IRSTI 81.93.29 DOI: https://doi.org/10.26577/JMMCS.2021.v112.i4.12

N.A. Kapalova, K.S. Sakan*, A. Haumen, O.T. Suleimenov
RK MES SC Institute of Information and Computational Technologies, Kazakhstan, Almaty
*e-mail: kairat_sks@mail.ru

REQUIREMENTS FOR SYMMETRIC BLOCK ENCRYPTION ALGORITHMS DEVELOPED FOR SOFTWARE AND HARDWARE IMPLEMENTATION

The hardware and software cryptographic information protection facility is one of the most important components of comprehensive information security in information and communication systems and computer networks. This article outlines and systematizes the basic requirements for modern cryptographic information protection facilities (CIPFs), and describes the stages of developing a symmetric block encryption algorithm. Based on the basic requirements for CIPFs, criteria for evaluating the developed cryptographic encryption algorithms were determined. The possibilities and necessary limitations of the types of cryptographic transformations (primitives) in the software and hardware implementation of the developed symmetric block encryption algorithm are considered and defined. On the basis of the developed encryption algorithm, SDTB Granit plans to implement a model of a hardware-software complex for off-line (linear) data encryption, taking into account all the listed requirements and characteristics. The article presents a new version of the "AL01" encryption algorithm, which is guided by the basic requirements for creating symmetric block ciphers.

Key words: symmetric block algorithm, cryptographic information protection facility, software, hardware-software, and hardware implementation of encryption algorithms, cryptographic primitives.

Н.А. Қапалова, Қ.С. Сақан*, А. Хаумен, О.Т. Сүлейменов
ҚР БҒМ ҒК Ақпараттық және есептеуіш технологиялар институты, Қазақстан, Алматы қ.
*e-mail: kairat_sks@mail.ru

Бағдарламалық-аппараттық іске асыру үшін әзірленетін симметриялық блоктық шифрлау алгоритмдеріне қойылатын талаптар

Ақпараттық-коммуникациялық жүйелері мен компьютерлік желілерде ақпараттық қауіпсіздігін кешенді түрде қамтамасыз етуде маңызды құрамдас бөліктің бірі болып ақпаратты криптографиялық қорғаудың бағдарламалық-аппараттық құралдары саналады. Бұл еңбекте ақпаратты криптографиялық тұрғыда қорғаудың заманауи құралдарына (АКҚҚ) қойылатын негізгі талаптар баяндалған және жүйелендірілен, симметриялық блоктық шифрлау алгоритмін әзірлеу кезеңдері көрсетілген. АКҚҚ-ға қойылатын негізгі талаптарын ескере отырып, әзірленетін криптографиялық шифрлау алгоритмдерін бағалау критерийлері талқыланған. Әзірленетін симметриялы блоктық шифрлау алгоритмін бағдарламалық-аппараттық іске асыруда криптографиялық түрлендірулердің (примитивтердің) түрлерінің ерекшеліктері, мүмкіндіктері мен қажетті шектеулері қарастырылып, нақтыланды.

Әзірленіп жатқан шифрлау алгоритмі "Гранит" АКТБ базасында көрсетілген талаптар мен сипаттамаларды ескере отырып, деректерді алдын ала (желілік) шифрлауға арналған бағдарламалық-аппараттық кешен жасап шығару жоспарлануда. Симметриялы блокты шифрлар жасаудың негізгі талаптарын ескере отырып, мақаланың мақалада "AL01" шифрлау алгоритмінің жаңа нұсқасы көрсетілген.

Түйін сөздер: симметриялы блоктық алгоритмі, ақпаратты криптографиялық қорғаудың құралдары, бағдарламалық, бағдарламалы-аппараттық және аппараттық іске асырудың шифрлау алгоритмдері, криптографиялық примитивтер.

Н.А. Капалова, К.С. Сакан*, А. Хаумен, О.Т. Сулейменов
Институт информационных и вычислительных технологий КН МОН РК, Казахстан, г. Алматы
*e-mail: kairat_sks@mail.ru

Требования к симметричным блочным алгоритмам шифрования, разрабатываемым для программно-аппаратной реализации

Программно-аппаратное средство криптографической защиты информации является одним из важнейших составляющих комплексного обеспечения информационной безопасности в инфокоммуникационных системах и компьютерных сетях. В данной статье изложены и систематизированы основные требования к современным средствам криптографической защиты информации (СКЗИ), приведены этапы разработки симметричного блочного алгоритма шифрования. Исходя из основных требований, предъявляемых к СКЗИ, определены критерии оценки разрабатываемых криптографических алгоритмов шифрования. Рассмотрены и определены возможности и необходимые ограничения видов криптографических преобразований (примитивов) при программно-аппаратной реализации разрабатываемого симметричного блочного алгоритма шифрования. На основе разрабатываемого алгоритма шифрования в СКТБ "Гранит" планируется реализовать модель программно-аппаратного комплекса для предварительного (линейного) шифрования данных с учетом всех перечисленных требований и характеристик. В статье представлена новая версия алгоритма шифрования "AL01" учитывающая все основные требования, предъявляемые при создании симметричных блочных шифров.

Ключевые слова: симметричный блочный алгоритм, средство криптографической защиты информации, программная, программно-аппаратная и аппаратная реализации алгоритмов шифрования, криптографические примитивы.

1 Introduction

Today cryptography is a relatively new research area at the intersection of mathematics and computer science and is related to information security. Its role and application to ensure confidentiality and integrity when processing information in modern information and telecommunication systems have become an integral part of the life of modern society [1-3].

