShiftRows к state.

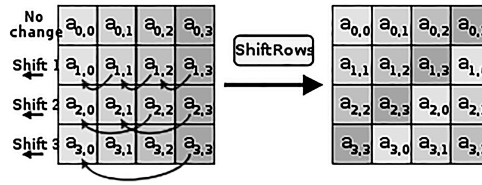


Рисунок 2.33 – ShiftRow – сдвиг строк в двумерном массиве на различные смещения

В преобразовании MixColumns – перемешивание столбца – столбцы состояния (state) рассматриваются как полиномы над полем $F(2^8)$ и умножаются по модулю $x^4 + 1$ на постоянный полином:

$$a(x) = 3x^3 + 1x^2 + 1x^2 + 2 \qquad (2.17)$$

Процесс умножения полиномов эквивалентен матричному умножению

$$\begin{bmatrix} S'_{0c} \\ S'_{1c} \\ S'_{2c} \\ S'_{3c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \bullet \begin{bmatrix} S_{0c} \\ S_{1c} \\ S_{2c} \\ S_{3c} \end{bmatrix},$$

где c – номер столбца массива *state* и $0 \le c \le 3$.

В результате такого умножения, байты столбца с $\{S_{0c}, S_{1c}, S_{2c}, S_{3c}\}$ заменяются, соответственно, на байты

$$\begin{aligned} S'_{0c} &= (2 \bullet S_{0c}) \oplus (3 \bullet S_{1c}) \oplus S_{2c} \oplus S_{3c}; \\ S'_{1c} &= S_{0c} \oplus (2 \bullet S_{1c}) \oplus (3 \bullet S_{2c}) \oplus S_{3c}; \\ S'_{2c} &= S_{0c} \oplus S_{1c} \oplus (2 \bullet S_{2c}) \oplus (3 \bullet S_{3c}); \\ S'_{3c} &= (3 \bullet S_{0c}) \oplus S_{1c} \oplus S_{2c} \oplus (2 \bullet S_{3c}). \end{aligned} \qquad (2.18)$$

Преобразование (2.18) применяется к каждому из четырех столбцов state.

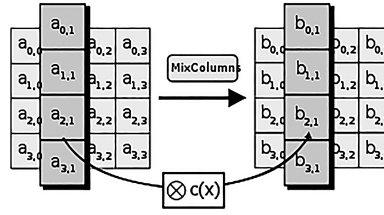


Рисунок 2.34 – MixColumn – математическое преобразование, перемешивающее данные внутри столбца

В преобразовании AddRoundKey, раундовый ключ RK добавляется к state посредством поразрядного XOR. Каждый раундовый ключ состоит из 16 байт расширенного ключа. Байты раундового ключа записываются в матрицу 4×4, подобную state. Каждый байт раундового ключа суммируется с соответствующим байтом из *state*, как показано на рисунке 2.35.

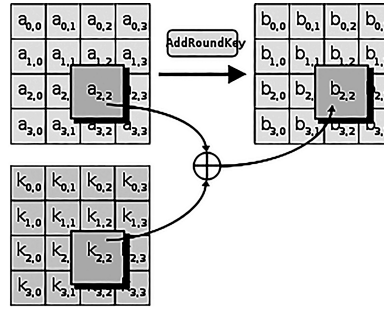


Рисунок 2.35 – AddRoundKey – добавление материала ключа операцией XOR

В последнем раунде операция перемешивания столбцов отсутствует, что делает всю последовательность операций симметричной.

Для работы алгоритма выполняется расширение заданного ключа и по этому расширению производит создание раундовых ключей.

Расширенный ключ W содержит 4×(10+1) слов – начальный ключ в 4 слова и по 4 слова расширенного ключа на каждый из 10 раундов. Расширенный ключ W состоит из слов (четыре байта на слово), обозначаемых ниже как w_i, где i находится в диапазоне [0..44]. Полная длина расширенного ключа 1408 бит, по 128 бит на каждый раунд.

В процессе расширения ключа используется массив констант Rcon. Элементы массива Rcon пронумерованы от 1 до 256+3. Значения элементов массива определены следующим набором функций:

$Rcon_1 = 1;$
$Rcon_k = 2 \bullet Rcon_{k-1} = 2^{k-1}$, для k = 2, 3,...255;
$Rcon_k = 0$, для k = 256, 257, 258;

Расширение ключа можно описать следующей последовательностью операций:

– четыре слова ключа шифрования K копируются в первые четыре слова расширенного ключа W: $w_i = k_i$ для i = 0, 1, 2, 3;
– остальные слова расширенного ключа W для i = 4, 5, ...44 генерируются так: если i кратно 4, то $w_i = SubBytes(RotByte(w_{i-1})) \oplus Rcon_{(i/4)}$; иначе если i не кратно 4, то $w_i = w_{i-4} \oplus w_{i-1}$.

Функция RotByte переставляет четыре байта исходного слова $\{a_0, a_1, a_2, a_3\}$ с помощью циклической перестановки, превращая в слово $\{a_3, a_1, a_2, a_0\}$. Функция SubBytes применяет к каждому из четырех байтов слова замену по таблице S-box.

Раундовый ключ RK для раунда k выбирается из расширенного ключа W как слова с w_{4k} по $w_{4(k+1)}$.

Все преобразования шифрования однозначны и, следовательно, имеют обратное преобразование, т.е. могут быть инвертированы и выполнены в обратном порядке, чтобы выполнить дешифрование для алгоритма AES.

Схема криптопреобразования для расшифрования может быть быть представлена следующим алгоритмом:

1. производится расширение ключа KeyExpansion;
2. выполняется 9 раундов из четырех шагов каждый;
   2.1. выполняется процедура AddRoundKey – суммирование с раундовым ключом;
   2.2. выполняется процедура InvMixColumns – обратная перестановка столбцов state;

2.3. выполняется процедура InvShiftRows – обратный циклический сдвиг строк state;

2.4. выполняется процедура InvSubBytes – обратная замена байтов *state* по таблице замен;

3. выполняется заключительный 10-й раунд;

3.1. выполняется процедура AddRoundKey – суммирование с раундовым ключом;

3.2. выполняется процедура InvShiftRows – обратный циклический сдвиг строк state;

3.3. выполняется процедура InvSubBytes – обратная замена байтов state по таблице замен.

Преобразование InvMixColumns является обратным для преобразования MixColumns. В преобразовании InvMixColumns, столбцы состояния (state) рассматриваются как полиномы над полем $F(2^8)$ и умножаются по модулю $x^4 + 1$ с постоянным полиномом $d(x) = a^{-1}(x)$, в поле $F(2^8)$:

$$d(x) = 0Bhx^3 + 0Dhx^2 + 09hx + 0Eh. \quad (2.19)$$

Преобразование InvShiftRows обратно преобразованию ShiftRows. Байты последних трех рядов массива state циклически сдвигаются вправо. Строка 1 (нумерация с нуля) смещается на 1 байт, строка 2 – на 2 байта, строка 3 – на 3 байта.

Преобразование InvSubBytes выполняет обратную замену байт с помощью обратной таблицы замен InvS-box.

Таблица 2.16 – Таблица InvS–box для обратной замены байт

	0	1	2	3	4	5	6	7	8	9	A	b	c	d	e	f
00	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
10	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
20	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
30	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
40	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
50	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
60	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
70	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
80	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
90	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a0	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b

b0	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c0	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d0	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e0	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f0	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

В июне 2003 года Агентство национальной безопасности США постановило, что шифр AES является достаточно надёжным, чтобы использовать его для защиты сведений, составляющих государственную тайну. Вплоть до уровня SECRET было разрешено использовать ключи длиной 128 бит, для уровня TOP SECRET требовались ключи длиной 192 и 256 бит. Однако, правительство США постановило, что AES должен периодически подвергаться проверкам и улучшениям, чтобы надежно хранить зашифрованные данные[44].

Криптостойкость современных алгоритмов симметричного шифрования для атак в лоб (методов сплошного перебора) в большей степени определяется длинной ключа. Так в таблице 2.17 показаны характеристики криптостойкости в количестве операций и оцениваемом времени для различных алгоритмов.

Таблица 2.17 – Криптостойкость алгоритмов шифрования[45]

Длина ключа (в битах)	Количество комбинаций	Ориенировочное время взлома
16	$2^{16} = 65536$	
56 (DES)	$2^{56} = 7{,}2 \cdot 10^{16}$	399 секунд
128 (AES-128)	$2^{128} = 3{,}4 \cdot 10^{38}$	$1{,}02 \cdot 10^{18}$ лет
256 (AES-256)	$2^{256} = 6{,}2 \cdot 10^{57}$	$1{,}872 \cdot 10^{37}$ лет
512 (AES-512)	$2^{512} = 1{,}1 \cdot 10^{77}$	$3{,}31 \cdot 10^{56}$ лет

[44] Поддержка AES (и только его) введена фирмой Intel в семейство процессоров x86 начиная с Intel Core i7-980X Extreme Edition, а затем на процессорах Sandy Bridge.

[45] Озвученные результаты исследования Алекса Бирюкова (Alex Biryukov), Орра Данклмана (Orr Dunkelman), Натана Келлера (Nathan Keller), Дмитрия Ховратовича (Dmitry Khovratovich) и Ади Шамира (Adi Shamir) на конференции CRYPTO 2011 результаты криптоанализа алгоритма AES указывают на способ атаки, позволяющий в четыре раза сократить трудоёмкость выполнения операций по подбору секретного ключа. Иными словами на деле криптостойкость AES-128 сводится к AES-126, а AES-192 к AES-189. Однако даже при этих решениях время «взлома» превышает миллиарды лет.

На базе алгоритма Rijndael, лежащего в основе AES, реализованы и альтернативные криптоалгоритмы. Среди наиболее известных алгоритмов – участники конкурса Nessie: Anubis на инволюциях, автором которого является Винсент Рэймен и усиленный вариант шифра – Grand Cru Йохана Борста.

При рассмотрении симметричных блочных шифров нельзя не остановиться еще на одном всемирно известном алгоритме – ГОСТ 28147-89 – советском и российском стандарте.

ГОСТ[46] введенный в 1990 году, также является и стандартом стран СНГ. Полное название – «ГОСТ 28147-89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования». Алгоритм при использовании метода шифрования с гаммированием, может выпол-нять функции поточного шифроалгоритма[47].

Алгоритм ГОСТ 28147-89 использует 256-битный ключ шифрования и 32 цикла преобразования 64-битных блоков исходного открытого текста. Алгоритм реализует классическую схему сети Фейштеля. Упрощенный алгоритм реализации образующей функции представлен на рисунке 2.36.

Изначально правая половина блока и 32 битный ключ раунда складываются по модулю 2. Результат сложения разбивается на восемь 4-битовых последовательностей, каждая и которых поступает на вход соответствующего S-блока.

Каждый блок представляет собой таблицу подстановки, которая заменяет поступающее на вход число в диапазоне [0..15] на другое число в том же диапазоне. Выходы всех S-блоков объединяются

---

[46] По некоторым сведениям, история этого шифра гораздо более давняя. Алгоритм, положенный впоследствии в основу стандарта, родился, предположительно, в недрах Восьмого Главного управления КГБ СССР (ныне в структуре ФСБ Российской Федерации), скорее всего, в одном из подведомственных ему закрытых НИИ, вероятно, ещё в 1970-х годах в рамках проектов создания программных и аппаратных реализаций шифра для различных компьютерных платформ.

С момента опубликования ГОСТа на нём стоял ограничительный гриф «Для служебного пользования», и формально шифр был объявлен «полностью открытым» только в мае 1994 года. К сожалению, история создания шифра и критерии разработчиков до сих пор не опубликованы.

[47] По аналогии с AES (и в отличие от DES), ГОСТ допущен к защите секретной информации без ограничений, в соответствии с тем, как это указано в российском стандарте. Таким образом, ГОСТ – это не аналог DES, а конкурент 3DES с тремя независимыми ключами или AES-256.

в 32-битное слово, которое затем циклически сдвигается влево на 11 битов и объединяется с левой частью блока операцией XOR.

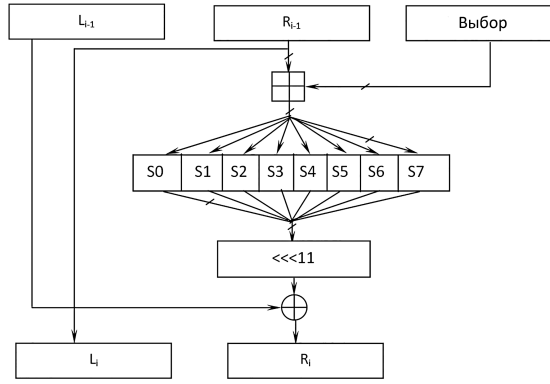


Рисунок 2.36 – Образующая функция алгоритма ГОСТ 28147-89

Формирование ключей раунда осуществляется по следующей схеме: 256-битный ключ разбивается на восемь 32-битных машинных слов. Они нумеруются с K_0 по K_7.

32 ключа раунда получаются применением этих машинных слов в следующем порядке:

$$K_0, K_1, K_2,..,. K_7, K_0, K_1, K_2,...,K_7, K_0, K_1, K_2,\ldots, K_7, K_7, K_6, K_5,..., K_0, \quad (2.20)$$

то есть в последних 8 раундах ключи подаются в обратном порядке.

Алгоритм ГОСТ является симметричным, и для расшифрования достаточно подать на вход алгоритма блоки зашифрованных сообщений и ключи раундов в порядке, обратном их следованию при шифрации.

Особенностью и, наверное, главным недостатком, алгоритма является отсутствие в стандарте каких-либо рекомендаций по выбору содержимого таблиц подстановок (S-блоков).

Первоначально 8•16•4 = 512 бит таблиц подстановок являлись также частью ключевой информации. Впоследствии требование к секретности содержимого таблиц было упразднено, однако статическое, обеспечивающее высокую криптостойкость алгоритма, содержимое таблиц подстановки так и не было опубликовано.

Набор S-блоков, рекомендуемый стандартом хеширования ГОСТ Р34.11-94, использующего блочный шифр ГОСТ 28147-89 в качестве основной преобразующей операции, приведен в таблице 2.18.

Таблица 2.18 – S-блоки алгоритма ГОСТ 28147-89

S0	4	10	9	2	13	8	0	14	6	11	1	12	7	15	5	3
S1	14	11	4	12	6	13	15	10	2	3	8	1	0	7	5	9
S2	5	8	1	13	10	3	4	2	14	15	12	7	6	0	9	11
S3	7	13	10	1	0	8	9	15	14	4	6	12	11	2	5	3
S4	6	12	7	1	5	15	13	8	4	10	9	14	0	3	11	2
S5	4	11	10	0	7	2	1	13	3	6	8	5	9	12	15	14
S6	13	11	4	1	3	15	5	9	0	10	14	7	6	8	2	12
S7	1	15	13	0	5	7	10	4	9	2	3	14	6	11	8	12

Достоинствами алгоритма ГОСТ можно назвать простоту реализации, причем как программной, так и аппаратной, поскольку алгоритм не использует битовых перестановок, большой размер ключа делает малоперспективной силовую атаку на шифр, большое количество раундов затрудняют дифференциальный и линейный криптоанализ. Алгоритм оптимизирован под 32-разрядные процессоры.

Единственной проблемой практического применения алгоритма является, как уже упоминалось, формирование таблиц подстановки[48].

Анализ криптоалгоритма ГОСТ позволяет говорить о его практической неуязвимости еще порядка 200 лет (естественно, при развитии вычислительной техники по Закону Мура).

От поточных шифров работа блочного отличается обработкой бит группами, а не потоком. При этом блочные шифры надёжней, но медленнее поточных.

[48] В дополнение к очень большому размеру ключа, ГОСТ имеет значительно более низкую стоимость исполнения по сравнению с AES. В действительности, он стоит намного меньше AES, которому требуется в четыре раза больше аппаратных логических вентилей ради значительно меньшего заявленного уровня безопасности.

Неудивительно, что ГОСТ стал интернет-стандартом, в частности, он включён во многие криптобиблиотеки, такие как OpenSSL и Crypto++. В 2010 году ГОСТ был заявлен на стандартизацию ISO как всемирный стандарт шифрования.

Крайне малое количество алгоритмов смогли стать международными стандартами. Международный стандарт ISO/IEC 18033-3:2010 описывает следующие алгоритмы: четыре 64-битных шифра – TDEA, MISTY1, CAST-128, HIGHT – и три 128-битных шифра – AES, Camellia, SEED.

К достоинствам блочных шифров относят сходство процедур шифрования и расшифрования, которые, как правило, отличаются лишь порядком действий. Это упрощает создание устройств шифрования, так как позволяет использовать одни и те же блоки в цепях шифрования и расшифрования. Гибкость блочных шифров позволяет использовать их для построения других криптографических примитивов: генератора псевдослучайной последовательности, поточного шифра, имитовставки и криптографических хэшей.

