Atlantis Highlights in Computer Sciences, volume 3

21st International Scientific Workshop on Computer Science and Information Technologies (CSIT 2019)

Development of an Encryption Method Based on Cyclic Codes

Vyacheslav Petrenko
Academic department of Organization and Technology of Information Protection
North-Caucasus Federal University
Stavropol, Russia
vip.petrenko@gmail.ru

Sergey Ryabtsev
Academic department of Applied Mathematics and Computer Security
North-Caucasus Federal University
Stavropol, Russia
nalfartorn@yandex.ru

Andrey Pavlov
Academic department of Applied Mathematics and Computer Security
North-Caucasus Federal University
Stavropol, Russia
losde5530@gmail.com

Artem Apurin
Academic department of Organization and Technology of Information Protection
North-Caucasus Federal University
Stavropol, Russia
apurin.a@icloud.com

***Abstract*—The constant increase in the number of new types of cyber threats actualizes the issues of their information transfer. This article (report) proposes a research on the development of an encryption method based on cyclic codes. The purpose of this article is to develop an open algorithm, even knowing which an attacker will not be able to obtain the source text. The developed software encryption module will solve the problem of protecting information for small companies and private users, and also saves a large amount of resources, since one module solves the problem of ensuring the confidentiality and noise stability of transmitted messages. Also, this module is intended for common operating systems. This module is also intended for common operating systems, and user work is carried out using a friendly user interface. The software module performs RSA (Rivest–Shamir–Adleman) asymmetric encryption and jam resistant coding using the BCH (Bose-Chaudhuri-Hocquenghem) code. It is provided an example of the work of the program module developed on the basis of the pro-posed method. Based on the collection and analysis of statistical data output software system, a conclusion about the efficiency of the proposed solution and comparison with analogues has been drawn.**

Keywords—cyclic code, linear block code, RSA algorithm, noise resistant coding, encryption module, error correction module.

I. INTRODUCTION

The development of information and telecommunication technologies in the modern world has reached the level that they have penetrated into all areas of activity. In this regard, the requirements for information security are constantly growing. In many industries, a few erroneous bits or loss of privacy can cause huge production losses. In such conditions, protection should be carried out both from unauthorized access, and from errors during message transmission. In the society where information is the main resource, ways of its protection should be available for everyone. The necessity of fixing bugs and information security is extremely great at present time, both for individuals and for organizations and enterprises of various forms of ownership. For such organizations and enterprises it would be convenient that one module carries out the task of confidentiality and noise stability. But the implementation of encryption and noise resistant codes in information protection systems requires resources that are not always available to small organizations. In this regard, all the requirements for the method are put forward in a way that it can be used by average people and small companies that do not have a security department, but need protection. With the increase in the value of information, such tools should be available to everyone, hence the implemented module must satisfy the requirements:

1. Availability, i.e. opportunity to get the required information service for an acceptable time and cost. This paper discusses the average people and small private companies. They do not have: a closed communication channel, huge computing power. In organizations of medium and small size, significantly more personal data losses were recorded than in large companies.

2. Ease of use, with regard to lack of specific computer skills. Also, the user does not need to see all the elements of the program, but only the necessary ones, this means that it is necessary to use an object-oriented programming language in which the principle of encapsulation is performed. These requirements mean that this module should primarily be intended for a common operating system with a friendly user interface.

3. Openness, i.e. the encryption algorithm should be not secret, but difficult for an attacker who even knows the algorithm to get the source text. Since the secret algorithm is not available to average users, it should be focused on the encryption of words, in particular, the audience in question needs protection of a small amount of personal data transmitted over the network.

4. Asymmetric encryption is the third condition. The most common algorithm is RSA, unfortunately, it contradicts the resource cost requirement. Therefore, we will use numbers with a smaller order, this lowers the cryptographic strength, but protection is more likely from accidental hacks than from specially organized attacks, so this condition can be considered as acceptable.

5. To provide not only encryption, but also noise stability of information. Transmission channels often do not have any means of insurance of noise stability for the average user and it is very convenient that one module would perform many functions, it is necessary to use cyclic codes for this task.

After analyzing the requirements put forward, it is possible to describe the means by which the proposed encryption method will be created: it is necessary to develop an RSA software encryption module for its usage in Windows OS, with the simplest interface and a small key length, to reduce performance costs, it is also necessary to use noise resistant coding of the sent message. To solve the problem, it is necessary to solve the following subtasks:

1. Develop an RSA software encryption module,
2. Develop a cyclic code correction scheme,
3. Combine them into one module.

It's necessary to analyze the literature on the research topic to solve the tasks. The variety of literature on the subject of the research is represented by numerous developments in this field and is due to the relevance of the usage of these developments in the modern world.

In the paper [1] two new fast and effective decoding algorithms to decode linear block codes on binary channels are presented. The main idea in the first decoder is based on a new effective hash function that permits to find the error pattern directly from the syndrome of the received word. The main disadvantage of the first decoder is the spatial complexity, because it requires to previously storing all corrigible error patterns in memory. For reminding this problem, a second decoder based also on hash is proposed but it requires storing only the weight of each corrigible error pattern instead of the error pattern itself.