To ensure the inaccessibility of the semantic part of confidential data, three types of encryption are used: hardware, hardware-software, and software encryptions. The main differences between them, in addition to their cost and maintenance costs, are the encryption methods and the level of data security. In practice, these three criteria are decisive when choosing the type of encryption for users. Currently, the most affordable of them are considered to be software following by hardware-software, and the most expensive are hardware cryptographic information protection facilities (CIPFs). Although the cost of hardware CIPFs is significantly higher than the cost of the other CIPFs, the difference in monetary terms is incomparable with a significant increase in the level of data security. The advantage of hardware CIPFs is the guaranteed invariability of the algorithm itself, whereas, with software or hardware-software implementation, the encryption algorithm can be deliberately changed. Also, a hardware encryptor, through technical protection aimed at increasing the security of technological data, excludes any interference in the encryption process.

Hardware CIPFs have the ability to load encryption keys into the cryptoprocessor, bypassing the PC's RAM, while in software CIPFs the cryptographic keys are stored in the PC's RAM. Based on hardware CIPFs, it is possible to implement control and restriction of access to a PC. Also, hardware encryption takes the load off the PC's central processor [3-5]. Therefore, it is relevant to develop and implement an encryption algorithm in the form of a hardware-software complex for off-line encryption. Off-line encryption includes a procedure for preliminary cryptographic transformation of the transmitted information, after which the method of its transmission is determined. This method of ensuring the confidentiality of information is used in the case of an e-mail when information is encrypted in advance and then transmitted through established communication channels.

2 Materials and methods

2.1 Determination of basic requirements for cryptographic information protection facilities

The use of cryptographic systems to ensure data security is determined by technological capabilities, depending on the conditions and scope of the tasks being solved. This often leads to a revision (tightening) of requirements for cryptographic strength, flexibility, and performance of encryption, as well as cost acceptability for hardware implementation. New areas of information security, in particular the development of modern types of encryption in the USA (NIST competition), Europe (NESSIE competition), and other developed countries, show the recognition of the scientific and technological value and the growing role of encryption. When developing ciphers, the use of new cryptographic primitives is promising for technological applications. The cryptographic primitives traditionally used in the building symmetric block cryptographic systems are substitutions and permutations, arithmetic and algebraic transformations, and also other additional operations that well implement data diffusion.

2.1.1 Principles and stages of development of a symmetric block encryption algorithm

At present, along with asymmetric systems, symmetric block encryption algorithms are considered to be one of the main cryptographic means of ensuring the necessary level of secrecy when storing, processing, and transmitting information in modern information and communication systems. When developing this type of ciphers, the following general requirements are imposed, such as in [6-8]:

1. Providing the required level of cryptographic strength;

2. Simplicity, availability, and cost of software, hardware-software, or hardware implementation.

3. Providing high performance and flexibility in software, software-hardware, or hardware implementation;

However, the above requirements are rather controversial and contradictory, since increasing the cryptographic strength requires additional rounds of encryption, and this, as a rule, affects the decrease in the encryption speed. Nevertheless, international algorithms such as DES, AES [9], NESSIE [10-12], CryptRec [13], and others indicate the possibility of achieving the most suitable indicators for practical application.

In practice, the length of a message encrypted with a symmetric block algorithm is significantly larger than the length of the encryption key (the entropy of messages exceeds the entropy of the key). In this case, practical criteria of strength are considered, that is, the inadmissibility of the successful implementation of a crypto attack on algorithms in the conditions of modern computing potential (taking into account the upcoming optimistic development of computer technology and nanotechnology) in a certain period of time.

As known, the strength of an algorithm depends on the complexity of the implementation of a crypto attack on a symmetric block algorithm. Proceeding from this, as its indicators, as a rule, the following are used [6]:

1. Time complexity is the mathematical expectation of time (guaranteed security time) required to carry out a cryptographic attack using available and expected in the short term computing means;

2. Space complexity is the dependence of the amount of memory occupied on the size of the input data when performing cryptographic analysis of the algorithm;

3. The minimum number of encrypted and corresponding plaintexts (the minimum number of pairs) required to implement the attack.

The primary assessment of cryptographic strength is made in relation to known brute-force attacks, such as complete key enumeration, dictionary attacks, and others. Provided that the required level of strength to such types of attacks is ensured, an assessment of strength to analytical attacks and statistical methods of cryptanalysis is carried out.

For modern symmetric block encryption algorithms, it is proposed to apply the following conclusions as a criterion for evaluating cryptographic strength to analytical attacks [8]:

1. The total number of cipher/plaintext pairs required to perform a cryptanalysis attack exceeds the number of all possible cipher/plaintext pairs;

2. The complexity of a brute force attack should be less than the complexity of any other analytical attack.

To assess the complexity of an analytical attack according to the second criterion - the complexity of a brute-force attack - the following quantitative indicators are considered:

- Time criterion is the required minimum number of encryption operations (actions) for the implementation of an analytical attack (no fewer than with a complete key enumeration);

- The minimum memory size required to store intermediate and additional results during an analytical attack (not fewer than when implementing a dictionary attack on a full cipher).

A cipher is protected from a cryptanalysis attack if the above brute force attack indicators are lower than the cryptanalysis attack complexity.

Thus, when designing modern symmetric block encryption algorithms, the following basic requirements should be taken into account:

1. Cryptographic strength against brute-force attacks (by the minimum memory size required to store intermediate and additional results and by the time criterion);

2. Lack of ways to find and solve a system of algebraic (Boolean) equations describing the relationship between the plaintext, ciphertext, and encryption key;

3. Practical inaccessibility of implementing well-known analytical crypto attacks on the algorithm or their high computational complexity;

4. Availability of the optimal "margin of cryptographic strength" of the algorithm, taking into consideration the dynamics of innovative technological development of the industry;

5. The cryptographic strength of a simplified version of the algorithm, where some operations of the basic version were replaced or simplified by more elementary ones;

6. Statistical properties of the output ciphertext (cryptographic bitstream, cryptograms) should be close to the properties of a truly random sequence.