2.5 Потоковые шифры

Поточный или потоковый шифр – это симметричный шифр, в котором каждый символ открытого текста преобразуется в символ шифрованного текста в зависимости не только от используемого ключа, но и от его расположения в потоке открытого текста.

Потоковые шифры на базе сдвиговых регистров активно использовались в годы Второй мировой войны, ещё задолго до появления электроники. Они были просты в проектировании и реализации.

В 1965 году Эрнст Селмер (Ernst Sejersted Selmer), главный криптограф норвежского правительства, разработал теорию последовательности сдвиговых регистров. Позже Соломон Голомб (Solomon Wolf Golomb), математик Агентства Национальной Безопасности США[49], написал книгу под названием «Shift Register Sequences» («Последовательности сдвиговых регистров»), в которой изложил свои основные достижения в этой области, а также достижения Селмера.

Так как практически во всех каналах передачи данных для потоковых систем шифрования присутствуют помехи, то для предотвращения потери информации криптоаналитики вынуждены решать проблему синхронизации шифрования и расшифрования текста. Поэтому по способу решения этой проблемы шифросистемы подразделяются на синхронные и системы с самосинхронизацией.

Схема шифрования с использованием синхронных потоковых шифров представлена на рисунке 2.37.

[49] Соломон Вольф Коломб – выдающийся математик, внесший значимый вклад в развитие информационных технологий XX века. Также известен как изобретатель полимино (обобщённого домино), вдохновившего российского программиста Алексея Пажитнова на создание компьютерной игры «Тетрис».

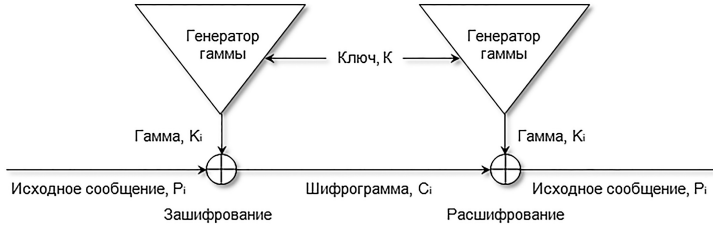


Рисунок 2.37 – Схема шифрования с использованием синхронного потокового шифра

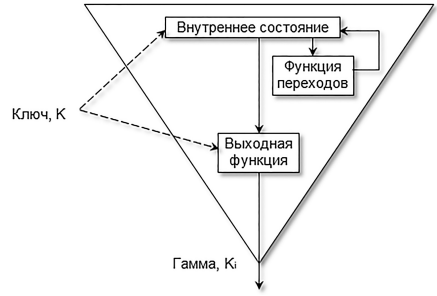


Рисунок 2.38 – Устройство генератора гаммы с внутренней обратной связью

При этом генератор гаммы, как правило, состоит из трех основных блоков (рисунок 2.39).

Внутреннее состояние описывает текущее состояние генератора гаммы. Начальное внутреннее состояние, как правило, определяется ключом K. Два генератора, с одинаковым ключом и одинаковым внутренним состоянием, создают одинаковые гаммы. Функция переходов считывает текущее внутреннее состояние и генерирует новое внутреннее состояние. Выходная функция считывает внутреннее состояние и генерирует бит (биты) гаммы K_i.

В другой разновидности, так называемых генераторах типа счетчик, отсутствует блок с функцией переходов. В отличие от генераторов с обратной связью, они позволяют вычислить i-й бит гаммы, не вычисляя всех предыдущих битов. Для этого генератор устанавливается в i-е внутреннее состояние, после чего вычисляется соответствующий ему i-й бит гаммы. Это свойство полезно использовать для обеспечения произвольного доступа к файлам данных, что позволяет

расшифровать отдельный фрагмент данных, не расшифровывая файл полностью.

В синхронном потоковом шифре гамма генерируется независимо от потока сообщения. На шифрующей стороне генератор гаммы последовательно выдает биты гаммы K_i. На расшифровывающей стороне другой генератор гаммы один за другим выдает идентичные биты гаммы. Эта схема работает нормально, если оба генератора синхронизированы.

Основным недостатком синхронных потоковых шифров является то, что если один из генераторов пропускает один из циклов или бит шифрограммы теряется при передаче, то все символы шифрограммы, следующие за ошибкой, расшифровываются некорректно. В этом случае отправитель и получатель должны синхронизировать генераторы и заново передать некорректно расшифрованную часть сообщения.

В потоковых шифрах если происходит искажение бита шифротекста или ключа, то это приводит только к потере одного бита расшифрованного текста. Более проблематичной становится ситуация если бит не искажается, а теряется – в этом случае искажается вся информация начиная с данного бита.

В самосинхронизирующемся потоковом шифре каждый бит гаммы представляет собой функцию фиксированного числа предыдущих битов шифрограммы. Используемые при таком шифровании генераторы гаммы называют генераторами с обратной связью по шифрограмме (шифртексту).

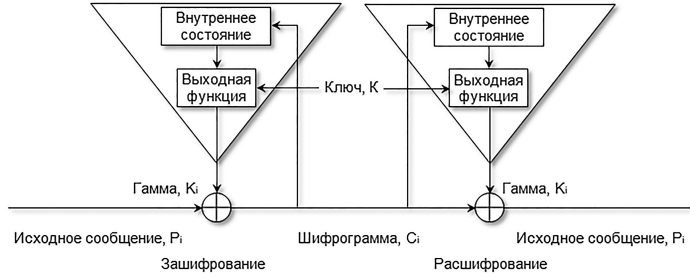


Рисунок 2.39 – Схема шифрования с использованием самосинхронизирующихся генераторов гаммы

Внутреннее состояние зависит от n предыдущих битов шифрограммы. Каждое сообщение начинает случайным заголовком (вектор инициализации, синхропосылка) длиной n бит, после прохождения, которого оба генератора гаммы синхронизируются.

Недостатками самосинхронизируюзихся потоковых шифров является следующее:

1) распространение ошибки. Для каждого бита шифрограммы, искаженного при передаче, расшифровывающий генератор выдает n некорректных битов гаммы. Следовательно, измененный бит влияет на внутреннее состояние, каждой ошибке шифрограммы будет соответствовать n ошибок открытого текста;
2) при потере бита Ci необходимо заново передать часть сообщения, но в отличие от синхронных потоковых шифров, синхронизация генераторов намного проще.

Но в отличии синхронных потоковых шифров генераторы самосинхронизирующихся потоковых шифров могут работать не постоянно, а только на момент передачи сообщений.

К настоящему времени создано большое количество алгоритмов потокового шифрования, таких, например, как: A3, A5, A8, MUGI, PIKE, RC4 и SEAL.

Алгоритм RC4[50] (от английского Rivest cipher или Ron's code) является синхронным потоковым шифром. Алгоритм создан на основе семейства алгоритмов RC1-RC3 сотрудником компании «RSA Security» Рональдом Ривестом в 1987 году. В течение семи лет шифр являлся коммерческой тайной, и точное описание алгоритма предоставлялось только после подписания соглашения о неразглашении, но в сентябре 1994 г. описание алгоритма было анонимно отправлено в список рассылки «Cypherpunks»[51].

Основные преимущества алгоритма RC4 высокая скорость работы, высокая криптостойкость и переменный размер ключа. Типичная реализация выполняет 19 машинных команд на каждый байт текста.

Алгоритм RC4 строится на основе генератора псевдослучайных битов. На вход генератора записывается ключ, а на выходе читаются псевдослучайные биты k_i. Длина ключа может составлять от 40 до 2048 бит[52]. Генерируемые биты последовательности имеют равномерное распределение.

---

[50] Алгоритм также известен как ARC4 или ARCFOUR.

[51] Поскольку RC4 является торговой маркой компании «RSA Security» и официально авторами не публиковался, то для того, чтобы избежать претензий со стороны владельца торговой марки алгоритм называют ARC4 или ARCFOUR (имея в виду английское «alleged RC4» – «предполагаемый» RC4)

[52] В США рекомендуемая длинна ключа внутри страны составляет 128 бит.

Рисунок 2.40 – Эрнст Селмер (1920-2006), Соломон Голоб (1932-2016) и Рон (Рональд) Ривест (1947-)

Шифрование сводится к операции гаммирования. На открытый текст посредством операции суммирования по модулю 2 (операция XOR) накладывается генерируемая последовательность бит. В результате подучается шифрограмма c_i.

$$c_i = m_i \oplus k_i \tag{2.20}$$

Расшифровывание текста сводится также к двум операциям: генерации псевдослучайной последовательности бит на стороне получателя и наложения данной последовательности на шифрограмму опять же посредством XOR.

$$m_i = c_i \oplus k_i = c_i \oplus k_i \oplus k_i \tag{2.21}$$

Главная часть алгоритма – это генератор псевдослучайной последовательности, однозначно определяемый ключом шифрования.

Алгоритм работы генератора псевдослучайной битовой последовательности в RC4 состоит из двух этапов:

- алгоритм KSA[53] – первым и основным этапом в генераторе является реализация функций инициализации, которая использует ключ переменной длины для создания начального состояния генератора ключевого потока. На этом этапе производится инициализация таблицы замен S.
- алгоритм PRGA[54] – на данном этапе вычисляются псевдослучайные числа.

[53] Key-scheduling algorithm.
[54] Pseudo-random generation algorithm.

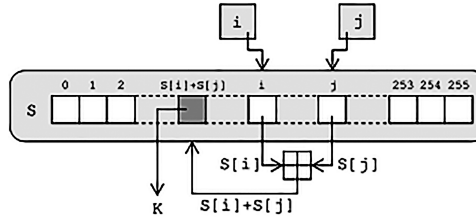


Рисунок 2.41 – Генератор ключевого потока RC4

Алгоритм RC4 является фактически классом алгоритмов, определяемых размером его блока. Этот параметр n является размером слова для алгоритма. Обычно, n = 8, но его можно как уменьшать, так и увеличивать. Так при n = 4 элементов в S-блоке 16, при n = 8 элементов в S-блоке 256. При увеличении n, допустим, до 16 бит, элементов в S-блоке становится 65 536 и соответственно время начальной итерации будет увеличено. Однако, возрастёт и скорость шифрования.

Массив, используемый как таблица замен, называемая *S-бокс*, получил обозначение как S. В каждый момент времени таблица S содержит все возможные n-битовые числа в перемешанном виде. Конкретная перестановка значений в таблице определяется ключом. Так как каждый элемент таблицы принимает значения в промежутке 0 до 255 (при n = 8), то его можно трактовать двояко: либо как число, либо как номер другого элемента в таблице.

Например, для n = 8 S-бокс может быть получен по нижеприведенному алгоритму.

```
int keyLength = key.Length;
for (int i = 0; i < 256; i++)
{
 S[i] = (byte)i;
}

int j = 0;
for (int i = 0; i < 256; i++)
{
 j = (j + S[i] + key[i % keyLength]) % 256;
 S.Swap(i, j);
}
```

После того, как таблица S подготовлена, можно начинать генерацию случайных n-битовых слов.

Например, для n = 8 случайное восьмибитное слово может быть получено по нижеприведенному алгоритму.

```
x = (x + 1) % 256;
y = (y + S[x]) % 256;
S.Swap(x, y);
z = S[(S[x] + S[y]) % 256];
```

Полученное значение z может использоваться в качестве ключа для шифрования очередного блока входного потока данных.

Алгоритм шифрования RC4 применяется в некоторых широко распространённых стандартах и протоколах шифрования (например, WEP, WPA, SSL и TLS).

Следует отметить, что в алгоритме RC4 было обнаружено множество уязвимостей. Они в первую очередь связаны с использованием не случайных или связанных ключей и ситуацией когда один ключевой поток используется повторно. Например, такая проблема криптографически не стойким сделала протокол WEP[55].

По состоянию на 2018 год существует предположение, что некоторые государственные криптографические агентства могут обладать способностью нарушать RC4 при использовании в протоколе TLS. IETF опубликовала RFC 7465, чтобы запретить использование RC4 в TLS; Mozilla и Microsoft выпустили аналогичные рекомендации.

Было сделано несколько попыток усилить алгоритм RC4. Такими алгоритмами стали, в частности, Spritz, RC4A, VMPC, и RC4+.

Вопросы для самоконтроля

1. Что такое симметричная криптосистема?
2. Для чего нужен криптографический ключ в криптосистеме?
3. Могут ли в качестве потоковых шифров использоваться блочные шифры?

[55] Все атаки на WEP основаны на недостатках шифра RC4, таких, как возможность коллизий векторов инициализации и изменения кадров. Для всех типов атак требуется проводить перехват и анализ кадров беспроводной сети. В зависимости от типа атаки количество кадров, требуемое для взлома, различно. С помощью программ, таких как Aircrack-ng, взлом беспроводной сети с WEP шифрованием осуществляется очень быстро и не требует специальных навыков.

4. Какие основные операции преобразования лежат в основе симметричных криптоалгоритмов?
5. Что такое сеть Фейстеля?
6. Для чего используются подстановочно-перестановочные сети?
7. Какие алгоритмы разработана на основе криптоалгоритма AES?
8. Что считают слабым местом криптографического алгоритма ГОСТ 28147-89?
9. Сколько бит информации будет потеряно при расшифровании шифрограммы при потере одного предпоследнего бита шифрограммы в самосинхронизирующемся потоковом шифре?
10. В чем заключаются основные уязвимости протокола WEP?

Рекомендуемая литература

1. А.П.Алферов, А.Ю.Зубов, А.С.Кузьмин, А.В.Черёмушкин. Основы криптографии. – Гелиос АРВ, 2002.
2. Kahn D. The Codebreakers: The Story of Secret Writing. – Macmillan, 1967.
3. А.В.Бабаш, Г.П.Шанкин. Криптография. – М. СОЛОН-ПРЕСС, 2007.
4. Фред Б. Риксон. Коды, шифры, сигналы и тайная передача информации. – Астрель, 2011.
5. Фомичёв В.М. Дискретная математика и криптология: Курс лекций / под ред. Н.Д.Подуфалов. – М.: Диалог-МИФИ, 2013.
6. Жельников В. Криптография от папируса до компьютера. – М.: ABF, 1996.
7. Габидулин Э.М., Кшевецкий А.С., Колыбельников А.И. Защита информации: учебное пособие. – М.: МФТИ, 2011.
8. Chris Christensen. Lester Hill Revisited // Taylor & Francis Group, LLC : Article. 2014.
9. Мао В. Современная криптография: Теория и практика. – М.: Вильямс, 2005.
10. V.N.Krishna, Dr. A.Vinaya Babu. A Modified Hill Cipher Algorithm for Encryption of Data In Data Transmission (англ.) // Computer Science and Telecommunications : Georgian Electronic Scientific Journal. 2007.
11. Federal Information Processing Standards Publication 197 November 26, 2001 Specification for the ADVANCED ENCRYPTION STANDARD (AES)
12. Баричев С.Г., Гончаров В.В., Серов Р.Е. 2.4.2. Стандарт AES. Алгоритм Rijdael // Основы современной криптографии. – 3-е изд. – М.: Диалог-МИФИ, 2011.

13. ГОСТ 28147-89 «Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования»

ГЛАВА 3. СИСТЕМЫ КРИПТОГРАФИИ С ОТКРЫТЫМ (ПУБЛИЧНЫМ) КЛЮЧОМ

Ключевые слова: асимметричная криптосистема, односторонняя функция, односторонняя функция с секретом, публичный или открытый ключ, секретный ключ, инвертированная функция, хэш, хэширование, эллиптическая кривая, алгоритм Диффи-Хеллмана, ключевая и бесключевая хэш-функция, криптографическая стойкость, коллизия, необратимая функция, электронно-цифровая подпись, квалифицированная и простая ЭЦП.

3.1 Односторонние функции и функции-ловушки

XX век отметился появлением нового типа криптографических систем – систем лишенных проблем передачи ключа шифрования между отправителем и получателем криптографически зашифрованных сообщений. Такими системами *стали алгоритмы шифрования с открытым (или, что более правильно, публичным) ключом.*

В основу данных криптографических преобразований положена идея, предложенная У.Диффи и М.Хелманом о разделении ключа для шифрования и ключа для расшифровывания сообщения. Если мы можем обеспечить невозможность получения ключа для расшифрования из ключа для шифрования, то становится возможным построение достаточно надежных криптографических систем и отпадает необходимость хранения в тайне ключа шифрования. Соответственно, возможно сделать публичным свой ключ для шифрования, чтобы каждый заинтересованный мог им зашифровать сообщение для Вас (отсюда и пошло название – публичный или открытый ключ). Однако расшифровать сообщение может только владелец ключа, у которого есть его вторая половина – секретный ключ для расшифрования.

Базисом для понимания получения секретного и публичного ключей положено введенное в 1975 году Диффи и Хелманом понятие односторонней функции (one-way function)[1].