In the paper [2] cyclic codes of odd length *n* over the local, non-chain ring $R = Z\,2\,s\,[u]\,/ < uk > = Z\,2\,s + uZ\,2\,s + \ldots + uk - 1\,Z\,2\,s\,(uk = 0)$ are considered, for any integers $s \geq 1$ and $k \geq 2$. This algebraic structure is then used to establish the duals of all cyclic codes. Among others, all self-dual cyclic codes of odd length *n* over the ring *R* are determined. Moreover, some examples producing several optimal codes are provided.

This article [3] proposes a pair of symbol codes to protect the pair in the reading channel of the pair of symbols from error. One of the main tasks in symbol-pair coding theory is to determine the minimum pair-distance of symbol-pair codes. In this paper, the symbol-pair distance of cyclic codes of length p e over Fpm is investigated. The exact symbol-pair distance of all cyclic codes of such length is determined.In this article [4] additive cyclic codes over Galois rings but in a more general ring family finite commutative chain rings are investigated. When focusing on non-Galois finite commutative chain rings, it is necessary to observe two different kinds of additivity. One of them is a natural generalization of the study in Caoetal. (2015), where as the other one has some unusual properties especially while constructing dual codes.The article [5] considers cyclic codes with a repeated root, the block length of which is divided by the characteristic of the underlying field. Cyclic self dual codes are also the repeated root cyclic codes. It is known about the one-level squaring construction for binary repeated root cyclic codes. In this correspondence, the two level square construction for binary repeated root cyclic codes of length 2 a bare introduced, a > 0, where b is odd.In this article [6] it is presented a new class of cyclic codes constructed as the direct sum of two cyclic codes with one weight. As it is shown, this new class of cyclic codes is in accordance with the previous conjecture, since its codes have exactly six nonzero weights.

In this article [7] an effective racetrack memory based in-memory design is proposed to accelerate the modular multiplication for asymmetric cryptography algorithms. A new two-stage scalable modular multiplication algorithm is proposed to significantly improve the delay. An efficient architecture is further developed to reduce the number of required adders by half.In this article [8] asymmetric algorithm based on Chinese Remainder Theorem and double sequence, which uses the sequence of random numbers generated from the interference of Logistic and Chebychev chaotic mapping to interfere with the backpack sequence, while setting the easy solutions of super-increasing knapsack problem as the limitation of the algorithm, and using Chinese remainder theorem to hide the sequence mentioned above, before making the hidden backpack sequence to be transformed modulus has been designed.

In this article [9] it is introduced the basic number theories of RSA cryptosystem applied t to key algorithm of RSA cryptosystem, such as Euclidean and its extension theorem, square-multiply algorithm and prime number test-ing. A description of Matlab simulation of key algorithm and RSA encryption and decryption are given. In this article [10] a new approach for DES (Data Encryption Standard) based on public key cryptography called Asymmetric-DES is proposed. The hybrid of DES and RSA algorithms are combined to be secure for only two rounds, when the symmetric DES is protected for sixteen rounds with different combinations of the key. In this article [11] a version of RSA encryption that uses the Chinese Remainder Teorem (CRT) for the purpose of concealing multiple plain-texts in one cipher-text is proposed. Such a scheme allows the sender to possibly send different information to multiple receivers, and each receiver is only able to decrypt the message intended for it. The new algorithm can also take advantage of current methods that speed up the decryption process of RSA.

II. METHODS

a. Development of the software encryption module

The synthesis of the RSA algorithm shown in Figure 1 can be divided into the stages: key generation, encryption, decryption. Key generation works as follows.

It is necessary to select two prime numbers *p* and *q* of a given size. When choosing such numbers it is necessary to take into account some factors, since crypto resistance is directly dependent on them. Generation of random numbers is suitable for research. It is possible to use the algorithm: Sundaram sieve. It serves to find all primes up to a given number, it is necessary to represent odd natural numbers as $2m + 1$, where m – a natural number.

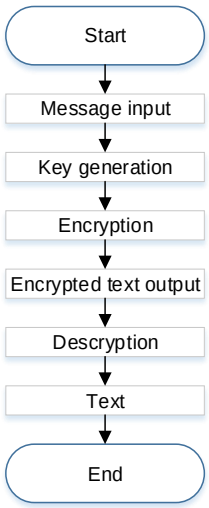

Fig. 1. Encryption algorithm

If the number *2m+1* is composite, it is represented as:

$$2m + 1 = (2i + 1)(2i + 1), \quad (1)$$

Where i and j-natural numbers, which is also equivalent to the ratio:

$$m = 2ij + i + j. \quad (2)$$

If all numbers of the form(2) are excluded, from the series of natural numbers, for each of the remaining numbers m the number $2m + 1$ must be simple. And vice versa, if the number $2m + 1$ is simple, then the number m cannot be represented in the form(2) and, thus, $m$will not be excluded in the process of the algorithm.