Considering the above, during the design and analysis of block symmetric algorithms, it is proposed to take into account the following approaches [6]:

- "Conservative design". This means that you should only use repeatedly verified, that is, reliable, designs, cryptographic primitives, and methods that provide guaranteed security;

- Strength against all known cryptanalysis attacks;

- Accessible and simple structure and design principles of the algorithm;

- Formation of the optimal "security margin" of the algorithm, the possibility of further secure use of the algorithm, given the emergence of potential cryptanalytic attacks and/or the development of computer technology;

- Protecting against all known vulnerabilities of the algorithm;

- The lack of a set of "bad" keys;

- The prevalence of strength over the performance of computer technology;

- Ensuring high performance close to the best world indicators.

When designing hardware-software and hardware CIPFs, it is also necessary to take into account the possibility of technical attacks based on changes in the temperature regime of the device, the appearance of ionizing radiation, measuring the consumed currents, side electromagnetic radiation, etc.

2.1.2 Design steps for block ciphers

The block cipher development process includes the following steps [8]:

1. Determination of the application area. At this step, the class of the cryptosystem is pre-selected, then the necessary list of requirements for its main initial parameters is drawn up.

2. Selection of key length. Currently, the actual key length is at least 128 bits, but in lightweight cryptography, a key with a length of 64 bits can be used, where timing is of particular interest.

3. Key generation scheme, consisting of a key installation system and a key management system. The key setting system includes algorithms and methods of generation, as well as a key verification rule. In turn, the key management system determines the further use, the order of transfer, storage, change, copying, and recovery, as well as the guaranteed destruction of keys.

4. Selection of basic cryptographic primitives and development of a cryptographic scheme. Clarification of the main approaches in the development of the algorithm, types and classes of symmetric block cryptosystems, the cost of their hardware, and hardware-software implementations. Optimal timing parameters are determined.

5. Assessment of technological resources for software, hardware-software, and hardware implementation of the encryption algorithm. Development work on encryptors and their software designs.

6. Estimation of the encryption speed. Various approaches are evaluated to obtain the required encryption speed for software, hardware-software, and hardware implementations. If the estimates derived at steps 5 and 6 do not correspond to the values obtained at step 1, then, taking into account the results obtained at the current step, it is proposed to return to step 4.

7. Evaluation of the cryptographic strength of the cipher using cryptanalysis and other methods.

8. Changing the algorithm with due account for the intermediate cryptanalysis of the cipher. Considering the results obtained at step 7, the main nodes of the cryptosystem are optimized to increase the efficiency of countering various types of crypto attacks. Until an acceptable degree of strength is obtained, this step can be repeated several times.

9. Conducting a basic, more detailed analysis of the cipher. If significant weaknesses are found, it is necessary to repeat step 8, if it is impossible to obtain a positive result, return to step 4.

10. At this step, the development of the encryption algorithm code or its implementation on the FPGA is carried out. The encrypted data is checked using statistical tests and special experiments to verify the completeness and reliability of the theoretical analysis. It is recommended to use well-known sets of statistical tests. All these tests are considered within the framework of mathematical statistics.

2.2 Requirements for symmetric block encryption algorithms developed for software and hardware implementation

The use of CIPFs is regulated by various normative and normative-methodological documents. The applied CIPFs should perform the following main functions:

- Key generation and key information management;
- Encryption of information in accordance with the standard ST RK 1073-2007;
- Identification and authentication of user connection and access to work (use of additional means of protection of CIPFs, such as tokens, iButton, etc.)

The developed symmetric block encryption algorithm, first of all, must comply with the standard ST RK 1073-2007 "Cryptographic information protection facilities. General technical requirements".

Given the experience gained during the previous R&D, it is advisable to use symmetric block ciphers previously developed in the information security laboratory, such as Qamal, EM, AL01, and others [14-17], which have been comprehensively studied and repeatedly tested against generally accepted requirements for symmetric block encryption algorithms.

The preliminary structure of the developed algorithm includes a variant of a substitution-permutation network (SP-network). This network uses an iterative transformation consisting of a substitution layer (nonlinear elements), a linear (mixing) layer, and a key adding layer. This design, due to the transformation of the entire data block at each iteration, provides a much faster mixing of the input vector in comparison with the Feistel network [18-19].

a) Building substitution boxes (nonlinear transformation nodes)

The use of pseudo-randomly generated lookup tables (S-boxes), selected according to the criteria of resistance to various cryptanalysis methods and the degree of nonlinearity of Boolean equations describing transformations, is considered relevant. These tables do not have an explicit mathematical structure inherent in known ciphers, allowing to reveal any algebraic dependencies between input, key, and output. This approach provides resistance to algebraic attacks.

To reduce the possibility of building and the probability of finding the correct solution to a system of algebraic equations, it is practiced to apply several substitutions at once.

When forming S-boxes of the developed cipher, the following basic requirements are imposed [5, 6]:

- Pseudo-random generation (ensuring a low probability of obtaining strict mathematical relationships between input and output data);

- Minimization of the maximum value of the probability of passing the difference through the substitution;

- Minimization of the maximum value of the probability of linear approximation of a substitution;

- Non-linear substitution order.