[1] Несмотря на многолетнюю работу математиков, истинная односторонняя функция так и не найдена. Однако, часть свойств этой гипотетической функции достаточно успешно используется в криптографии. Например, такое свойство как сложность (точнее классы сложности) вычисления прямой и инвертированной функций. И если сложность вычисления инвертированной функции становится больше современных вычислительных возможностей, то ее можно применить как условно одностороннюю.

Односторонней называется такая математическая функция $F(X) \rightarrow Y$, обладающая двумя уникальными свойствами:

- существует полиномиальный алгоритм вычисления значений $F(x)$;
- не существует полиномиального алгоритма инвертирования функции F, т.е. решения уравнения $F(X) = Y$ относительно X.

Функция необходима для понимания процесса. Однако в таком виде такая функция применения не нашла. Другим понятием, более близким к понятиям, применяемым в традиционной криптографии, является понятие односторонней функции с секретом[2].

Односторонней функцией с секретом K называется такая функция $F_K(X) \rightarrow Y$, зависящая от параметра K и обладающая тремя свойствами:

- при любом K существует полиномиальный алгоритм вычисления значений $F_K(X)$;
- при неизвестном K не существует полиномиального алгоритма инвертирования F_K;
- при известном K существует полиномиальный алгоритм инвертирования F_K.

Как раз секрет K и выступает в роли ключа расшифрования.

Для криптографии построено множество функций, которые могут считаться односторонними с секретом. Это означает, что для них второе свойство пока строго не доказано, но известно, что задача инвертирования эквивалентна некоторой трудной математической задаче и не решаема на современном техническом уровне. Стоит также отметить, что для некоторых таких функций математиками уже найдены инвертированные функции и их применение в криптографии не обеспечивает защиту информации.

В последнее время такие криптосистемы стали еще называть асимметричными, так как в них есть асимметрия в алгоритмах шифрования и дешифрования. В отличие от таких систем традиционные шифры с одним секретным ключом стали называть симметричными. Для асимметричных систем алгоритм шифрования общеизвестен, но восстановить по нему алгоритм дешифрования за полиномиальное время невозможно

[2] Иногда еще употребляются термины функция с ловушкой, функция опускной двери (английское название: one-way trap-door function).

3.2 Криптосистема RSA

Лауреатами премии Тьюринга за разработку алгоритмов ассиметричного шифрования по праву называют Уитфилда Диффи и Мартина Хеллмана. Однако в своем алгоритме 1975 года они не предложили односторонней функции, удобной для реализации.

Это было сделано в 1977 году тройкой, наверное самых известных математиков XX века: Рональдом Ривестом, Эдди Шамиром и Леонардом Адлеманом из Массачусетского технологического института. Предложенная ими система на основе функции вычисления остатка целочисленного деления оказалась чрезвычайно практичной, а также получила широкое распространение под названием «система RSA» – по первым английским буквам фамилий авторов.

В основе алгоритма имеются два ключа: публичный и секретный. Для их получения необходимо выполнить следующие действия:

- взять два больших простых числа p and q;
- определить n, как результат умножения p on q (n = p•q);
- выбрать случайное число, которое назовем d. Это число должно быть взаимно простым (не иметь ни одного общего делителя, кроме 1) с результатом умножения (p-1)•(q-1);
- определить такое число e, для которого является истинным следующее соотношение (e•d) mod ((p-1) • (q-1))=1.

Далее числа e и n принимаются как публичный или открытый ключ. Соответственно, числа d и n принимаются как секретный ключ.

Для того, чтобы зашифровать данные по открытому ключу {e,n}, необходимо следующее:

- разбить шифруемый текст на блоки, каждый из которых может быть представлен в виде числа M(i) = 0,1,2..., n-1 (т.е. только до n-1).
- зашифровать текст, рассматриваемый как последовательность чисел M(i) по формуле

$$C(i) = M(i)^e \bmod n. \qquad (3.1)$$

Чтобы расшифровать эти данные, используя секретный ключ {d,n}, необходимо выполнить следующие вычисления:

$$M(i) = C(i)^d \bmod n. \qquad (3.2)$$

В результате будет получено множество чисел M(i), которые представляют собой исходный текст.

Пример: зашифруем и расшифруем сообщение «CAB» по алгоритму RSA. Для простоты возьмем небольшие числа – это расчеты сделает проще и понятнее.

1) Выберем p = 3 и q = 11.

2) Определим n = 3•11= 33.

3) Найдем (p-1)•(q-1) = 20. Следовательно, d будет равно, например, 3: (d = 3).

4) Выберем число e по следующей формуле: (e•3) mod 20 = 1. Значит e будет равно, например, 7: (e = 7).

5) Представим шифруемое сообщение как последовательность чисел в диапазоне от 0 до 32 (не забывайте, что кончается на n-1). Буква A = 1, B = 2, C = 3.

Имитируем работу отправителя сообщения и зашифруем сообщение, используя открытый ключ {7,33}

C1 = (3^7) mod 33 = 2187 mod 33 = 9;

C2 = (1^7) mod 33 = 1 mod 33 = 1;

C3 = (2^7) mod 33 = 128 mod 33 = 29;

Далее имитируем работу получателя сообщения и расшифруем данные, используя закрытый ключ {3,33}.

M1 = (9^3) mod 33 = 729 mod 33 = 3 (C);

M2 = (1^3) mod 33 = 1 mod 33 = 1 (A);

M3 = (29^3) mod 33 = 24389 mod 33 = 2 (B).

На самом деле, изложенный способ шифрования очень слаб и никогда не используется. Причина проста – это всего лишь моноалфавитная подстановка – одна и та же буква будет шифроваться одним и тем же числом. Такие шифры люди научились взламывать еще в прошлом тысячелетии. Криптоаналититику даже не нужно выяснять ключи – он дешифрует сообщение, даже не зная о ключах.

Наиболее часто используемым в настоящее время является смешанный алгоритм шифрования, в котором сначала алгоритмом RSA шифруется сеансовый ключ, а потом уже с его помощью участники шифруют свои сообщения симметричными системами, например AES. После завершения сеанса сеансовый ключ, как правило, уничтожается.

В августе 1977 года в колонке «Математические игры» Мартина Гарднера в журнале «Scientific American», с разрешения Рональда Ривеста, появилось первое описание криптосистемы RSA. Читателям также было предложено расшифровать английскую фразу, зашифрованную описанным алгоритмом:

9686	9613	7546	2206
1477	1409	2225	4355
8829	0575	9991	1245
7431	9874	6951	2093
0816	2982	2514	5708
3569	3147	6622	8839
8962	8013	3919	9055
1829	9451	5781	5154

Рисунок 3.1 – Пример шифрограммы Рона Ривеста

Сам Рон Ривест считал, что данное сообщение, зашифрованное 425 битным ключом, может быть расшифровано 40 квадриллионов лет[3]. За расшифрование сообщение Рон Ривест предложил 100 долларов США. Однако, всего через 15 лет данное сообщение было расшифровано группой 600 энтузиастов на 1600 ЭВМ в течение всего лишь полугода[4].

Размер ключа в алгоритме RSA связан с размером модуля n. Два числа p и q, произведением которых является модуль, должны иметь приблизительно одинаковую длину, поскольку в этом случае найти сомножители (факторы) сложнее, чем в случае, когда длина чисел значительно различается. Однако, если два числа чрезвычайно близки друг к другу или их разность близка к некоторому предопределенному значению, то возникает потенциальная угроза безопасности, однако такая вероятность – близость двух случайно выбранных чисел – незначительна.

Оптимальный размер модуля определяется требованиями безопасности: модуль большего размера обеспечивает большую безопасность, но и замедляет работу алгоритма RSA.

Длина модуля выбирается в первую очередь на основе значимости защищаемых данных и необходимой стойкости защищенных данных и во вторую очередь – на основе оценки возможных угроз.

Проведенная еще в 1997 году оценка показала, что 512-битный ключ RSA может быть вскрыт (факторингом) за один миллион долларов США и восемь месяцев работы. В 1999 году 512-битный ключ был вскрыт за семь месяцев и это означает, что 512-битные ключи уже

[3] Аналитики, однако, считали, что для дешифрования сообщения на компьютерах тех лет достаточно всего лишь 20 000 лет.

[4] Вся эта история была прекрасным рекламным ходом для Ривеста, Адлемана и Шамира, запатентовавших RSA, и получивших в результате $900 млн. прибыли.

не обеспечивают достаточную безопасность, за исключением очень краткосрочных задач.

В настоящее время Лаборатория RSA рекомендует для обычных задач ключи размером 1024 бита, а для особо важных задач – 2048 битов.

Следует также отметить, что размеры ключей в криптосистеме RSA (а также и в других криптосистемах с публичным ключом) намного больше размеров ключей систем симметричного шифрования. Однако надежность ключа RSA намного меньше надежности ключа аналогичной длины симметричной системы шифрования.

Алгоритм RSA является, пожалуй, одним из самых распространенных на текущий момент криптоалгоритмов и используется в большом числе криптографических приложений, включая PGP, S/MIME, TLS/SSL, IPSEC/IKE и многих других.

3.3 Криптосистемы, основанные на эллиптических кривых

Эллиптические кривые являются достаточно хорошо изученным математическим аппаратом. Древнейшим дошедшим до нашего времени источником, в котором рассматриваются такие кривые, является «Арифметика» древнегреческого математика Диофанта. Однако до конца XX века они не представляли практической ценности. Всё изменилось в 1985 году.

В 1985 году независимо профессором математики Вашингтонского университета Нилом Коблицем (Neal Koblitz) и математиком исследовательского центра Принстонского института IDA Виктором Миллером (Victor Saul Miller) было предложено использовать в криптографии алгебраические свойства эллиптических кривых. С этого момента началось бурное развитие нового направления криптографии, для которого используется термин «криптография на эллиптических кривых».

Открытие Коблинца-Миллера помогло создать эллиптические варианты алгоритмов Диффи-Хеллмана, Эль-Гамаля, MQV, DSS, ГОСТ Р 34.10-94, которые изначально использовали мультипликативную группу конечного поля. В результате новые алгоритмы (за исключением ГОСТ) получили префикс EC или ECC – Elliptic Curve Cryptography: ECDH, EC ElGamal, ECMQV, ECDSS, а российский ГОСТ Р 34.10-94 трансформировался в ГОСТ Р 34.10-2001 (а потом в более надежный 34.10-2012).

В криптографии на эллиптических кривых роль основной криптографической операции выполняет операция скалярного умножения точки на эллиптической кривой на данное целое число, определяемая через операции сложения и удвоения точек эллиптической кривой. Последние, в свою очередь, выполняются на основе операций сложения, умножения и инвертирования в конечном поле, над которыми рассматривается кривая. Особый интерес к криптографии эллиптических кривых обусловлен теми преимуществами, которые дает её применение в беспроводных коммуникациях – высокое быстродействие и небольшая длина ключа.

Рисунок 3.2 – Нил Коблиц (1948) и Виктор Миллер (1947)

Как понятно из названия, в основу криптографии на эллиптических кривых положены эллиптические кривые. Эллиптические кривые – это не эллипсы. Они так называются просто потому, что описываются кубическими уравнениями, подобными тем, которые используются для вычисления кривой и эллипса.

В общем случае кубические уравнения для эллиптических кривых над конечными полями имеют вид:

$$y^2 + axy + by = x^3 + cx^2 + dx + e, \qquad (3.3)$$

где a, b, c, d и e являются действительными числами, удовлетворяющими некоторым простым условиям.

Это уравнение E может рассматриваться над произвольными полями и, в частности, над конечными полями F, представляющими для криптографии особый интерес.

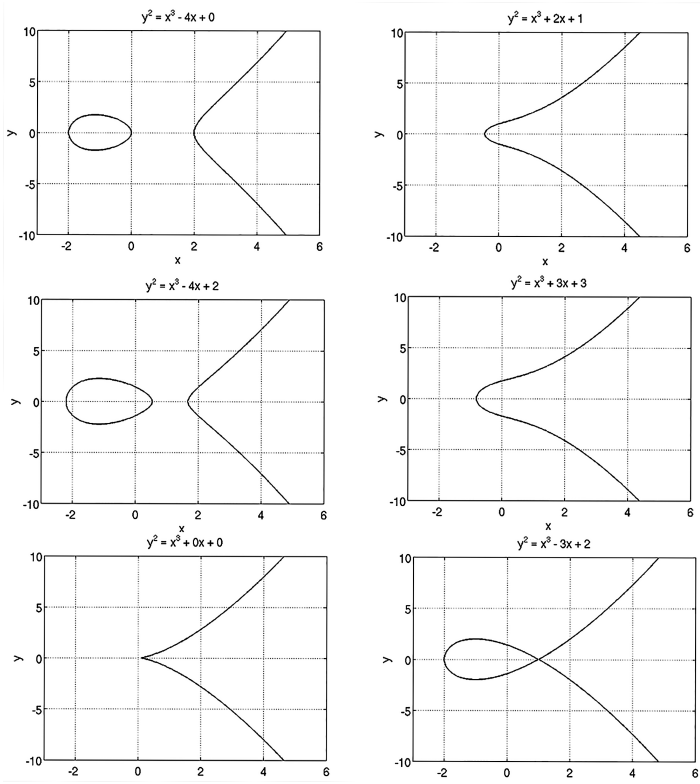


Рисунок 3.3 – Примеры графиков эллиптических кривых

Эллиптические кривые, представленные на первых 4-х рисунках, называются *несингулярными* или *гладкими*. В то время как две нижние кривые относятся к так называемым *сингулярным эллиптическим кривым*.

Уравнение эллиптической кривой задано формулой

$$y^2 = x^3 + ax + b \tag{3.4}$$

или

$$y = \pm\sqrt{x^3 + ax + b}.$$

График данной кривой симметричен относительно оси абсцисс. Точки его пересечения с осью являются решением кубического уравнения

$$x^3 + ax + b = 0 \quad (3.5)$$

Дискриминант данного уравнения $B = \left(\frac{a}{3}\right)^3 + \left(\frac{b}{2}\right)^2$.

В данной ситуации возможны три решения:

– если D < 0, то уравнение имеет три разных действительных корня. Типичными графиками будут графики 1 и 3 на рисунке 3.2;
– если D = 0, то уравнение три корня, два из которых одинаковы. В этом случае имеется особая точка и кривая сингулярная. Типичными графиками будут являться графики 5 и 6 рисунка 3.2;
– если D > 0, то уравнение имеет один действительный корень и два комплексных. Типичными графиками будут 2 и 4 рисунка 3.2.

То есть кривая будет несингулярной при условии, что ее дискриминант не равен 0, что в свою очередь эквивалентно следующему выражению:

$$4a^3 + 27b^2 \neq 0 \quad (3.6)$$

Для гладких кривых, любая прямая, проходящая через две различные точки кривой пересекает ее (кривую) в единственной точке. Кроме того, к любой точке кривой можно провести только одну касательную.

Такие свойства кривой позволяют задавать групповую операцию, называемую сложением точек эллиптической кривой. Так сложение двух точек можно представить графически (рисунок 3.4).

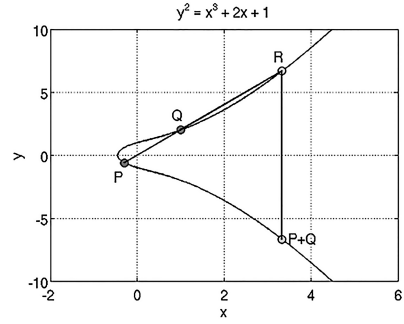


Рисунок 3.4 – Пример получения точек P и Q на эллиптической кривой $y^2 = x^3+2x+1$

Как видно из рисунка, для сложения точек P и Q, необходимо провести между ними прямую линию, которая обязательно пересечет кривую в какой-либо третьей точке R. Эта точка R будет называться суммой двух точек P и Q.

Отразим точку R относительно горизонтальной оси координат и получим искомую точку P+Q.

Для нахождения координат точки P+Q используется выражение 3.7.

$$x_{P+Q} = \alpha^2 - x_P - x_Q$$
$$y_{P+Q} = -y_P + \alpha(x_P - x_R), \qquad (3.7)$$

где $\alpha = (y_Q - y_P)/(x_Q - x_P)$.

Определение эллиптической кривой включает также некий элемент, обозначаемый O и называемый «несобственным элементом»[5].

Осталось уточнить всего один нюанс. Все рассмотренные выше кривые относятся к эллиптическим кривым над вещественными числами. И это приводит к необходимости округления и порождаемым ее проблемам – используя кривые над вещественными числами, невозможно получить биекцию между исходным текстом и зашифрованными данными. Чтобы решить проблему округления в криптографии используются только кривые над конечными полями. Это означает, что под эллиптической кривой понимается набор точек, чьи координаты принадлежат конечному полю.

Для сингулярных кривых решение обратной задачи значительно проще, чем для гладких кривых. Поэтому в криптографии их применение крайне нежелательно. Поскольку неудачный выбор эллиптической кривой может пов-лечь за собой снижение обеспечиваемого уровня безопасности, организации по стандартизации выделяют целые блоки кривых, обладающих необходимой надёжностью. Использование стандартизированных кривых рекомендуется и потому, что становится возможной лучшая совместимость между различными реализациями протоколов информационной безопасности.