Next, consider the cryptographic primitives used in the design of the algorithm.

b) Linear transformation block

Currently, a comprehensively studied MDS transform is used as a linear transformation unit to obtain the best diffusion characteristics. It is recommended to use a 64-bit MDS code, which provides the best "avalanche effect" after the second round of encryption, which gives the best performance.

c) Subkey generation scheme

At this stage, it is necessary to exclude as much as possible the following disadvantages of known encryption algorithms:

- The ability to fully or partially recover the master key based on a known one or more subkeys;

- A fairly simple mathematical link connecting the subkeys. This flaw may allow for a "related-key attack";

- Acceptance of the master key itself as the first subkey;

- Using a design other than the round function in the generation circuit;

- Lack of the "avalanche effect" property on the subkeys, i.e. low effect of changes to the master key or subkey on other subkeys;

- differences in the computational complexity of generating subkeys for encryption and decryption.

Requirements key generating schemes of the developed algorithm:

- There should be a non-linear relationship between each bit of a subkey and the master key. In other words, there should be an "avalanche effect" when a change in one input bit of the master key should lead to changes in about 50

- High resistance - the ability to resist cryptanalysis methods and all known types of crypto attacks;

- Lack of a class of weak keys that make the round key generation scheme vulnerable to some attacks aimed at detecting any weaknesses in cryptographic properties;

- Recovery of a master key based on one or several subkeys should require high computational complexity;

- The simplicity of the software, hardware-software, and hardware implementation through the use of cyclic transformation;

- The computational complexity of generating subkeys should not exceed the computational complexity of the encryption algorithm itself;

- Generation of subkeys should be carried out in any direction of encryption.

Summing up, based on the results of the analysis of the requirements for encryption algorithms given in Section 1, for the designed encryption algorithm, the following conclusions and requirements can be formulated:

- The encryption algorithm should be symmetric and block based;
- The algorithm should provide a high level of cryptographic strength;
- The algorithm should be efficiently modified for all security levels and meet the relevant requirements as much as possible;
- The strength of the algorithm should not be based on the unavailability of the algorithm, i.e. the algorithm should be publicly available;
- The algorithm should provide for implementation on different platforms. It should be possible to effectively implement it in software on modern universal microprocessors and work on integrated microcontrollers and other small and medium-sized processors while maintaining the optimal ratio between size, cost, and performance;
- The algorithm should be easily adapted on specially designed encryption equipment, and its implementation in the form of a CIPF should be energy-efficient;
- The algorithm should be simple and easy to write program code to prevent errors that allow engineering analysis;
- The algorithm should use simple and low-resource operations that are effective on microcontrollers and microprocessors (XOR, addition, substitution boxes, cyclic shift);
- The algorithm should not have a set of "weak" keys, which facilitates its cryptanalysis, i.e. accept any random string of a certain length as the master key.

3 Results and discussions

The conditions and requirements for any symmetric block cipher algorithms are listed above. Let's take a look at a new encryption algorithm that has been designed with these requirements in mind.

The encryption algorithm AL03 is one of the modifications of the symmetric block encryption algorithm AL01 [8 - 9]. The developed algorithm uses blocks and keys with a length of 128 bits. The structure of the cipher is a variant of a substitution-permutation network (SP-network), and encryption is performed in R1=24 rounds. The encryption process includes key addition using the exclusive or (XOR) operation, substitution S-boxes, and the bitwise shift operation (Figure 1).

Each round of the encryption process consists of 3 transformations, called Step-1, Step-2, and Step-3, and ends with the addition of the round keys modulo 2 with the results obtained.

In the **Step-1** transformation, an input block of 16 bytes (128 bits) is divided into 4 subblocks equal to 4 bytes (32 bits): $a_0^0, a_1^0, a_2^0, a_3^0, a_0^1, a_1^1, a_2^1, a_3^1, a_0^2, a_1^2, a_2^2, a_3^2, a_0^3, a_1^3, a_2^3, a_3^3$ where the superscript represents the subblock number and the subscript represents the byte number in the subblock. The internal transformation of subblocks is performed as follows: 1st and 2nd bytes, 2nd and 3rd bytes, and 3rd and 4th bytes are summed modulo 2 and form new 1st, 2nd, and 3rd bytes, respectively. Further, the new 1st byte, after passing through the nonlinear transformation using the first S-box, is added modulo 2 to the 4th byte and produces a new 4th byte. The same transformation is performed for the remaining 3 subblocks, that is $b_i^j = a_i^j \oplus a_{i+1}^j, b_3^j = S_1(b_1{}^j) \oplus a_3^j, i = 0, 1, 2; j = 0, 1, 2, 3.$

Thus, all the results of the addition operation pass through the substitution S-box S_1.

$$c_i^j = S_1(b_i^j), i = 0, 1, 2, 3; j = 0, 1, 2, 3.$$

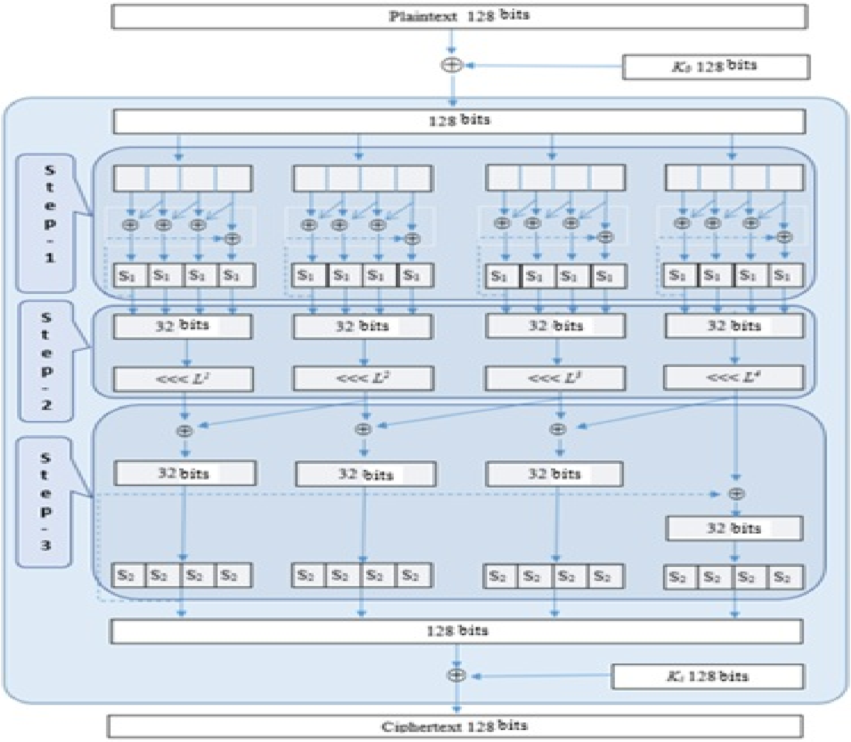

Figure 1: Scheme of the encryption algorithm AL03

Step-2. In each subblock, a concatenation operation is performed over its 4 bytes, followed by a left cyclic shift by a predetermined number of bits $L^j(L^j = 1, 3, 5, 3 \text{ bits})$ for each subblock, respectively:

$$c_0^j \parallel c_1^j \parallel c_2^j \parallel c_3^j = (c_0^j \parallel c_1^j \parallel c_2^j || c_3^j) \lll L^j, j = 0, 1, 2, 3.$$

The **Step-3** transformation is carried out similarly to **Step-1**, except that the operations are performed not on adjacent bytes, but on neighboring subblocks. The corresponding values of the 1st and 2nd subblocks, the 2nd and 3rd subblocks, and the 3rd and 4th subblocks are summed modulo 2 and give the new values of the 1st, 2nd, and 3rd subblocks. The new values of the 4th subblock are determined as follows: the newly obtained values of the 1st subblock after the second S-box transformation are summed modulo 2 with the corresponding values of the 4th subblock:

$$d_i^j = c_i^j \oplus c_i^{j+1}, d_i^3 = S_2(d_i^0) \oplus c_i^3, i = 0, 1, 2, 3; j = 0, 1, 2.$$

Thus, all the obtained values of the 4 subblocks pass through the substitution S-box S_2.

$$e_i^j = S_2(d_i^j), i = 0, 1, 2, 3; j = 0, 1, 2, 3.$$

Each i-th round of encryption ends by adding modulo 2 values e_i^j to the round key K_i.

$S_1(256) = \{98, 233, 16, 142, 0, 40, 127, 30, 25, 76, 169, 130, 19, 72, 58, 93, 77, 235, 148, 162, 196, 150, 38, 232, 82, 152, 177, 164, 211, 101, 188, 245, 165, 145, 115, 13, 121, 9, 234, 214, 180,$

117, 26, 138, 147, 247, 33, 189, 183, 179, 32, 255, 161, 2, 172, 83, 218, 167, 21, 95, 201, 199, 80,
28, 157, 96, 109, 60, 74, 190, 113, 137, 85, 205, 84, 143, 66, 62, 206, 146, 181, 55, 12, 59, 31, 91,
90, 24, 14, 191, 51, 159, 228, 65, 248, 244, 231, 135, 194, 129, 213, 114, 79, 111, 184, 102, 122,
207, 208, 163, 134, 128, 151, 100, 254, 227, 20, 29, 223, 118, 220, 176, 230, 197, 212, 48, 89, 222,
52, 202, 1, 106, 105, 175, 149, 224, 210, 39, 170, 68, 41, 61, 8, 168, 192, 5, 249, 182, 241, 54, 78,
45, 174, 215, 246, 99, 171, 6, 63, 140, 132, 4, 187, 160, 64, 186, 226, 86, 11, 110, 250, 112, 203, 67,
124, 216, 35, 57, 155, 119, 158, 166, 87, 73, 120, 75, 252, 103, 56, 217, 97, 47, 42, 251, 50, 221, 242,
240, 116, 88, 156, 34, 123, 141, 198, 18, 10, 44, 236, 193, 71, 239, 7, 125, 154, 53, 136, 23, 219, 229,
3, 238, 69, 107, 43, 185, 36, 108, 126, 92, 15, 253, 37, 46, 104, 237, 131, 81, 94, 139, 209, 225, 144,
49, 27, 200, 133, 70, 173, 17, 204, 22, 153, 243, 178, 195};

$S_2(256) = \{$168, 176, 8, 107, 204, 99, 10, 223, 243, 160, 118, 180, 146, 179, 230, 30, 59, 244, 212,
219, 105, 120, 92, 201, 163, 152, 193, 101, 36, 56, 26, 161, 44, 254, 166, 88, 83, 123, 178, 188, 84, 15,
31, 97, 190, 224, 48, 220, 29, 213, 129, 35, 76, 183, 124, 198, 17, 137, 229, 240, 78, 3, 67, 5, 109, 226,
132, 227, 222, 169, 234, 66, 95, 39, 214, 111, 86, 187, 32, 162, 194, 139, 81, 207, 141, 127, 40, 100,
126, 114, 46, 74, 153, 155, 47, 16, 211, 175, 196, 165, 182, 38, 140, 37, 98, 80, 149, 199, 75, 195, 209,
27, 251, 102, 202, 77, 237, 54, 242, 250, 60, 128, 121, 6, 131, 151, 115, 116, 0, 173, 112, 154, 117, 90,
231, 68, 113, 192, 63, 110, 12, 241, 20, 73, 119, 238, 216, 135, 13, 171, 94, 125, 156, 91, 189, 14, 69,
43, 138, 11, 133, 185, 148, 157, 1, 247, 58, 174, 158, 239, 130, 205, 57, 64, 235, 24, 21, 19, 61, 218, 51,
50, 104, 34, 22, 253, 72, 45, 164, 236, 108, 184, 206, 215, 53, 28, 143, 203, 23, 167, 2, 25, 79, 89, 9,
49, 134, 87, 225, 106, 150, 41, 62, 52, 70, 197, 255, 252, 18, 103, 144, 7, 217, 221, 71, 159, 181, 200,
249, 210, 245, 142, 145, 93, 147, 233, 172, 42, 208, 65, 177, 82, 33, 248, 136, 191, 186, 122, 232, 246,
96, 228, 170, 4, 55, 85}.