В криптографии рассматривается два вида эллиптических кривых:

– над конечным полем Z_P;
– над полем $GF(2^m)$.

У эллиптических кривых над полем $GF(2^m)$ есть одно важное преимущество, элементы поля $GF(2^m)$ могут быть легко представлены в

[5] Также этот элемент иногда называют («бесконечным элементом», «нулевым элементом», «точкой в бесконечности»).

виде n-битных кодовых слов, это позволяет увеличить скорость аппаратной реализации эллиптических алгоритмов.

Все математические операции на эллиптических кривых над конечным полем производятся по законам конечного поля, над которым построена эллиптическая кривая. Т.е. для вычисления, например, суммы двух точек кривой E над кольцом вычетов Z_p все операции производятся по модулю числа p.

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

В случае криптографии с использованием эллиптических кривых приходится иметь дело с редуцированной формой эллиптической кривой, которая определяется над конечным полем. Особый интерес для криптографии представляет объект, называемый эллиптической группой по модулю p, где p является простым числом. Эллиптическая кривая над конечным полем задаётся уравнением

$$y^2 = x^3 + ax + b \ (\text{mod } p). \tag{3.8}$$

Главной арифметической операцией в эллиптической криптографии является операция скалярного умножения точек кривой, которая позволяет определить точку Q

$$Q = k \bullet P \tag{3.9}$$

Скалярное умножение осуществляется посредством нескольких комбинаций сложения и удвоения точек эллиптической кривой. Например, точка 11•P может быть представлена, как 11•P = = 2• (2• (2•P) + P) + P.

Помимо уравнения, важным параметром кривой является базисная (генерирующая) точка G, выбираемая для каждой кривой отдельно.

Секретным ключом в соответствии с технологией эллиптической криптографии является большое случайное число k, а сообщаемым открытым ключом – произведение k на базисную точку G.

Генерация ключей и обмен ими в эллиптической криптографии могут быть выполнен по схеме Диффи-Хеллмана аналогично алгоритму RSA. Сначала выбирается большое простое число p и параметры уравнения кривой. Это задаёт группу точек, в которой выбирается базисная точка G. При выборе G важно, чтобы наименьшее значение n, при котором nG = O, оказалось очень большим простым числом. Уравнение кривой и точка G известны всем участникам процесса. Обмен

ключами между получателем и отправителями сообщения можно провести по следующей схеме:

1) участник А выбирает целое число nA, меньшее n. Это число будет его личным секретным ключом. Затем участник А генерирует открытый ключ РА = nA|G. Открытый ключ представляет собой некую точку на кривой;
2) точно так же участник Б выбирает личный ключ nБ и вычисляет открытый ключ РБ.
3) Участник А генерирует секретный ключ

$$K = nA \times РБ, \tag{3.10}$$

а участник Б генерирует секретный ключ

$$K = nБ \times РА. \tag{3.11}$$

Формулы для секретных ключей дают один и тот же результат:

$$nA \times РБ = nA \times (nБ \times G) = nБ \times (nA \times G) = nБ \times РА. \tag{3.12}$$

Чтобы взломать эту схему, противник должен будет решить задачу дискретного логарифма на кривой, что предполагается трудноразрешимой проблемой.

Пример: сгенерировать общий ключ для двух пользователей по схеме Диффи-Хеллмана, если выбрана эллиптическая кривая $E_{211}(0, -4) \Rightarrow y^2 = (x^3 - 4) \bmod 211$ и точка P(2, 2).

Тогда $y = \sqrt{x3-4} \bmod 211$. Результатом вычисления квадратного корня по модулю p будут являться два числа y и p-y.

Тогда пусть пользователь А выберет секретный ключ c = 121. Тогда пользователь А вычисляет 121P = 121(2, 2) = (115, 48). Пользователь А передает пользователю В свой открытый ключ – точку (115, 48).

Пусть пользователь В выберет секретный ключ d = 203. Тогда пользователь В вычисляет 203P = 203(2, 2) = (130, 203). Пользователь В передает пользователю А свой открытый ключ – точку (130, 203).

Далее пользователи А и В производят следующие вычисления:

- **пользователь А вычисляет 121(130, 203) = (161, 169);**
- **пользователь В вычисляет 203(115, 48) = (161, 169).**

Общим секретным ключом будет точка (161, 169). Ее значения или значения какой-либо функции можно в дальнейшем использовать как секретный ключ для симметричного шифро-

вания. Например, можно взять в качестве секретного ключа абсциссу $161_{10} = 10100001_2$.

Надёжность и криптостойкость эллиптической криптографии основана на трудности решения задачи дискретного логарифма на эллиптической кривой ECDLP (Elliptic Curve Discrete Logarithm Problem) суть которой заключается в отыскании целого числа k по известным точкам P и Q.

При использовании алгоритмов на эллиптических кривых полагается, что не существует субэкспоненциальных алгоритмов для решения задачи дискретного логарифмирования в группах их точек. При этом порядок группы точек эллиптической кривой определяет сложность задачи. Считается, что для достижения такого же уровня криптостойкости как и в RSA, требуются группы меньших порядков, что уменьшает затраты на хранение и передачу информации.

Например, на конференции RSA 2005 Агентство национальной безопасности США объявило о создании «Suite B» («Набор B»), в котором используются исключительно алгоритмы эллиптической криптографии, причём для защиты информации, классифицируемой до «Top Secret», используются всего лишь 384-битные ключи (для примера, в RSA рекомендуется ключ длинной не менее 2048 бит)[6]. Все базовые алгоритмы и протоколы «Suite B», были выстроены на основе криптографии на эллиптических кривых, а для RSA отводилась вспомогательная роль «первого поколения», нужного лишь для плавного перехода к новой, более эффективной так называемой квантово-безопасной криптографии[7].

3.4 Хэш-функции и хэширование

Еще одним важным аспектом применения криптографических преобразований стало *хэширование или вычисление хэш-функций*. Хэш-

[6] На самом деле по оценкам криптоматематиков 256-битные операции ECC эквивалентны работе с модулем длиной 3072 бита в RSA. А 160-битный и 224-битный ключи ECC обеспечивают те же уровни защиты, что и 1024-битный и 2048-битный ключи RSA соответственно.

[7] Под этим общим термином (в другой версии он же звучит как постквантовая криптография) в области защиты информации принято понимать широкий круг всевозможных алгоритмов, протоколов и устройств для коммуникаций, способных противостоять угрозам со стороны квантовых компьютеров.

функции стали основой современной аутентификации, постановки ЭЦП, технологии блокчейн и много другого.

Появился этот термин в середине прошлого века среди специалистов, занимающихся обработках массивов данных. Так в 1953 году Ханс Петер Лун (Hans Peter Luhn) предложил использовать «хэш-кодирование» в программах фирмы IBM. В 1956 году Арнольд Думи (Arnold Dumey) описал идею «хэширования» в таком виде как ее используют в настоящее время. В 1957 году в журнале «IBM Journal of Research and Development» была опубликована статья Уэсли Питерсона (W. Wesley Peterson) о поиске текста в больших файлах. Эта работа считается первой «серьёзной» работой по «хэшированию». Спустя 1963 году опубликована работа Вернера Бухгольца (Werner Buchholz), в которой было проведено обширное исследование «хэш-функций», а в 1968 году Роберт Моррис[8] (Robert Morris) своим обзором по хэшированию ввел понятие о «хэшировании» в научный оборот.

Изначально слово «хэш» происходит от английского «hash», одно из значений которого трактуется как «мешанина» или «путаница». Собственно, это определение довольно полно описывает реальное значение этого термина. Часто про такой процесс говорят, что происходит процесс «хэширования», что опять же является производным от английского hashing («рубить», «крошить», «спутывать» и т.п.).

Рисунок 3.5 – Ханс Петер Лун (1896-1964), Уэсли Питерсон (1924-2009)

В настоящее время *хэширование (англ. hashing) – это преобразование массива входных данных произвольной длины в выходную*

[8] Американский криптограф (1932-2011). Не путать с Моррис, Роберт Тэппэн (род. 1965) – адъюнкт-профессор Массачусетского технологического института; более известный, как создатель первого сетевого червя.

битовую строку установленной длины, выполняемое определённым алгоритмом.

Функция, воплощающая алгоритм и выполняющая преобразование, называется «*хэш-функцией*» или «*функцией свёртки*». Исходные данные называются «входным массивом», «ключом» или «сообщением». Результат преобразования (выходные данные) называется «хэшем», «хэш-кодом», «хэш-суммой», «сводкой сообщения».

Хэш-функцией называется всякая функция $h:X \rightarrow Y$, легко вычислимая и такая, что для любого сообщения M значение $h(M) = H$(свертка) имеет фиксированную битовую длину. Где X – множество всех сообщений, Y – множество двоичных векторов фиксированной длины.

Выделяют два важных вида криптографических хэш-функций – ключевые и бесключевые.

Бесключевые хэш-функции называются кодами обнаружения ошибок. Они дают возможность с помощью дополнительных средств (шифрования, например) гарантировать целостность данных. Основным используемым свойством бесключевых хэш-функции являлась ее чувствительность. Это значит, что при изменении входного массива данных хотя бы в одном бите будет изменено и значение хэш-функции.

Это свойство простых (не надежных, но легко рассчитываемых) хэш-функций нашло применение при контроле целостности передаваемых данных. Например, такой хэш-функцией является функция расчета контрольной суммы (CRC), применяемая как при хранении массивов данных на носителях, так и при передаче данных в вычислительных сетях для выявления аппаратных ошибок и сбоев – так называемое избыточное кодирование). Если рассчитанное значение хэша совпадает с хранимым с файлом или отправленным вместе с пакетом (так называемой контрольной суммой), то значит, потерь при хранении или передаче не было и файл/пакет можно использовать.

Похожая схема используется и в технологии блокчейн, где хэш выступает гарантией целостности цепочки транзакций (платежей) и защищает транзакции от несанкционированных изменений. Благодаря хэшу и распределенным вычислениям взломать блокчейн практически невозможно. На технологии блокчейн разработаны практически все криптовалюты, например, биткоин.

Главное условие для криптографических хэш-функций – невозможность по конечному результату (хэшу) вычислить начальный массив данных. Второе главное условие таких хэш-функций – стойкость к

коллизиями, то есть низкая вероятность получения двух одинаковых хэш-сумм из двух разных массивов данных при обработке их этой функцией[9]. Расчеты по таким алгоритмам более сложны, но здесь уже главный фактор не скорость, а надежность.

Также хэширование используется и при построении электронно-цифровой подписи как инструмент, снижающий вычислительную сложность постановки и верификации подписи, а также ее объем.

Существует множество алгоритмов хэширования, отличающихся различными свойствами. Наиболее часто используют свойства разрядности хэш-функции, вычислительной сложности и криптостойкости. Выбор той или иной хэш-функции определяется спецификой решаемой ею задачи.

К бесключевым функциям предъявляют требования:

– однонаправленность;
– устойчивость к коллизиям;
– устойчивость к нахождению второго прообраза.

Под однонаправленностью или необратимостью понимают высокую сложность нахождения сообщения по заданному значению свертки. Следует отметить, что не доказано существование необратимых хэш-функций, для которых вычисление какого-либо прообраза заданного значения хэш-функции теоретически невозможно. Обычно нахождение обратного значения является лишь вычислительно сложной задачей.

Под устойчивостью к коллизиям понимают сложность нахождения пары сообщений с одинаковыми значениями свертки. Обычно именно нахождение способа построения коллизий криптоаналитиками служит первым сигналом устаревания алгоритма и необходимости его скорой замены.

Под устойчивостью к нахождению второго прообраза понимают сложность нахождения второго сообщения с тем же значением свертки для заданного сообщения с известным значением свертки.

Из семейства бесключевых алгоритмов хэширования наибольшую известность получили такие алгоритмы как CRC16/32, семейство алгоритмов MD2/4/5/6 и семейство алгоритмов SHA.

Алгоритмы CRC16/32 или циклически избыточный код предназначен для проверки целостности данных и используется в помехоустойчивом кодировании. Алгоритм CRC базируется на свойствах деления с остатком двоичных многочленов, то есть многочленов над конечным

[9] Также явление коллизий при получении хэша получило название «эффект близнецов» или «эффект дня рождения».

полем GF(2). Значение CRC является остатком от деления многочлена, соответствующего входным данным, на некий фиксированный порождающий многочлен.

Для нахождения CRC из файла берётся первое слово. Если старший бит в слове «1», то слово сдвигается влево на один разряд с последующим выполнением операции XOR с порождающим полиномом. Соответственно, если старший бит в слове «0», то после сдвига операция XOR не выполняется. После сдвига теряется старый старший бит, а младший бит освобождается – его значение устанавливается равным нулю. На место младшего бита загружается очередной бит из файла, и операция повторяется до тех пор, пока не загрузится последний бит файла. После прохождения всего файла, в слове остается остаток, который и является контрольной суммой. Как уже говорилось, алгоритм не является криптографическим преобразованием и используется только для проверки целостности данных.

Алгоритмы MD1/2/3/4/5/6 являются творением Рональда Райвеста, одного из авторов алгоритма RSA. Алгоритм MD1 был первым алгоритмом в этой линейке алгоритмов, однако спецификация на него никогда не была опубликована. Алгоритм MD2 разработан Ривестом в 1989 году для использования в качестве одного из криптографических алгоритмов, входящих в стандарт защищенной электронной почты PEM (Privacy-Enhanced Mail). Его реализация на языке Си была приведена в RFC 1115. А в 1990 году MD2 был предложен в качестве замены BMAC (Bidirectional MAC). Впоследствии спецификация и обновленная реализация MD2 были опубликованы в RFC 1319. В 2011 году MD2 был официально списан из-за множества успешных криптоатак.

Алгоритм MD3 никогда не был опубликован. По всей видимости, разработка MD3 была заброшена. После MD2 были разработаны MD4, MD5 и MD6 в 1990, 1991 и 2008 годах соответственно.

Рональд Ривест отмечал, что MD4 создавался, прежде всего, как очень быстрый алгоритм хэширования, поэтому он может быть плох в плане криптостойкости. Как показали последовавшие исследования, он был прав, и для приложений, где важна прежде всего криптостойкость, стал использоваться алгоритм MD5. Уязвимости алгоритма MD4 были продемонстрированы в статье Берта ден Бура и Антона Босселарса уже в 1991 году. Первая коллизия была найдена Гансом Доббертином через шесть лет после публикации в 1996 году.

128 битный алгоритм хэширования MD5 имел некогда большую популярность, но первые предпосылки взлома появились еще в конце

девяностых, в начале XXI рядом криптоаналитиков продемонстрированы успешные быстрые способы поиска коллизий. В конце 2008 года US-CERT призвал разработчиков программного обеспечения, владельцев web-сайтов и пользователей прекратить использовать MD5 в любых целях, так как исследования продемонстрировали ненадёжность этого алгоритма, а 2011 году алгоритм официально признан небезопасным и было рекомендовано отказаться от его использования.

Алгоритм MD6 (Message Digest 6) предназначен для получения дайджестов сообщений произвольной длинны. Алгоритм выдвигался на конкурс SHA-3, но, к сожалению, Ривест не успел довести его до кондиции и отозвал его со второго этапа конкурса[10]. В 2009 году Ривест опубликовал данный алгоритм с исправлениями выявленных ошибок. Несмотря на исправления, алгоритм остается достаточно медленным и проигрывает по быстродействию алгоритмам SHA[11].

Алгоритмы линейки SHA были разработаны NIST. В 1993 году NSA совместно с NIST разработали алгоритм безопасного хэширования сейчас известный как SHA-0. Однако вскоре алгоритм был отозван разработчиками и на замену ему в 1998 году предложен 120 битный алгоритм хэширования SHA-1 сообщений произвольной длинны. В настоящее время на алгоритм осуществлено множество успешных как теоретических, так и практических атак[12]. Поэтому ряд компаний отказалось от использования данного алгоритма: например, Google отказался от него в 2014 году, а Яндекс с 2016 года.

В 2002 году АНБ и НИСТ опубликовали новую версию алгоритма хэширования, получившее название SHA-2. Более точно SHA-2 – собирательное название алгоритмов SHA224, SHA256, SHA384 и SHA512. В марте 2012 года вышла последняя на данный момент редакция алгоритма. На текущий момент для алгоритма известны уязвимости найденные индийскими исследователями Сомитра Кумар Санадия и Палаш Саркар. Однако ввиду алгоритмической схожести SHA-2 с SHA-1 и наличия у последней потенциальных уязвимостей принято

---

[10] Комментарий Брюса Шнайера: «Это первоклассный случай самоотвода, который мы могли бы ожидать от Рона Райвиста, особенно учитывая тот факт, что на алгоритм не было никаких атак, в то время как другие алгоритмы подвержены серьёзным атакам, но представляющие их авторы продолжают делать вид, что никто не обращает на это внимание».