The algorithm for generating round keys. An algorithm for generating round keys from the master key $K(k_0, k_1, k_2, \ldots, k_{15})$ with a length of 16 bytes is considered. We will use the master key K as the round key K_0. The total number of round keys (excluding K_0) is the same as the number of rounds R_2. The values of the round key $K_0(k_0, k_1, k_2, \ldots, k_1 5)$ are written as a 4x4 square matrix A:

$$A = \begin{pmatrix} k_0 & k_1 & k_2 & k_3 \\ k_4 & k_2 & k_6 & k_7 \\ k_8 & k_9 & k_{10} & k_{11} \\ k_{12} & k_{13} & k_{14} & k_{15} \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix};$$

The algorithm for generating round keys, schematically shown in Figure 2, consists of **Stage-1**, **Stage-2**, and **Stage-3** transformations, which are described below. **Stage-1 transformation**

At this stage of the transformation, which consists of two steps, from a given matrix A, we will obtain a new matrix A of the same size.

Step 1. The intermediate values c_{ij} of the matrix A are determined from left to right, from top to bottom by adding modulo 2 all four elements of the ith row and three elements of the jth column, except for the c_{ij} itself.

Step 2. At this step, the obtained value c_{ij} passes through the substitution S-box and is written to the same place as the new value of the matrix A.

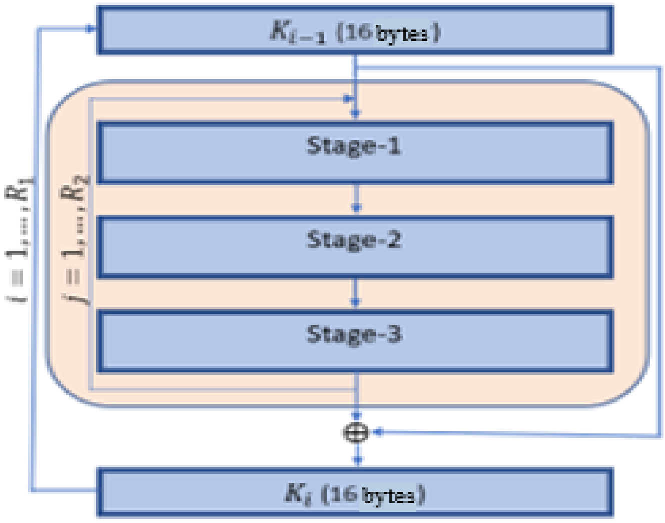

Figure 2: Scheme of the algorithm generating round keys

The Stage-1 transformation, consisting of the 1st and 2nd steps, can be written as:

$$m_{ij} = \oplus \sum_{k=0}^{3} a_{ik} \oplus (\oplus \sum_{k=0}^{3} a_{kj});$$

$$a_{ij} = S_1(m_{ij}); i = 0, .., 3; j = 0, .., 3;$$

where c_{ij} is the intermediate value of the matrix A, S is the substitution using the S-box S_1.

Stage-2 transformation

This transformation consists of only one operation. The elements of matrix A, obtained in Stage-1, are written in the form of a one-dimensional array $(a_{00}, a_{01}, a_{02}, a_{03}, a_{10}, a_{11}, a_{12}, a_{13}, a_{20}, a_{21}, a_{22}, a_{23}, a_{30}, a_{31}, a_{32}, a_{33})$. Further, all elements of the array are treated as bytes and are concatenated using the concatenation operator: $a_{00} \parallel a_{01} \parallel a_{02} \parallel a_{03} \parallel a_{10} \parallel a_{11} \parallel a_{12} \parallel a_{13} \parallel a_{20} \parallel a_{21} \parallel a_{22} \parallel a_{23} \parallel a_{30} \parallel a_{31} \parallel a_{32} \parallel a_{33}$.

Next, a cyclic left shift is performed by 1 bit. The obtained 16-byte result is taken as the new values of the 4x4 matrix A whose elements are located from left to right, from top to bottom.

Stage-3 transformation

This transformation is similar to the Stage-1 transformation. Here, too, the matrix Aundergoes a two-step transformation. The only difference is that the elements of the matrix are selected from right to left, from bottom to top. After transforming Stage-3, we get a new matrix A of the same size. We write this transformation, consisting of the 1st and 2nd steps, similar to the previous one:

$$m_{ij} = \oplus \sum_{k=0}^{3} a_{ik} \oplus (\oplus \sum_{k=0}^{3} a_{kj});$$

$$a_{ij} = S_2(m_{ij}); i = 0, .., 3; j = 0, .., 3;$$

where c_{ij} is the intermediate value of the matrix A, S is the substitution using the S-box S_2. Finally, the result of one round is the values obtained after the Stage-3 transformation. The Stage-1, Stage-2б and Stage-3 transformations are repeated $R_2 = 8$ times, and then the resulting 16-byte value is added to the previous round key $K_{i-1} modulo 2(XOR)$ and finally we get the next round key K_i, where $i = 1, \ldots, R_1$.