[11] MD6-512 медленнее в полтора раза, чем SHA2-512 на 32-битных платформах и почти в четыре раза на 64-бинтых.

[12] 23 февраля 2017 года специалисты из Google и CWI объявили о практическом взломе алгоритма

решение, что SHA-3 будет базироваться на совершенно ином алгоритме. Соответственно, 2 октября 2012 года NIST утвердил в качестве SHA-3 алгоритм Keccak, разработанный группой авторов во главе с Йоаном Дайменом.

В Российской Федерации для получения хэш-функций для криптографии в 1994 году рекомендован к использованию 256 битный Российский стандарт – ГОСТ 34.11-94. В 2008 году командой экспертов из Австрии и Польши была обнаружена техническая уязвимость, сокращающая поиск коллизий в 223 раз. Количество операций, необходимое для нахождения коллизии, таким образом, составляет 2105, что, однако, на данный момент практически не реализуемо. В 2012 году алгоритм признан устаревшим и выведен из эксплуатации. С 1 января 2013 года заменён на ГОСТ Р34.11-2012 «Стрибог».

Существуют также ключевые хэш-функции, не использующие какую-либо основу типа блочного шифрования или вычисления бесключевой хэш-функции, а разработанные независимо с учетом эффективной реализации на современных ЭВМ. Например, ключевая хэш-функция, используемая в алгоритме MAA (Message Authenticator Algorithm), утвержденном стандартом ISO 8731-2.

Ключевые хэш-функции называют кодами аутентификации сообщений. Они дают возможность без дополнительных средств гарантировать как правильность источника данных, так и целостность данных в системах с доверяющими друг другу пользователями.

3.5 Электронно-цифровая подпись

Развитие сети Internet породило множество возможностей. Одной из таких возможностей стал *электронный документооборот*. В настоящее время для совершения юридических операций нет необходимости использовать бумажные носители, править и подписывать их, пересылать друг другу и бояться их потери или хищения. Электронные хранилища информации и средства передачи данных решили эти вопросы. Однако сложнее дело обстояло с подписью данных документов и защитой их от подделок и модификаций.

Все, наверное, помнят бунт мошенничества связанный с подделкой документов после появления цветных принтеров, которые позволяли с высоким качеством распечатать любую ранее отсканированную подпись или печать. Это привело даже к законодательно введенным требованиям ставить штамп «копия» на цветные ксерокопии документов,

хотя, естественно, эта мера абсолютно не защищает от данных видов мошенничества. Еще более осложнилась ситуация с появлением и резким снижением стоимости лазерных гравировальных станков, которые могли в течение минуты за смешные деньги изготовить копию любой печати или клише подписи.

Но это проблемы бумажных носителей информации в которых информация физически прикреплена к носителю – бумаге. В данном случае могут остаться хотя бы какие-либо физические следы манипулирования документами. Сложнее дело с электронными документами. Здесь информация и есть документ. В данном случае также существуют проблемы доверия между участниками процессов, существуют махинации подделки, переделки, ренегатства и повторного использования документов. Для решения данных проблем электронного документооборота и было предложено использование механизмов электронно-цифровой подписи (ЭЦП).

В настоящее время *ЭЦП – это реквизит электронного документа, полученный в результате криптографического преобразования информации с использованием закрытого или секретного ключа подписи.* ЭЦП как механизм позволяет проверить отсутствие искажения информации в электронном документе с момента формирования подписи (целостность), принадлежность подписи владельцу сертификата ключа подписи (авторство), а в случае успешной проверки подтвердить факт подписания электронного документа (неотказуемость).

Цифровая подпись строится на механизмах ассиметричного шифрования. Поэтому естественно ее развитие началось с работы Диффи-Хэллмана в 1976 году и появления системы RSA Ривеста-Шамира-Адлемана в 1977 году. Система RSA и стала первым механизмом постановки и верификации цифровой подписи.

В дальнейшем значимыми вехами развития методов ЭЦП стали:

- 1981 год – разработан алгоритм DSA, который и сейчас используется как стандарт США для электронной подписи;
- 1984 год – создание криптосистемы Эль-Гамаля и форулирование Шафи Гольдвассером, Сильвио Микали и Рональдом Ривестом требований безопасности к алгоритмам цифровой подписи. Ими же были и описаны модели атак на алгоритмы ЭЦП;
- 1991 год – опубликован стандарт на электронную подпись DSS (Digital Signature Standard), разработчиком которого явился Национальный институт стандартизации и технологий (NIST) США. В этом же году разработан закон об электронно-цифровой

подписи в Российской Федерации (хотя принят он будет только в 2001 году);
- 1997 год – принят закон об электронно-цифровой подписи в Германии;
- 2003 год – приняты законы о ЭЦП в Республике Казахстан и Украине.

С точки зрения законодательства в нстоящее время во многих странах введены понятия неквалифицированная электронная подпись и простая электронная подпись.

Квалифицированная отличается от неквалифицированной тем, что квалифицированную подпись выдает аккредитованный удостоверяющий центр, а неквалифицированную – соответственно не аккредитованный центр.

Сейчас надежнее и безопаснее использовать квалифицированную электронную подпись. Неквалицированная подпись сейчас практически не используется. Необходимость в такой подписи была на переходном этапе, когда закон был, а аккредитованных удостоверяющих центров еще не было.

Квалифицированная электронная подпись предназначена для определения лица, подписавшего электронный документ, и является электронным аналогом собственноручной подписи в случаях, предусмотренных законом.

Квалифицированная электронная подпись применяется при совершении гражданско-правовых сделок, оказании государственных и муниципальных услуг, исполнении государственных и муниципальных функций, при совершении иных юридически значимых действий.

Существует несколько схем построения цифровой подписи:
- на основе алгоритмов симметричного шифрования. Данная схема предусматривает наличие в системе третьего лица – арбитра, пользующегося доверием обеих сторон. Авторизацией документа является сам факт зашифрования его секретным ключом и передача его арбитру. Однако такая схема очень уязвима к атакам на арбитра. Единожды скомпроментированный арбитр уже вряд ли будет пользоваться доверием;
- на основе алгоритмов асимметричного шифрования. На данный момент такие схемы электронной подписи наиболее распространены и находят широкое применение как в отношениях «гражданин-правительство», так и «бизнес-бизнес».

Кроме этого, существуют другие разновидности цифровых подписей (групповая подпись, неоспоримая подпись, доверенная подпись), которые являются модификациями описанных выше схем. Их появление обусловлено разнообразием задач, решаемых с помощью ЭЦП.

Поскольку подписываемые документы переменного и зачастую очень большого размера, то в большинстве случаев подпись ставится не на сам документ, а на хэш-функцию данного документа.

Использование хэш-функций даёт следующие преимущества:

– *преимущество вычислительной сложности*. Обычно хэш цифрового документа делается во много раз меньшего объёма, чем объём исходного документа, и алгоритмы вычисления хэша являются более быстрыми, чем алгоритмы ЭЦП. Поэтому формировать хэш документа и подписывать его получается намного быстрее, чем подписывать сам документ;
– *преимущество совместимости*. Большинство алгоритмов оперирует со строками бит данных, но некоторые используют другие представления. Хэш-функцию можно использовать для преобразования произвольного входного текста в подходящий формат;
– *преимущества целостности*. Без использования хэш-функции большой электронный документ в некоторых схемах нужно разделять на достаточно малые блоки для применения алгоритмоп постановки ЭЦП. При верификации достаточно сложно определить, все ли блоки получены и в правильном ли они порядке.

Однако использование хэш-функции не обязательно при электронной подписи, а сама функция не является частью алгоритма ЭЦП, поэтому хэш-функция может использоваться любая или не использоваться вообще.

Также следует отметить, что использование хэширования вносит в схему ЭЦП и свои уязвимости.

Асимметричные схемы ЭЦП относятся к криптосистемам с открытым ключом. В отличие от асимметричных алгоритмов шифрования, в которых шифрование производится с помощью открытого ключа, а расшифровка – с помощью закрытого, в асимметричных схемах цифровой подписи подписание производится с применением закрытого ключа, а проверка подписи – с применением открытого.

Для примера рассмотрим алгоритм постановки и верификации цифровой подписи. Вместо документа используем текст «1 2 3 4 5 6 7». Для простоты понимания возьмем

алгоритм получения хэш-функции, как суммы значений всех чисел сообщения. При генерации ключей используем алгоритм системы RSA:

1) Выберем p = 3 и q = 11.

2) Определим n = 3•11 = 33.

3) Найдем (p-1)•(q-1) = 20. Следовательно, d будет равно, например, 3: (d = 3).

4) Выберем число e по следующей формуле: (e•3) mod 20 = 1. Значит e будет равно, например, 7: (e = 7).

5) Пара значений e и n примем как открытый ключ {7,33} и опубликуем его для того чтобы любой мог проверить истинность подписи.

6) Пару значений d и n примем секретным ключом подписи {3,33} и будем его использовать для постановки ЭЦП

7) Для постановки ЭЦП для текста получим хэш-функцию, которая будет равна h = 1+2+3+4+5+6+7 = 28.

8) Произведем шифрование h с помощью секретного ключа подписи {3,33}

ЭЦП1 = (2^3) mod 33 = 8 mod 33 = 8;
ЭЦП2 = (8^3) mod 33 = 512 mod 33 = 17.

Таким образом, будет передаваться сообщение «1 2 3 4 5 6 7» и его электронная цифровая подпись «8 17». Также будет опубликован открытый ключ владельца подписи {7,33}.

При получении сообщения «1 2 3 4 5 6 7» и его ЭЦП «8 17» выполняем следующую последовательность действий:

1) Вычислим хэш-функцию сообщения, которая будет равна h1 = 1+2+3+4+5+6+7 = 28.

2) Расшифруем ЭЦП с целью получения хэш-функции исходного сообщения h с использованием открытого ключа

h(1) = (8^7) mod 33 = 2 097 152 mod 33 = 2;
h(2) = (17^7) mod 33 = 410 338 673 mod 33 = 8.

Так как вычисленная хэш-функция h1 и расшифрованная хэш-функция совпали, поэтому принимаем решение, что сообщение «1 2 3 4 5 6 7» не подвергалось модификациям и подписано владельцем ключа, что правда.

Рассмотрим пример при получении другого сообщения, имеющего ту же подпись и тот же ключ. Например, получено сообщение «2 3 4 5 6 7 8». Тогда вычисленное получате-

лем значение хэш-функции будет равно 35, а расшифрованное значение будет равно, как мы посчитали выше, 28. Соответственно, значения не равны и вывод, что данный документ (сообщение) не соответствует представленной подписи, что тоже правда.

И, наконец, рассмотрим пример, показывающий возможность возникновения коллизий в хэш-функции и, соответственно, последствий для безопасности. Используемый нами алгоритм хэширования не чувствителен к перестановке символов в сообщении, а также легко позволяет вычислять сообщения-близнецы, имеющие такое же значение хэш-функции. Например, получателю досталось подделанное сообщение «10 8 5 5», ЭЦП и ключ от документа первого примера. В результате вычислений получатель получит h1=28, которая будет совпадать с расшифрованным значением. Естественно, на основании таких данных он примет решение, что сообщение «10 8 5 5» с ЭЦП «8 17» и было отправлено владельцем ключа, хотя это и ложь.

Как видно из примеров, на ЭЦП могут быть осуществлены атаки трех типов: коллизии первого и второго порядка и социальные атаки. Подделка документа (коллизия первого рода).

Коллизия первого порядка или подделка документа заключается в том, что злоумышленник может попытаться подобрать документ к данной подписи, чтобы подпись к нему подходила (как показано в примере). Однако в подавляющем большинстве случаев такой документ может быть только один. Причина в следующем:

- документ все же представляет из себя осмысленный текст и подобрать удовлетворяющие злоумышленника изменения практически невозможно;
- текст документа оформлен по установленной форме, что также значительно усложняет подделку документа (естественно только в терминах ЭЦП);
- документы редко оформляют в виде текстовых-файла, чаще всего в формате DOC или PDF.

Если у фальшивого набора байт и произойдет коллизия с хэшем исходного документа, то выполнение и этих трех условий делает вероятность данной атаки практически равной нулю.

Куда более вероятна атака второго рода. В этом случае злоумышленник фабрикует два документа с одинаковой подписью, и в нужный

момент подменяет один другим. При использовании надёжной хэш-функции такая атака должна быть также вычислительно сложной. Однако эти угрозы могут реализоваться из-за слабостей конкретных алгоритмов хэширования, подписи, или ошибок в их реализациях. В частности, таким образом можно провести атаку на SSL-сертификаты и алгоритм хэширования MD5.

Социальные атаки направлены не на взлом алгоритмов цифровой подписи, а на манипуляции с открытым и закрытым ключами пользователей:

- злоумышленник, укравший закрытый ключ, может подписать любой документ от имени владельца ключа;
- злоумышленник может обманом вынудить владельца секретного ключа передать этот ключ ему;
- злоумышленник может обманом заставить владельца подписать какой-либо документ, например, используя протокол слепой подписи;
- злоумышленник может подменить открытый ключ владельца на свой собственный, выдавая себя за него.

Использование протоколов обмена ключами, установление срока жизни ключей ЭЦП и защита закрытого ключа от несанкционированного доступа позволяет снизить опасность этих атак.

3.6 Перспективы развития и проблемы применения криптосистем с открытым (публичным) ключом

Развитие облачных технологий, криптовалют и просто электронных денег, технологии блокчейн, да и «уход» в электронный бизнес, естественно, будет повышать требования к применяемым технологиям криптографической защиты данных, надежной аутентификации пользователей и обеспечения юридической защищенности сделок. Решение этих вопросов невозможно без использования методов распространения ключей в незащищенной цифровой среде – *методов ассиметричной криптографии.*

Ожидаемое появление квантовых компьютеров, распараллеливание вычислений требует от исследователей в области криптографической защиты принципиального усиления криптографических алгоритмов. И одним из таких шагов в области ассиметричного шифрования стало появление *алгоритма SHA3.*

Естественно, это приведет к увеличению вычислительной сложности криптоалгоритмов. Поэтому одним из возможных бизнес-подходов может стать активное внедрение легких криптоалгоритмов, которые позволяют получить практически ту же криптостойкость, при значительно меньшей вычислительной стойкости.

Также не следует отбрасывать тот факт, что вся современная ассиметричная криптография построена на отсутствии результатов по построению математических или вычислительных алгоритмов решения обратных задач. И никто не гарантирует непоявление таких алгоритмов в будущем. Так, например, в 1994 году был принят первый российский стандарт в области электронной цифровой подписи – ГОСТ Р34.10-94 «Информационная технология. Криптографическая защита информации. Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма». Он определял процедуры работы с ЭЦП на основе схемы Эль Гамаля. Невозможность подделки подписи основана на сложности решения задачи дискретного логарифмирования в поле из p элементов. Однако математика не стоит на месте, и математиками разработан так называемый *метод решета числового поля*. С его помощью можно «взломать» ЭЦП, сформированную по схеме Эль Гамаля, по крайней мере в случае 512-битного модуля p.

Вопросы для самоконтроля

1. Что такое асимметричная криптосистема?
2. Кто использует открытый (публичный) ключ в ассиметричной криптосистеме?
3. Кто предложил идею ассиметричного шифрования ?
4. Какая система ассиметричного шифрования была первой?
5. Какой алгоритм шифрования положен США в основу постквантовой криптографии?
6. Какова рекомендуемая длина ключей в ассиметричных криптосистемах?
7. Что такое хэш-функция?
8. Какие алгоритмы хэширования в настоящее время рекомендованы для использования в США?
9. Что такое электронно-цифровая подпись?

10. Какие алгоритмы постановки и верификации ЭЦП применяются в Республике Казахстан?

Рекомендуемая литература

1. А.П.Алферов, А.Ю.Зубов, А.С.Кузьмин, А.В.Черёмушкин. Основы криптографии. – Гелиос АРВ, 2002.
2. Kahn D. The Codebreakers: The Story of Secret Writing.- Macmillan, 1967.
3. А.В.Бабаш, Г.П.Шанкин. Криптография. – М. СОЛОН-ПРЕСС, 2007.
4. Фомичёв В.М. Дискретная математика и криптология: Курс лекций / под ред. Н.Д.Подуфалов. – М.: Диалог-МИФИ, 2013.
5. Габидулин Э.М., Кшевецкий А.С., Колыбельников А.И. Защита информации: учебное пособие. – М.: МФТИ, 2011.
6. Мао В. Современная криптография: Теория и практика. – М.: Вильямс, V. Miller, Use of elliptic curves in cryptography, Advances in cryptology-CRYPTO 85, Springer Lecture Notes in Computer Science vol 218, 1985.
7. Приказ Росстандарта от 7 августа 2012 г. № 216-ст. Проверено 31 мая 2013.
8. Diffie W., Hellman M. E. New Directions in Cryptography // IEEE Trans. Inf. Theory / F. Kschischang. – IEEE, 1976.
9. Rivest R., Shamir A., Adleman L. A method for obtaining digital signatures and public-key cryptosystems // Commun. ACM. – New York City: ACM, 1978.
10. Приказ ЦБ РФ от 31.01.1995 N 02-13 «О вводе в действие в системе центрального банка российской федерации государственных стандартов Российской Федерации» (рус.) ПРИКАЗ ЦБ РФ № 02-13.
11. Rivest R. RFC 1321, The MD5 Message-Digest Algorithm: The MD5 Message-Digest Algorithm // Request for Comments. – Internet Engineering Task Force, 1992/
12. Ah Kioon, Mary Cindy, Wang Z., Deb Das S. Security Analysis of MD5 Algorithm in Password Storage // Applied Mechanics and Materials. 2013.
13. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си = Applied Cryptography. Protocols, Algorithms and Source Code in C. – М.: Триумф, 2002.
14. Рябко Б.Я., Фионов А.Н. Основы современной криптографии для специалистов в информационных технологиях. – Научный мир, 2004.

15. Нильс Фергюсон, Брюс Шнайер. Практическая криптография = Practical Cryptography: Designing and Implementing Secure Cryptographic Systems. – М.: Диалектика, 2004.
16. Закон Республики Казахстан от 7 января 2003 года «Об электронном документе и электронной цифровой подписи».
17. ГОСТ Р34.11-2012 «Информационная технология. Криптографическая защита информации. Функция хэширования».
18. Б.А.Фороузан. Схема цифровой подписи Эль-Гамаля // Управление ключами шифрования и безопасность сети / Пер. А.Н.Берлин. – Курс лекций.
19. Menezes A.J., Oorschot P. v., Vanstone S.A. Handbook of Applied Cryptography. – CRC Press, 1996.
20. Некоммерческий проект проверки криптостойкости алгоритмов при помощи распределенных вычислений – www.distributed.net.

ГЛАВА 4. УПРАВЛЕНИЕ КРИПТОГРАФИЧЕСКИМИ КЛЮЧАМИ

Ключевые слова: центр распределения ключей, оказия, сессионный ключ, SSL, TSL

4.1 Управление ключами в системах с открытым (публичным) ключом

Одной из главных сфер применения схемы шифрования с открытым ключом является решение проблемы распределения ключей. Имеются две совершенно различные области использования шифрования с открытым ключом в этой сфере:

- распределение открытых ключей;
- использование шифрования с открытым ключом для распределения секретных ключей.

Для распределения открытых ключей предложено несколько методов. Фактически их можно сгруппировать в следующие общие классы:

- публичное объявление;
- публично доступный каталог;
- авторитетный источник открытых ключей;
- сертификаты открытых ключей.

Самым простым способом распространения является публичное объявление открытых ключей или так называемое *неконтролируемое распределение ключей.* В данном методе любая участвующая в обмене данными сторона может предоставить свой открытый ключ любой другой стороне или передать ключ по средствам коммуникаций вообще для всех.

Этот подход удобен, но имеет один недостаток: такое публичное объявление может сделать кто угодно, в том числе и злоумышленник. Это значит, что кто-то представившись пользователем А может послать открытый ключ другому пользователю сети или предложить такой открытый ключ для всеобщего пользования. Пока пользователь А откроет подлог и предупредит других пользователей, фальсификатор сможет прочитать все шифрованные сообщения, пришедшие за это время для А, и сможет использовать фальсифицированные ключи для аутентификации.

Второй путь – это использование публично доступного каталога или централизованная схема. За сопровождение и распространение публичного каталога в такой ситуации должен отвечать некоторый надежный центр. Такая схема должна включать следующие элементы:

– уполномоченный объект, поддерживающий каталог с записями вида {имя, открытый ключ} для каждого из участников;
– каждый участник регистрирует свой открытый ключ. Такая регистрация должна происходить либо при личной явке участника, либо по защищенным каналам коммуникации;
– при компрометации ключа участник может заменить существующий ключ новым в любой момент с использованием средств аутентификации;
– каталог регулярно обновляется.

Эта схема более защищена, чем индивидуальные публичные объявления, но и она уязвима. Если противнику удастся получить личный ключ объекта, уполномоченного вести каталог, он может выдавать фальсифицированные открытые ключи и, следовательно, выступать от имени любого из участников обмена данными и читать сообщения, предназначенные любому участнику.

Третий вариант – это использованием авторитетного источника открытых ключей. Сценарий такого варианта предполагает наличие некоторого центра распределения ключей, уполномоченного поддерживать динамический каталог открытых ключей всех участников обмена данными. Кроме того, каждому из участников достоверно известен открытый ключ центра, но только центр знает соответствующий личный ключ. При этом выполняются следующие действия:

– инициатор А посылает сообщение с меткой даты/времени (оказией N_1) авторитетному источнику открытых ключей с запросом о текущем открытом ключе участника В;

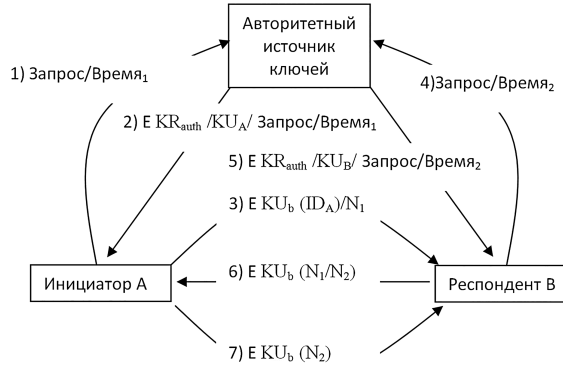


Рисунок 4.1 – Сценарий распределения ключей при использовании авторитетного источника ключей

– авторитетный источник отвечает сообщением, которое шифруется с использованием личного ключа авторитетного источника KR_{auth}. Это сообщение инициатор А может дешифровать, используя открытый ключ авторитетного источника. Поэтому отправитель А может быть уверенным в том, что сообщение исходит от авторитетного источника. Это сообщение должно включать: открытый ключ участника В (KU_b), оригинальный запрос, (чтобы сторона А имела возможность убедиться, что запрос не был изменен на пути к авторитетному источнику) и оригинальную метку даты/времени (оказия N_1), чтобы отправитель А мог удостовериться, что это ответ именно на данный запрос;
– инициатор А сохраняет открытый ключ участника В и использует его для шифрования сообщения, направляемого получателю В и содержащего идентификатор отправителя А (ID_A) и оказию N_1;
– респондент В получает открытый ключ участника А от авторитетного источника точно таким же способом, каким отправитель А получил открытый ключ получателя В.

К этому моменту открытые ключи оказываются доставленными участникам А и В, так что теперь А и В могут начать защищенный обмен данными. Но перед этим им желательно выполнить два следующих дополнительных действия:

– респондент В посылает сообщение инициатору А, шифрованное с помощью KU_A и содержащее оказию отправителя А (N_1), а также новую оказию, сгенерированную участником В (N_2). Присутствие N_1 в этом сообщении убеждает участника А в том, что отправителем полученного сообщения был В;
– инициатор А возвращает N_2 шифрованное с помощью открытого ключа участника В, чтобы тот мог убедиться в том, что отправителем ответа является А.

В этом варианте распределения ключей потребуется семь сообщений. Однако отсылать первые четыре сообщения требуется нечасто, так как и обе стороны могут сохранить открытые ключи друг друга для дальнейшего использования, что обычно называют *кэшированием*.

В четвертом варианте используют сертификаты открытых ключей. В предыдущем сценарии Авторитетный источник является узким местом системы. Альтернативный подход предложил в 1978 году Лорен Конфельдер (Loren Kohnfelder). В данном методе каждый сертификат содержит открытый ключ и другую информацию, создается

авторитетным источником сертификатов и выдается участнику. В системе предъявляются следующие требования:

– любой участник должен иметь возможность прочитать сертификат, чтобы определить имя и открытый ключ владельца сертификата;
– любой участник должен иметь возможность проверить, что сертификат исходит от авторитетного источника сертификатов и не является подделкой;
– только авторитетный источник сертификатов должен иметь возможность создавать и изменять сертификаты.

Деннинг (Denning) к этим правилам добавил следующее требование – любой участник должен иметь возможность проверить срок действия сертификата.

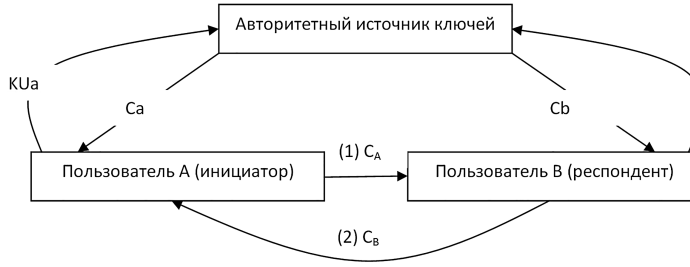


Рисунок 4.2 – Обмен сертификатами открытых ключей

Каждый участник в такой системе обращается к центру сертификатов, предоставляя открытый ключ и запрашивая для него по защищенной форме связи сертификат.

Центр пересылает сертификаты C_A и C_B, содержащие

– время действия сертификата;
– идентификатор владельца;
– открытый ключ владельца сертификата.

При этом сертификаты зашифрованы с помощью личного ключа авторитетного источника.

При этом пользователь А может переслать сертификат любому участнику.

Получатель сертификата использует открытый ключ KUauth центра сертификатов, чтобы прочитать сертификат. Это дает гарантию, что сертификат пришел именно от него.

4.2 Протокол обмена секретным ключом

При симметричном шифровании обе стороны должны иметь один и тот же секретный ключ. Поэтому криптографическая надежность любой симметричной криптографической системы значительно зависит от используемой системы распределения ключей.

Для двух сторон А и В распределение ключей возможно организовать четырьмя различными способами:

- ключ выбран стороной А и физически доставлен В;
- ключ выбирает третья сторона и физически доставляет А и В;
- одна из сторон передает новый ключ в зашифрованном виде, используя старый ключ;
- третья сторона С доставляет ключ А и В по защищенным каналам связи, т.е. используется некий Центр распределения ключей (ЦРК).

В таком случае для симметричных криптосистем схема (протокол) распределения ключей может быть централизованной и распределенной (с посредником и самодостаточной).

Использование ЦРК предполагает организацию иерархии ключей (минимум два уровня). Связь между конечными пользователями шифруется с использованием временного ключа, называющегося *сеансовым ключом*. Сеансовый ключ получают от ЦРК по тем же каналам связи, что используются для доставки данных. Сеансовые ключи передаются в зашифрованном виде, а для их шифрования используется главный или мастер-ключ, общий для ЦРК и данного пользователя.

В такой схеме главных ключей требуется N (по числу пользователей). Их распределяют некриптографическим способом (физической доставкой адресату).

Предположим, что при использовании централизованной схемы распределения ключей пользователь А хочет передать информацию пользователю В и для защиты данных требуется одноразовый сеансовый ключ.

При этом пользователь А имеет секретный ключ $К_a$, известный только ему и ЦРК, а пользователь В имеет $К_b$ ($К_a$ и $К_b$ – главные ключи пользователей А и В соответственно, $К_s$ – одноразовый сеансовый ключ).

Обмен информацией происходит следующим образом:

1. Пользователь А посылает запрос в ЦРК на получение сеансового ключа для защиты связи с В. При этом посылаемый запрос должен включать:

– информацию, позволяющую однозначно определить А и В (ID_A, ID_B);
– некоторый идентификатор N_1, уникальный для каждого запроса и называемый *оказией*. Оказией может быть время, счетчик, случайное число.

2. ЦРК отвечает на запрос пользователя А, шифруя ответ ключом Ка (главным ключом пользователя А). Единственным пользователем, кто сможет прочесть ответ, является А (следовательно, А уверен, что сообщение пришло от ЦРК).

Сообщение-ответ ЦРК включает следующие элементы, предназначенные для А:

– одноразовый сеансовый ключ K_s (для связи пользователя А с пользователем В);
– запрос с оказией N_1, чтобы пользователь А мог сопоставить ответ с запросом.

Таким образом, пользователь А может удостовериться что его запрос не был изменен на пути в ЦРК, а оказия не позволяет перепутать ответ на данный запрос с ответом на предыдущие запросы.

Сообщение-ответ ЦРК включает следующие элементы, предназначенные для В.

– одноразовый сеансовый ключ K_s;
– идентификатор пользователя А – ID_A.

Оба элемента шифруются с помощью ключа K_B (главного ключа ЦРК и пользователя В).

3. Пользователь А сохраняет свой сеансовый ключ и пересылает пользователю В информацию от ЦРК, предназначенную для В.

Пользователь В получает K_s и знает, что полученная информация пришла от ЦРК (так как она зашифрована K_B, который знают только В и ЦРК).

Сеансовый ключ, таким образом, есть как у пользователя А, так и В. Но перед обменом данными желательно выполнить следующее:

4. Используя полученный сеансовый ключ K_s пользователь В посылает пользователю А новую оказию N_2.

5. С помощью K_s пользователь А в ответ возвращает $f(N_2)$. Это необходимо, чтобы убедить пользователя В в том, что первоначально полученное им сообщение не было воспроизведено злоумышленником.

Таким образом, обеспечивается не только передача ключа, но и аутентификация (шаги 4 и 5).

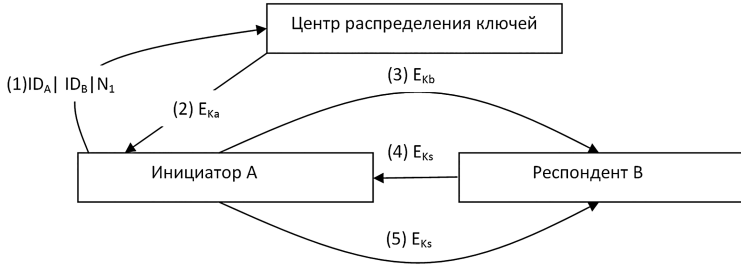


Рисунок 4.3 – Централизованная схема распределения секретных ключей

Следует отметить, что необязательно возлагать функцию распределения ключей на один ЦРК. Более выгодно использовать некоторую иерархию ЦРК. Чем чаще меняются сеансовые ключи, тем более они надежны, но распределение сеансовых ключей задерживает начало сеанса обмена данными и увеличивает загрузку сети.

Использование ЦРК предполагает, что ЦРК должен внушать доверие и быть надежно защищенным от посягательств. От этих требований можно отказаться, если использовать децентрализованную схему распределения ключей (самодостаточную).

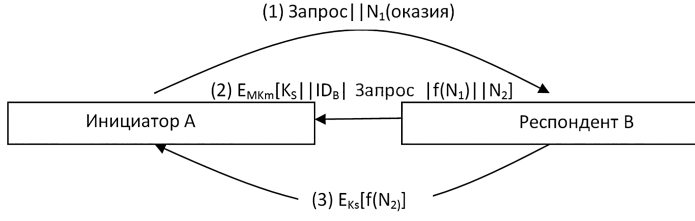


Рисунок 4.4 – Децентрализованная схема распределения секретных ключей

В децентрализованной схеме распределения ключей сеансовый ключ может быть определен в результате следующей последовательности действий:

- пользователь А посылает запрос на получение K_s + оказия N_1;
- пользователь В отвечает, шифруя ответ с использованием общего у А и В главного ключа E_{MKm};
- пользователь А возвращает $f(N_2)$, шифруя с помощью K_s.

4.3 Сертификаты безопасности

Сертификаты безопасности или открытых ключей – это структуры данных, предназначенные для хранения, распространения или пересылки по незащищенным каналам открытых ключей с гарантией их целостности и аутентичности (принадлежности конкретным субъектам).

Цель использования сертификатов – сделать открытые ключи одних субъектов (взаимодействующих сторон) доступными для других так, чтобы их аутентичность (т.е. принадлежность конкретным субъектам) и действительность можно было надежно проверять.

В сертификате идентифицирующая информация об объекте (стороне, владельце ключа) достоверно связывается с его открытым ключом. Достоверность обеспечивается проверкой подписи издателя – удостоверяющего центра или центра сертификации, которому вы доверяете.

Сертификат открытого ключа – это цифровой документ (структура данных), состоящий из раздела данных и раздела подписи.

Раздел данных содержит открытые данные включающие, как минимум, открытый ключ и идентифицирующую сторону информацию (имя объекта и дополнительные сведения). Пользовательское имя в формате отличительного имени (DN). DN определяет название пользователя, и любые дополнительные атрибуты, требуемые для уникального идентификатора пользователя (например, DN может содержать пользовательский номер служащего). Открытый ключ требуется для того, чтобы пользователи могли реализовать услуги по защите информации через использование открытых ключей в сертификате.