We have modified the AL01 symmetric encryption algorithm. The structure of the algorithm uses XOR operations and substitution S-boxes. In one round of encryption, only 56 operations are performed, therefore the complete algorithm is performed in 1,344 operations. Thus, with an effective software implementation, the encryption rate of this algorithm will be close to or even exceed the encryption rate of other known algorithms.

One of the tasks of the research work is to create a hardware prototype of the encryption algorithm developed in the LIS and adapted for this purpose. The creation of such a device is possible at the Special Design and Technology Bureau (SDTB) Granit LLP. The SDTB is engaged in the development and production of radar stations and single-board minicomputers. It has developed a device that is designed to emulate the operating mode of the CIPF. On its basis, SKTB Granit, within the framework of this project, plans to implement a software and hardware complex for preliminary encryption of information with the characteristics required for the project. The developed encryption algorithm will be implemented on a device that is a single-board minicomputer for CIPFs. Taking into account the peculiarities of the operation of the file encryption software, the device will be made in a truncated hardware configuration without connecting a keyboard and monitor.

4 Conclusion

The paper provides an overview of the basic requirements for modern CIPFs, studies the stages of developing a symmetric block encryption algorithm, systematizes the basic requirements for the developed cryptographic information security facilities, on the basis of which criteria for evaluating the developed cryptographic information protection systems are determined. The technical task for the creation of a software and hardware complex that implements the developed encryption algorithm has been clarified and agreed upon.

The research being carried out is new. The development and study of the encryption algorithm designed to ensure the protection of information, including classified information constituting a state secret, by the cryptographic transformation of data presented in the form of files, its implementation in the form of a hardware-software complex is aimed at the creation and development of domestic cryptographic means to ensure information security.

In addition, this article presents a new structure of the symmetric block cipher algorithm, its key generation algorithm that meets the basic requirements and recommendations of block cipher algorithms. Currently, work is underway to analyze the cryptographic strength of the algorithm using statistical and algebraic approaches.

5 Acknowledgments

The work was carried out within the framework of the grant funding program AP08856426 "Development and study of an encryption algorithm and creation of a software and hardware

complex for its implementation "of the Ministry of Education and Science of the Republic of Kazakhstan.

References

[1] Ivanov M.A., Chugunkov I.V.,"Teoriya, primenenie i ocenka kachestva generatorov psevdosluchajnyh posledovatelnostej [Theory, application, and quality assessment of pseudorandom sequence generators]" , Moscow, K-OBRAZ (2003), 136, [in Russian].

[2] Babenko L.K., Ischukova E.A.,"Sovremennye algoritmy blochnogo shifrovaniya i metody ih analiza [Modern block cipher algorithms and methods for their analysis]" , Moscow, Helios ARV (2006), p. 376, [in Russian].

[3] Schneier B., "Applied Cryptography,: Protocols, Algorithms, and Source Code in C". 2-nd ed.; John Wiley & Sons, Inc. (1996): 118.

[4] Ivanov M.A., "Kriptograficheskie metody zashchity informacii v komp'yuternyh sistemah i setyah [Cryptographic methods of information security in computer systems and networks]" , Moscow, K-OBRAZ (2001), 368, [in Russian].

[5] Gorbenko I. D., Dolgov V., Oleynikov R. V., Ruzhentsev V. I., Mikhaylenko, M. S., Gorbenko, Y. I.,"Razrabotka trebovanij i princip proektirovaniya perspektivnogo simmetrichnogo blochnogo algoritma shifrovaniya [Development of requirements and design principle perspective symmetrical block encryption algorithm]" , Izvestiya yuzhnogo federal'nogo universiteta. Tekhnicheskie nauki no. 1, V. 76 (2007), 183-189, [in Russian].

[6] Gorbenko I. D., Dolgov V., Oleynikov R. V., Ruzhentsev V. I., Mikhaylenko, M. S., Gorbenko, Y. I.,"Razrabotka trebovaniy i printsip proektirovaniya perspektivnogo simmetrichnogo blochnogo algoritma shifrovaniya [Development of requirements and design principle perspective symmetrical block encryption algorithm] Izvestiya YUFU. Tekhnicheskie nauki. no. 1. URL: https://cyberleninka.ru/article/n/razrabotka-trebovaniy-i-printsip- proektirovaniya- perspektivnogo-simmetrichnogo-blochnogo-algoritma-shifrovaniya (2007), (3.11.2020), [in Russian].

[7] Apparatnoe shifrovanie dlya PK [Hardware encryption for PC].Press center Company Active, 2013. URL: https://www.aktiv-company.ru/press-center/publication/2003-04-10.html (23.11.2020), [in Russian].

[8] Znaenko N.S., Kapitanchuk V.V., Petrishchev I.O., Shubovich V.G.,"Nekotorye kriterii ocenki kachestva algoritmov shifrovaniya [Some criteria for evaluating the quality of encryption algorithms]" , NovaInfo.Ru. Tekhnicheskie nauki no. 59 (2017) URL: https://novainfo.ru/article/11211 (23.11.2020), [in Russian].

[9] AES discussion forum:http://aes.nist.gov.

[10] New European Schemes for Signatures, Integrity, and Encryption NESSIE:URL: http://cryptonessie.org.

[11] Final report of European project number IST-1999-12324, named New European Schemes for Signatures, Integrity, and Encryption. Springer-Verlag, Berlin Heidelberg NewYork, etc. (2004).

[12] NESSIE public report D20.NESSIE Security Report. URL: http://cryptonessie.org.

[13] URL: http://cryptrec.org/ Cryptography Research and Evaluation Committees.

[14] Report on research work "Development of software and firmware for cryptographic protection of information during its transmission and storage in info-communication systems and general-purpose networks 2018, State registration no. 0118PK01064..

[15] Biyashev R.G., Smolarsh A., Algazy K.T., Khompysh A., "Encryption algorithm "QAMAL NPNS"using non-positional polynomial notations Journal of Mathematics, Mechanics, and Computer Science, Bulletin of KazNU no. 1 (105), Almaty (2020), 198-207. .

[16] Nursulu Kapalova, Ardabek Khompysh, Muslum Arici, Kunbolat Algazy., A block encryption algorithm based on exponentiation transform // Cogent Engineering. - 2020. - No. 7 (1788292). - P. 1-12 // https://doi.org/10.1080/23311916.2020.1788292.

[17] Kapalova N.A., Haumen A.,"Simmetrichnyj blochnyj algoritm shifrovaniya dannyh "VS-2"[BC-2 symmetric block algorithm for data encryption]" , Bezopasnye informacionnye tekhnologii". Sbornik trudov Desyatoj mezhdunarodnoj nauch-no-tekhnicheskoj konferencii, Moscow, Bauman MSTU, 2019, 161-166, [in Russian].

[18] Jonathan K., Yehuda L.,"Introduction to Modern Cryptography CRC PRESS, London-New York- Washington, (2007), 160.

[19] Panasenko S.P., "Algoritmy shifrovaniya [Encryption algorithms]", Special reference book, Saint Petersburg, BHV-Petersburg (2009), 576, [in Russian].

Список литературы

[1] Иванов М. А., Чугунков И. В.,Теория, применение и оценка качества генераторов псевдослучайных последовательностей. - М.: К-ОБРАЗ. - 2003. - 136 с.

[2] Бабенко Л.К., Ищукова Е.А., Современные алгоритмы блочного шифрования и методы их анализа. - М.: Гелиос АРВ, 2006. - 376 с.

[3] Schneier B., "Applied Cryptography.: Protocols, Algorithms, and Source Code in C". 2-nd ed.; John Wiley & Sons, Inc.(1996): 118.

[4] Иванов М.А., Криптографические методы защиты информации в компьютерных системах и сетях. - М.:К-ОБРАЗ, 2001. - 368 с.

[5] Горбенко И.Д., Долгов В., Олейников Р.В., Руженцев В.И., Михайленко М.С., Горбенко Ю.И., Разработка требований и принцип проектирования перспективного симметричного блочного алгоритма шифрования // Известия южного федерального университета. Технические науки. - 2007. - Т. 76, № 1. - С. 183-189.

[6] Горбенко И.Д., Долгов И.В., Олейников Р.В., Руженцев В.И., Михайленко М.С., Горбенко Ю.И., Разработка требований и принцип проектирования перспективного симметричного блочного алгоритма шифрования // Известия ЮФУ. Технические науки. 2007. №1. URL: https://cyberleninka.ru/article/n/razrabotka-trebovaniy-i-printsip-proektirovaniya-perspektivnogo-simmetrichnogo-blochnogo-algoritma-shifrovaniya (3.11.2020).

[7] Аппаратное шифрование для ПК // Пресс-центр "Компания "Актив". 2013. URL: https://www.aktiv-company.ru/press-center/publication/2003-04-10.html (23.11.2020).

[8] Знаенко Н.С., Капитанчук В.В., Петрищев И.О., Шубович В.Г., Некоторые критерии оценки качества алгоритмов шифрования // NovaInfo.Ru. Технические науки. 2017. №59. URL: https://novainfo.ru/article/11211 (23.11.2020).

[9] AES discussion forum: http://aes.nist.gov.

[10] New European Schemes for Signatures, Integrity, and Encryption NESSIE: URL: http://cryptonessie.org.

[11] Final report of European project number IST-1999-12324, named New European Schemes for Signatures, Integrity, and Encryption. Springer-Verlag, Berlin Heidelberg NewYork, etc. 2004.

[12] NESSIE public report D20. NESSIE Security Report. URL: http://cryptonessie.org.

[13] URL: http://cryptrec.org/ Cryptography Research and Evaluation Committees.

[14] Report on research work "Development of software and firmware for cryptographic protection of information during its transmission and storage in info-communication systems and general-purpose networks 2018, State registration no. 0118PK01064.

[15] Biyashev R.G., Smolarsh A., Algazy K.T., Khompysh A., "Encryption algorithm "QAMAL NPNS"using non-positional polynomial notations Journal of Mathematics, Mechanics, and Computer Science, Bulletin of KazNU no. 1 (105), Almaty (2020), pp. 198-207.

[16] Nursulu Kapalova, Ardabek Khompysh, Muslum Arici, Kunbolat Algazy., A block encryption algorithm based on exponentiation transform // Cogent Engineering. - 2020. - No. 7 (1788292). - P. 1-12 // https://doi.org/10.1080/23311916.2020.1788292.

[17] Капалова Н.А., Хаумен А., Симметричный блочный алгоритм шифрования данных "BC-2"// "Безопасные информационные технологии". Сборник трудов Десятой международной научно-технической конференции - М.: МГТУ им. Н.Э. Баумана, 2019. - С. 161-166.

[18] Jonathan K., Yehuda L., "Introduction to Modern Cryptography CRC PRESS, London-New York- Washington, (2007), 160.

[19] Панасенко С.П., Алгоритмы шифрования. Специальный справочник. - СПб.: БХВ-Петербург, 2009. - 576 с.