Дополнительная информация может включать:

- срок действия открытого ключа;
- серийный номер или имя, идентифицирующее сертификат или ключ;
- дополнительную информацию о владельце сертификата (например, обычный или сетевой адрес);
- дополнительную информацию о ключе (например, алгоритм и намечаемое использование);
- особые характеристики, относящиеся к идентификации представляемого объекта, генерированию ключевой пары или другим проблемам политики;
- информацию, облегчающую проверку подписи;

– конкретные операции, для которых должен использоваться открытый ключ (шифрования данных или ЭЦП).

Когда говорят об издании сертификата, то подразумевается подписание раздела данных органом сертификации. Издавая сертификат, издатель удостоверяет подлинность (дает свои гарантии подлинности) связи между открытым ключом субъекта и идентифицирующей его информацией.

Одним из первых протоколов, использующих сертификаты был протокол SSL. SSL (англ. Secure Sockets Layer – уровень защищённых сокетов) – криптографический протокол, который подразумевает более безопасную связь. Он использует асимметричную криптографию для аутентификации ключей обмена, симметричное шифрование для сохранения конфиденциальности, коды аутентификации сообщений для целостности сообщений.

SSL изначально разработан компанией *Netscape Communications* для добавления протокола HTTPS в свой web-браузер Netscape Navigator. Впоследствии на основании протокола SSL 3.0 был разработан и принят стандарт RFC, получивший имя TLS. В 2014 году правительство США сообщило об уязвимости в текущей версии протокола SSL. SSL должен быть исключён из работы в пользу TLS.

По сути, SSL-сертификат – цифровая подпись сайта, подтверждающая его подлинность и безопасность клиенту. Использование сертификата позволяет защитить как владельца сайта, так и его клиентов. SSL-сертификат даёт возможность владельцу применить к своему сайту технологию SSL-шифрования. *Таким образом, назначение SSL-сертификата – обеспечить безопасное соединение между сервером и браузером пользователя, надёжно защитить данные от перехвата и подмены.*

Приобретают сертификаты обычно не напрямую у удостоверяющего центра, а через партнёров. В Казахстане и России продажей сертификатов известных удостоверяющих центров, таких как Comodo, Geotrust, GoDaddy, GlobalSign, Symantec и прочих, занимается множество компаний. Корневые SSL-сертификаты этих центров предустановлены в качестве доверенных во всех популярных браузерах.

Существуют сертификаты разных уровней проверки. Для защиты персональных данных пользователей обычно используют сертификат с упрощённой проверкой – DV (Domain validation).

Следующий уровень – *сертификат OV (Organization validation) для организаций, применяемый для проверки связи между доменным име-*

нем, хозяином домена и использующей сертификат компанией. То есть такой сертификат удостоверяет не только доменное имя, но и то, что сайт принадлежит действительно существующей организации.

Для более качественной проверки компании и её полномочий на приобретение сертификатов используются так называемые сертификаты с расширенной проверкой – EV (Extended validation).

Также существуют и *национальные сертификаты безопасности.*

4.4 Анонимное распределение ключей

В случае, если пользователи сами не могут выбирать собственные ключи, то они должны пользоваться услугами центра распределения ключей. Проблема заключается в том, что ключи должны распределяться так, чтобы никто не мог определить, кто получил какой ключ. Процедура распределения ключей в этом случае получила название «анонимное распределение ключей» и может выглядеть так:

- пользователь А выбирает пару «открытый ключ, секретный ключ»;
- ЦРК генерирует непрерывный поток ключей;
- ЦРК шифрует ключи, один за другим, своим открытым ключом;
- ЦРК передаёт зашифрованные ключи, один за другим, в сеть;
- пользователь А выбирает ключ случайным образом;
- пользователь А шифрует выбранный ключ своим открытым ключом;
- пользователь А ожидает некоторое время и посылает дважды зашифрованный ключ обратно в ЦРК;
- ЦРК расшифровывает дважды зашифрованный ключ своим секретным ключом, оставляя ключ зашифрованным один раз открытым ключом пользователя А;
- ЦРК посылает зашифрованный ключ назад пользователю А;
- пользователь А расшифровывает ключ своим секретным ключом.

Полностью анонимная сессия может быть установлена при использовании, например как в SSL, алгоритма RSA или Диффи-Хеллмана для создания ключей обмена. В случае использования RSA клиент шифрует свой секретный ключ (pre_master_secret) с помощью открытого ключа несертифицированного сервера. Открытый ключ клиент узнает из сообщения обмена ключами от сервера. Результат посылается в сообщении обмена ключами от клиента к серверу. Поскольку

перехватчик не знает закрытого ключа сервера, то ему будет невозможно расшифровать секрет (pre_master_secret).

При использовании алгоритма Диффи-Хеллмана открытые параметры сервера содержатся в сообщении обмена ключами от сервера, и клиенту посылают в сообщении обмена ключами. Перехватчик, который не знает приватных значений, не сможет найти секрет (pre_master_secret).

Вопросы для самоконтроля

1. В чем заключается проблема обмена ключами?
2. Каким образом реализуется обмен ключами для симметричного шифрования?
3. Кто предложил идею обмена ключами?
4. Какая система обмена ключами в настоящее время наиболее распространена?
5. Какие центры распределения ключей существуют?
6. Каковы проблемы использования центров распределения ключей?
7. Какие протоколы используют механизмы управления криптографическими ключами?
8. Почему США рекомендовало отказаться от использования SSL-сертификатов?
9. Что такое TSL-сертификат?
10. Какие сертификаты безопасности применяются в Республике Казахстан?

Рекомендуемая литература

1. Бабаш А.Б. Криптографические методы защитф информации. – М.: КНОРУС, 2018
2. Sepehrdad P. Discovery and Exploitation of New Biases in RC4. – Springer Berlin Heidelberg, 2011.
3. Menezes A.J., Paul C. van Oorschot, Scott A. Vanstone. Handbook of Applied Cryptography. 1996.
4. Claessens J. Computer Security and Industrial Cryptography. – 3-t. – Leuven-Heverlee, Belgium, 2002.
5. Viega J. Network Security with OpenSSL. – 1-st. – O'Reilly Media, USA, 2002.

6. Rescorla E.. SSL and TLS: Designing and Building Secure Systems. – 1-st. – Addison-Wesley Professional, 2000.
7. Thomas S. SSL & TLS Essentials: Securing the Web. – 1-st. – Wiley, February 11, 2000.
8. Шнайер Б. Прикладная криптография. – М. Наука, 2006

ГЛАВА 5. КРАТКИЕ СВЕДЕНИЯ О КРИПТОАНАЛИЗЕ

Ключевые слова: криптоанализ, криптографическая атака, полный перебор, перебор по словарю, частотный анализ, дифференциальный анализ, линейный анализ, атаки по побочным каналам.

5.1 Краткая история возникновения криптоанализа

Термин «криптоанализ» в научную терминологию введен в 1920 году американским криптографом Уильямом Ф. Фридманом. И хотя этому термину менее сотни лет, сами действия по дешифрованию секретных сообщений стали предприниматься с момента появления первых шифров[1].

В VIII-XV веках значительный вклад в развитие криптоанализа сделали арабские ученые. В тот период написаны труды Халиля аль-Фарахиди («Книга тайного языка»), Аль-Кинди («Манускрипт о дешифровке криптографических сообщений»[2]), Шихаба аль-Калкашанди («Субх ал-Ааша»[3]).

В XV-XIX веках развитие механизмов дешифрования перехваченных сообщений активно стало развиваться и в Европе. Труды таких известных европейских ученых тового времени как Леон Баттиста Альберти, Иоганн Трисемус, Фридрих Касиски, Огюст Керкхоффс посвящены как описанию шифров, так и вопросам криптоанализа.

Также в этот период в Европе стали выделяться и первые государственные службы, в которых занимались вопросами дешифрования перехваченных секретных сообщений. Одной из первых таких служб стали организованные военным министром Франсуа Лавуа французские «черные кабинеты» (фр. Cabinet Noir). Английский черный кабинет появился в 1655 году при секретной службе Парламента (англ. Secret Office) для перлюстрации писем. Во времена правления императрицы Марии Терезии в Австрии была создана дешифровальная служба (нем. Geheime Kabinets-Kanzlei). На пике своей эффективности, пришедшемся на 1730-1760 годы, Kabinets-Kanzlei считалась лучшей организацией подобного рода в Европе. В России датой учреждения первой государственной шифровальной службы можно считать

[1] В качестве примера можно привести одно из первых известных ныне дешифровальных устройств – «антисциталу».

[2] В книге Аль-Кинди встречается первое известное упоминание о частотном криптоанализе.

[3] В книге Шихаба ал-Калкашанди опять же приведены первые известные ныне таблицы частоты появления букв в арабском языке на основе текста Корана.

1549 год – образование «посольского приказа» с «циферным отделением».

После Первой мировой войны, формально закрытые в 1911 году «черные кабинеты» перешли под юрисдикцию военных разведок и контрразведок. В этот период в криптоанализ активно внедряются электромеханические дешифровальные устройства, наиболее известным из которых явилась «Bomb machine» Алана Тьюринга.

Появление электронных вычислительных машин породило новые направления криптоанализа.

Первоначально методы криптоанализа основывались на лингвистических закономерностях естественного текста и реализовывались с использованием ручных вычислений. Со временем в криптоанализе возросла роль чисто математических методов анализа, реализация которых невозможна без использования вычислительной техники. В настоящее время для криптоанализа используются, как правило, специализированные криптоаналитические компьютеры.

5.2 Методы криптоанализа

В настоящее время под термином «криптоанализ» понимается наука, занимающаяся вопросами оценки сильных и слабых сторон методов шифрования, а также разработкой методов, позволяющих взламывать криптосистемы.

	Частотный анализ	Полный перебор	Атака по ключам	Факторизация/дискретное логарифмирование	Метод «встреча посередине»	Дифференциальный анализ	Линейный анализ	Метод коллизий	Анализ по побочным каналам	Квантовый анализ
Симметричные шифры	■	■	■			■	■		■	
Ассиметричные шифры		■	■		■				■	■
Хэш-функции		■			■			■		

Рисунок 5.1 – «Цели» методов криптоанализа

Попытку раскрытия конкретного шифра с применением методов криптоанализа в настоящее время называют криптографической атакой на этот шифр. Криптографическую атаку, в ходе которой раскрыть шифр удалось, называют «взломом» или «вскрытием» шифра.

Одним из первых видов криптографических атак был частотный анализ (англ. frequency analysis). Метод основан на том, что частота появления заданной буквы алфавита в достаточно длинных текстах одна и та же для разных текстов одного языка. При этом, в случае моноалфавитного шифрования, если в шифротексте будет символ с аналогичной вероятностью появления, то можно предположить, что он и является указанной зашифрованной буквой. Аналогичные рассуждения применяются к N-граммам в случае полиалфавитных шифров. Частотный метод породил требование равномерного распределения символов в шифртексте. Сегодня принципы частотного анализа широко применяются в программах по подбору паролей и позволяют на несколько порядков сократить время поиска.

Метод полного перебора (англ. brute-force) также иногда называют «*методом грубой силы*». Возможность вскрывать шифры методом перебора ключей появилась у криптоаналитиков с распространением высокопроизводительной вычислительной техники.

При осуществлении попытки атаки на основе только шифртекста требуется анализировать выходные данные алгоритма и проверять их «осмысленность». Задачу выделения осмысленного текста, то есть определения факта правильной дешифрации, решают при помощи ЭВМ с использованием так называемых «цепей Маркова» или конечных автоматов.

Атака по ключам (англ. dictionary attack). В большинстве используемых людьми ключей наблюдается фонетическое сходство со словами естественного языка, вызванные простотой запоминания такого рода информации в отличие от случайно сгенерированных ключей. Поэтому использование специально сгенерированных словарей позволяет значительно сократить время подбора ключей.

Генераторы псевдослучайных чисел – еще один источник угрозы для стойкости криптосистемы. Если для генерации ключей используется криптографический слабый алгоритм, то независимо от используемого шифра вся система будет нестойкой.

Наибольший прогресс в разработке методов раскрытия блочных шифров был достигнут в самом конце XX века, и связан с появлением

двух методов – разностного, или дифференциального, криптоанализа и линейного криптоанализа.

Метод дифференциального криптоанализа сочетает в себе идею общей линейной структуры с применением вероятностно-статистических методов исследования. Основан на изучении разностей между шифруемыми значениями на различных раундах для пары подобранных открытых сообщений при их шифровании с одним и тем же ключом. Метод предложен в 1990 году израильскими специалистами Эли Бихамом и Ади Шамиром. Является статистической атакой, в результате работы которой предлагается список наиболее вероятных ключей шифрования.

Возникновение этого метода криптоанализа привело к появлению требования на равномерность распределения разности шифртекстов, на соответствие которому шифры проверялись на известных конкурсах, таких как AES и NESSIE. Также следует отметить, что дифференциальный анализ применим и для взлома хеш-функций.

Подобно дифференциальному анализу, *линейный анализ* является комбинированным методом, сочетающим в себе поиск линейных статистических аналогов для уравнений шифрования и статистический анализ имеющихся открытых и шифрованных текстов, использующий также методы согласования и перебора. Этот метод исследует статистические линейные соотношения между отдельными координатами векторов открытого текста, соответствующего шифртекста и ключа, и использует эти соотношения для определения статистическими методами отдельных координат ключевого вектора.

Метод линейного криптоанализа позволил получить наиболее сильные результаты по раскрытию ряда итерационных систем блочного шифрования, в том числе и системы DES. Метод линейного криптоанализа в неявном виде был предложен еще в работе Шона Мерфи также в 1990 году, где он успешно применялся при анализе системы блочного шифрования FEAL. В 1992 году старший научный сотрудник Mitsubishi Electric Company Мицуру Мацуи (Mitsuru Matsui) формализовал этот подход, а позже успешно применил его к анализу криптоалгоритма DES.

Практически все используемые алгоритмы асимметричной криптографии основаны на задачах факторизации (например, криптосистема RSA) и дискретного логарифмирования в различных алгебраических структурах (схема электронно-цифровой подписи Эль-Гамаля). Поэтому для криптоанализа асимметричных криптосистем можно приме-

нять универсальные методы – например, метод «встречи посередине». Другой подход заключается в решении обратной математической задачи, положенной в основу асимметричного шифра. За последние годы в области изучения проблемы факторизации целых чисел и дискретного логарифмирования наблюдался значительный прогресс. Подтверждением этому может служить следующий факт: в 1977 году считалось, что разложение на множители 125-разрядного числа потребует 40 квадриллионов лет, однако уже в 1994 году было факторизовано число, состоящее уже из 129 двоичных разрядов.

Наиболее эффективные сегодня алгоритмы факторизации и дискретного логарифмирования имеют уже не экспоненциальную, а субэкспоненциальную временную сложность. Это алгоритмы, использующие факторную базу. Первый субэкспоненциальный алгоритм для вычисления дискретного логарифма в простом поле Zp был предложен Леонардом Адлеманом. На практике алгоритм Адлемана оказался недостаточно эффективным; Дон Копперсмит (Don Coppersmith), Эндрю Одлизко (Andrew Odlyzko) и Ричард Шреппель (Richard C. Schroeppel) предложили свою версию субэкспоненциального алгоритма дискретного логарифмирования – «COS», а алгоритм решета числового поля, предложенный Оливером Широкауэром, при $p > 10^{100}$ работает эффективнее и различных модификаций метода COS.

Рисунок 5.2 – Мицуру Мацуи (1961), Эндрю Майкл Одлизко (1949), Ричард Шреппель (1948)

Ряд успешных атак на системы, основанные на сложности дискретного логарифмирования в конечных полях, привел к тому, что американские и российские стандарты электронной цифровой подписи, которые были приняты в девяностых годах и базировались на схеме Эль-Гамаля, в двухтысячных годах были обновлены и переведены на эллиптические кривые.

Атаки по сторонним, или побочным, каналам используют информацию, которая может быть получена с устройства шифрования и не является при этом ни открытым текстом, ни шифртекстом. Такие атаки основаны на корреляции между значениями физических параметров, измеряемых в разные моменты во время вычислений, и внутренним состоянием вычислительного устройства, имеющим отношение к секретному ключу. Этот подход менее обобщенный, но зачастую более мощный, чем классический криптоанализ.

В последние годы количество криптографических атак, использующих слабости в реализации и размещении механизмов криптоалгоритма, резко возросло. Противник может замерять время, затрачиваемое на выполнение криптографической операции, или анализировать поведение криптографического устройства при возникновении определенных ошибок вычисления. Другой подход предполагает отслеживание энергии, потребляемой системой в процессе выполнения операций с секретным ключом (например, расшифрования или генерации подписи). Побочную информацию собрать порой несложно – сегодня выделено более десяти побочных каналов, в том числе электромагнитное излучение, ошибки в канале связи, кэш-память и световое излучение.

В 1994 году Питер Шор открыл так называемый «ограниченно-вероятностный» алгоритм факторизации, который позволяет с помощью квантового компьютера разложить на множители число за полиномиальное от размерности задачи время. Алгоритм Шора разложения чисел на множители явился главным достижением в области квантовых вычислительных алгоритмов. Именно с этого момента началось усиленное финансирование работ по созданию квантовых компьютеров.

Важно отметить, что алгоритм Шора чрезвычайно прост и довольствуется гораздо более скромным аппаратным обеспечением, по сравнению с устройством универсального квантового компьютера. В настоящее время уже есть конкретные результаты в области квантовой криптографии. Так, корпорация IBM в 2017 году на саммите Института инженеров электротехники и электроники (IEEE Industry Summit on the Future of Computing) объявила о создании прототипа 50-кубитного квантового компьютера, а на облачном 20-кубитном квантовом компьютере с 2016 года могут поработать все желающие.

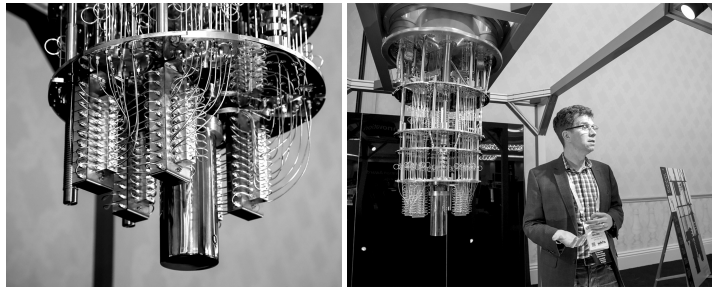


Рисунок 5.3 – 50-кубитный квантовый компьютер IBM (2017 год)

Вопросы для самоконтроля

1. Что такое криптоанализ?
2. Какие атаки используют на моноалфавитные и полиалфавитные шифры?
3. В чем заключается основная уязвимость моноалфавитного шифра?
4. Почему метод сплошного перебора не применяется к слабоструктурированным и логически несвязанным сообщениям?
5. Какие атаки используют для взлома симметричных шифров?
6. В чем смысл дифференциального криптоанализа?
7. Почему ключи шифрования не должны содержать логически взаимосвязанных цепочек символов?
8. Что такое линейный криптоанализ?
9. Какая из криптосистем обладает большей криптостойкостью в настоящее время: система Эль-Гамаля или система на эллиптических кривых?
10. Почему для криптографии опасны квантовые компьютеры?

Рекомендуемая литература

1. Rejewski, Marian. Summary of Our Methods for Reconstructing ENIGMA and Reconstructing Daily Keys, and of German Efforts to Frustrate Those Methods. – Appendix C to Kozaczuk, 1984
2. Панасенко, С. Современные методы вскрытия алгоритмов шифрования. – Chief Information Officer, 2006.

3. Авдошин С., Савельева А. Криптоанализ: вчера, сегодня, завтра.- Открытые системы, 2009
4. Шнайер Б. Криптоанализ // Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си = Applied Cryptography. Protocols, Algorithms and Source Code in C. – М.: Триумф, 2002.
5. Пилиди В.С. Криптография. Вводные главы. – Ростов-на-Дону: ЮФУ, 2009.

ЗАКЛЮЧЕНИЕ

История криптографии насчитывает уже несколько тысячелетий – от примитивной криптографии, построенной на простейшей моноалфавитной подстановке, до постквантовой криптографии. Также изменялись технические устройства криптографии, от папируса и сциталы, цилиндров Джефферсона, роторных шифровальных машин, до современных квантовых криптографических компьютеров и электронных систем хранения и передачи данных.

История криптографии знает множество «взлетов» и «падений». В какие-то моменты в науке преобладали методы защиты, в какие-то – методы криптоанализа. Были моменты и когда криптографию приравнивали к запрещенному знанию. Однако наука успешно развивается и служит инструментом развития коммуникаций человечества.

Во многих отношениях криптография в настоящее время занимает центральное место среди программно-технических регуляторов безопасности. Например, для портативных компьютеров, планшетов или смартфонов, физически защитить которые крайне трудно, только криптография позволяет гарантировать конфиденциальность информации даже в случае кражи.

Вызовы XX века – появление вычислительной техники – породили появление *нового направления симметричного шифрования – блочного шифрования, сетей Фейстеля и подстановочно-перестановочных сетей.*

В наше время особую роль приобретает знание криптографических способов защиты данных, идентификации и аутентификации пользователей в связи с развитием облачных технологий и электронного документооборота. Появление этих технологий на особую роль выдвинуло такие проблемы как безопасное распространение ключей шифрования во враждебной среде, возможность распараллеливания вычислений при атаках грубой силы. Эти вызовы времени породили появление таких решений как шифрование с открытым (публичным) ключом, хэширование, электронно-цифровая подпись.

Развитие информационных ресурсов и глобальных сетей породило появление сертификатов безопасности и, в частности, сделало популярным SSL-сертификаты.

Появление к конце XX века криптовалют – биткоина, дарккоина, лайткоина – породило появление технологий блокчейн.

Последние вызовы – появление квантовых компьютеров – потребовало от криптографии развития новых защищенных от «взлома» на квантовых компьютерах алгоритмов шифрования и стандартов, построенных на них. Эти вызовы породили появление алгоритмов шифрорвания на эллиптических кривых и появление алгоритма хэширования SHA3.

Новые вызовы нового времени диктуют требование развитие новых направлений криптографии – квантовую криптографию, вероятностное, функциональное и медовое шифрование, ДНК-шифрование и многое другое.

СПИСОК РЕКОМЕНДОВАННЫХ ИСТОЧНИКОВ

1. Ah Kioon, Mary Cindy, Wang Z., Deb Das S. Security Analysis of MD5 Algorithm in Password Storage // Applied Mechanics and Materials. 2013.
2. Chris Christensen. Lester Hill Revisited // Taylor & Francis Group, LLC: Article. 2014.
3. Claessens J. Computer Security and Industrial Cryptography. – 3-t. – Leuven-Heverlee, Belgium, 2002.
4. Diffie W., Hellman M. E. New Directions in Cryptography // IEEE Trans. Inf. Theory / F. Kschischang. – IEEE, 1976.
5. Federal Information Processing Standards Publication 197 November 26, 2001 Specification for the ADVANCED ENCRYPTION STANDARD (AES)
6. Gardner M. A new kind of cipher that would take millions of years to break.- Mathematical Games, Scientific American, 1978. – 237(2).
7. ISO/IEC 10116 1997. Information technology – Security techniques – Modes of operation for an n-bit block cipher
8. ISO/IEC 10118-1 2000. Information technology – Security techniques – Hash- functions – Part 1: General
9. ISO/IEC 10118-2 2000. Information technology – Security techniques – Hash-functions – Part 2: Hash-functions using an n-bit block cipher
10. ISO/IEC 11770-1 1996. Information technology – Security techniques – Key management – Part 1: Framework
11. ISO/IEC 11770-2 1996. Information technology – Security techniques – Key management – Part 2: Mechanisms using symmetric techniques
12. ISO/IEC 13335-1 2004. Information technology – Security techniques – Management of information and communications technology security – Part 1: Concepts and models for information and communications technology security management
13. ISO/IEC 13888-1 2004. IT security techniques – Non-repudiation – Part 1: General
14. ISO/IEC 13888-3 1997. Information technology – Security techniques – Non-repudiation – Part 3: Mechanisms using asymmetric techniques
15. ISO/IEC 17799 2000. Информационные технологии – Практические правила управления информационной безопасностью
16. ISO/IEC 9796-2 2002. Information technology – Security techniques – Digital signature schemes giving message recovery – Part 2: Integer factorization based mechanisms.

17. ISO/IEC 9796-3 2000. Information technology – Security techniques – Digital signature schemes giving message recovery – Part 3: Discrete logarithm based mechanisms.
18. ISO/IEC 9797-2 2002. Information technology – Security techniques – Message Authentication Codes (MACs) – Part 2: Mechanisms using a dedicated hash-function.
19. ISO/IEC 9798-1 1997. Information technology – Security techniques – Entity authentication – Part 1: General
20. ISO/IEC 9798-2 1999/ Information technology – Security techniques – Entity authentication – Part 2: Mechanisms using symmetric encipherment algorithms
21. ISO/IEC 9798-3 1998. Information technology – Security techniques – Entity authentication – Part 3: Mechanisms using digital signature techniques
22. ISO/IEC 9798-4 1999. Information technology – Security techniques – Entity authentication – Part 4: Mechanisms using a cryptographic check function
23. ISO/IEC 9979 1999. Information technology – Security registration of cryptographic algorithms
24. ISO/IEC TR 13335-3 1998. Information technology – Guidelines for the management of IT Security – Part 3: Techniques for the management of IT Security
25. Kahn D. The Codebreakers: The Story of Secret Writing. – Macmillan, 1967.
26. Luciano D., Prichett G. Cryptology: From Caesar Ciphers to Public-Key Cryptosystems. – The College Mathematics Journal. – Mathematical Association of America, 1987. – Vol. 18, Iss. 1.
27. Menezes A.J., Paul C. van Oorschot, Scott A. Vanstone. Handbook of Applied Cryptography.- 1996.
28. Miller, Use of elliptic curves in cryptography, Advances in cryptology – CRYPTO 85, Springer Lecture Notes in Computer Science vol 218, 1985.
29. Rejewski, Marian. Summary of Our Methods for Reconstructing ENIGMA and Reconstructing Daily Keys, and of German Efforts to Frustrate Those Methods. – Appendix C to Kozaczuk, 1984
30. Rescorla E.. SSL and TLS: Designing and Building Secure Systems. – 1-st. –Addison-Wesley Professional, 2000.
31. Rivest R. RFC 1321, The MD5 Message-Digest Algorithm: The MD5 Message-Digest Algorithm // Request for Comments. – Internet Engineering Task Force, 1992/

32. Rivest R., Shamir A., Adleman L. A method for obtaining digital signatures and public-key cryptosystems // Commun. ACM. – New York City: ACM, 1978.
33. Sepehrdad P. Discovery and Exploitation of New Biases in RC4. – Springer Berlin Heidelberg, 2011.
34. Thomas S. SSL & TLS Essentials: Securing the Web. – 1-st. – Wiley, February 11, 2000.
35. V.N.Krishna, Dr. A.Vinaya Babu. A Modified Hill Cipher Algorithm for Encryption of Data In Data Transmission (англ.) // Computer Science and Telecommunications : Georgian Electronic Scientific Journal. 2007.
36. Viega J. Network Security with OpenSSL. – 1-st. – O'Reilly Media, USA, 2002.
37. Абдикаликов К.А., Задирака В.К. Элементы современной криптологии и методы защиты банковской информации. – Алматы.: Республиканский издательский кабинет Казахской академии образования имени И. Алтынсарина, 1999
38. Авдошин С., Савельева А. Криптоанализ: вчера, сегодня, завтра. – Открытые системы, 2009
39. Амиров А. Ж., Султанова Б. К., Шаханов Д. Ж. История развития криптологии. Этапы. – Молодой ученый. 2016. – №1.
40. Бабаш А.Б. Криптографические методы защиты информации. – М.: КНОРУС, 2018
41. Бабаш А.В., Шанкин Г.П. Криптография (аспекты защиты). – М.: СОЛОН-ПРЕСС, 2007. – 512 с.
42. Баричев С.Г., Гончаров В.В., Серов Р.Е. 2.4.2. Стандарт AES. Алгоритм Rijdael // Основы современной криптографии. – 3-е изд. – М.: Диалог-МИФИ, 2011.
43. Болотов А.А., Гашков С.Б., Фролов А.Б., Часовских А.А. Алгоритмические основы эллиптической криптографии. – М.: МАИ, 2000
44. Габидулин Э.М., Кшевецкий А.С., Колыбельников А.И. Защита информации: учебное пособие. – М.: МФТИ, 2011.
45. Габидулин Э.М., Кшевецкий А.С., Колыбельников А.И. Защита информации: учебное пособие. – М.: МФТИ, 2011.
46. ГОСТ 28147-89 «Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования»
47. ГОСТ 28147-89 Системы обработки информации. Защита криптографическая. Алгоритмы криптографического предобразования.
48. ГОСТ Р34.10-94 Информационная технология. Криптографическая защита информации. Процедуры выработки и проверки электрон-

ной цифровой подписи на базе асимметричного криптографического алгоритма.
49. ГОСТ Р34.11-2012 «Информационная технология. Криптографическая защита информации. Функция хэширования».
50. Жельников В. Криптография от папируса до компьютера. – М.: ABF, 1996.
51. Закон Республики Казахстан от 7 января 2003 года «Об электронном документе и электронной цифровой подписи».
52. Информационный портал «Закон KZ». www.zakon.kz
53. Информационный портал «Информационная безопасность». www.sec.ru
54. Котухов М.М., Марков А.С. Законодательно-правовое и организационно-техническое обеспечение информационной безопасности автоматизированных систем. – СПб.: БХВ-Петербург, 1998
55. Кулябов Д.С. Защита информации в сетях. – СПб.: БХВ-Петербург, 2004. – 130 с.
56. Максимов Ю.Н. Технические методы и средства защиты информацию. – СПб.: Полигон, 2000
57. Мао В. Современная криптография: Теория и практика. – М.: Вильямс, 2005.
58. Некоммерческий проект проверки криптостойкости алгоритмов при помощи распределенных вычислений – www.distributed.net.
59. Панасенко С. Современные методы вскрытия алгоритмов шифрования.- Chief Information Officer, 2006.
60. Пилиди В.С. Криптография. Вводные главы. – Ростов-на-Дону: ЮФУ, 2009.
61. Приказ Росстандарта от 7 августа 2012 г. № 216-ст. Проверено 31 мая 2013.
62. Приказ ЦБ РФ от 31.01.1995 N 02-13 «О вводе в действие в системе центрального банка российской федерации государственных стандартов Российской Федерации» (рус.) ПРИКАЗ ЦБ РФ № 02-13.
63. Риксон Ф.Б. Коды, шифры, сигналы и тайная передача информации.- Астрель, 2011.
64. Рябко Б.Я., Фионов А.Н. Основы современной криптографии для специалистов в информационных технологиях. – Научный мир, 2004.
65. Сингх С., Книга шифров. Тайная история шифров и их расшифровки. – М.: Астрель, 2007. – 448 с.

66. Соболева Т.А. История шифровального дела в России. – М.: ОЛМА-ПРЕСС Образование, 2002.
67. Спивак С.И., Вильданов А.Н., Зарипова Л.И. Достижения и приложения современной информатики, математики и физики: материалы III Всероссийской научно-практической заочной конференции (г. Нефтекамск, 20-22 октября 2014 г.). – Уфа: РИЦ БашГУ, 2014.
68. Уголовный кодекс Республики Казахстан
69. Фергюсон Н., Шнайер Б. Практическая криптография = Practical Cryptography: Designing and Implementing Secure Cryptographic Systems. – М. : Диалектика, 2004.
70. Фомичёв В.М. Дискретная математика и криптология: Курс лекций / под ред. Н.Д.Подуфалов. – М.: Диалог-МИФИ, 2013.
71. Фороузан Б.А. Схема цифровой подписи Эль-Гамаля // Управление ключами шифрования и безопасность сети / Пер. А.Н.Берлин. – Курс лекций.
72. Хорев А.А. Защита информации от утечки по техническим каналам. Часть 1. Технические каналы утечки информации. Учебное пособие. – М.: Гостехкомиссия России, 1998
73. Черемушкин А.В. Лекции по арифметическим основам криптографии. – М.: МЦНМО, 2002
74. Чмора А.Л. Современная прикладная криптография. – М.: «Гелиос АРВ», 2001.
75. Шнайер Б. Криптоанализ // Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си = Applied Cryptography. Protocols, Algorithms and Source Code in C. – М.: Триумф, 2002.
76. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си = Applied Cryptography. Protocols, Algorithms and Source Code in C. – М.: Триумф, 2002.

М.Ю.Зарубин, Г.С.Ыбытаева

CRYPTOGRAPHIC SYSTEMS

КРИПТОГРАФИЧЕСКИЕ СИСТЕМЫ

TEXTBOOK
УЧЕБНОЕ ПОСОБИЕ

ISBN 978-601-7660-08-6

Компьютерная верстка, дизайн обложки – **Любовицкая Ольга**

Подписано в печать в 2021 г.
Формат 60x84 1/16. Объем 20 печ.л.
Гарнитура Times New Roman. Печать офсетная.
Заказ № ____. Тираж 300 экз.

Издательство «Бастау».
Гос. лицензия № 0000036
Министерства образования и науки РК.
Сертификат Национальной государственной
книжной палаты РК №155 о присвоении
международного регистрационного кода 978-601-281.
Национальный сертификат «Лидер отрасли-2018»
Национального бизнес-рейтинга Республики Казахстан.
г. Алматы, пр. Сейфуллина, 458/1.
Тел.: 279 49 53, 279 97 32.

Отпечатано в типографии
«Полиграфсервис» (тел.: 233 32 53).
г. Алматы, ул. Зеленая, 13а.