Calculate the mod p and q, n $= pq$.

Euler function is calculated according to the formula (3)

$$\varphi(n) = (p - 1)(q - 1), \quad (3)$$

Where $\varphi(n)$– Euler function of the number n.

The exponent d, is randomly selected so that it is mutually simple with the Euler function. The Euclidean algorithm is used for calculation mutual simplicity. Calculate the decryption keye, so that it meets the condition:

$$de \equiv 1(mod\varphi(n)). \quad (4)$$

The calculation of encrypted text is shown in Figure 2.

Meanwhile e and n must also be mutually prime numbers.The first thing to do is to select the text to encrypt and enterit. It is also necessary to obtain an ASCII (American standard code for information interchange) code number for each symbol. This is necessary because further encryption of the text will be carried out not with the symbol, but with its number in the ASCII code table.

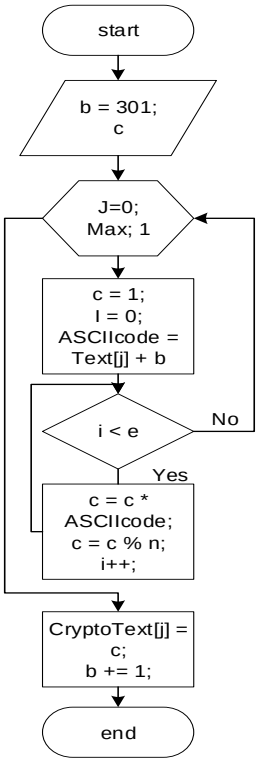

Fig. 2. Message encryption unit

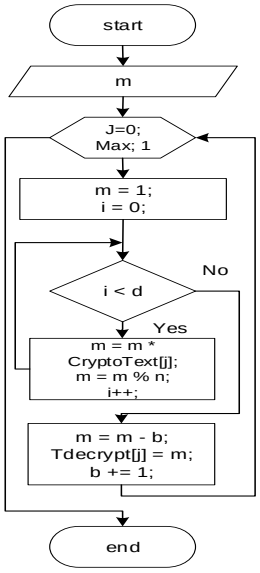

Fig. 3. Message decryption unit

The calculation of the encrypted text is performed according to the formula:

$$c = D = c^d mod n. \quad (5)$$

Decryption: it is necessary to calculate the original message, decryption of each message is shown in Figure 3 and is carried out according to the formula:

$$c = E = m^e mod n \quad (6)$$

Thus, the RSA encryption algorithm has been synthesized.

b. Development of the cyclic error correction code

At the next stage of the development it is necessary to create a correction cyclic code. Since the RSA algorithm uses the ASCII table, its dimension is 255 symbols. Therefore, it is necessary to synthesize a control circuit for an information word not less than $2^i \geq 256,\ i \geq 8$. The data is transmitted in small volumes, therefore, for effective control there will be enough code capable for correcting 1 error and detecting the presence of a double one. To detect a double one it is necessary to have another extra digit. Thus, it is necessary to synthesize a self-correcting code for an information word with a length of 9 bits. Its algorithm is shown in the figure 4.

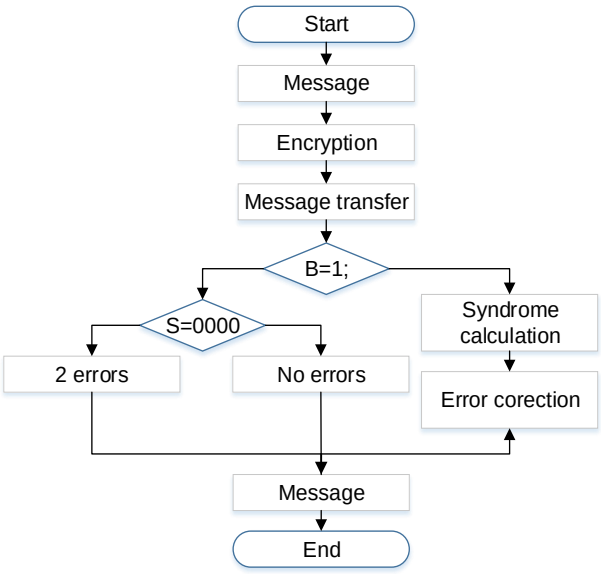

Fig. 4. Error correction algorithm

BCH codes is a large class of cyclic codes used for noise resistant coding, which are a generalization of the Hamming code with the possibility of correcting multiple errors [12]. The most suitable of the BCH codes is the BCH code (15,11,1).

Calculate and consider its parameters:

1. The generating polynomial of such a code is equal to $\text{tog(x)} = \text{x}^4 + \text{x} + 1$.

2. Block length is $15 = 2^4 - 1.\oplus$

3. Number of check bits: $15 - 11 = 4 * 1$.

4. The minimum code distance must satisfy the condition: $d_{min} \geq 2 * t + 1$.Since it is necessary to correct 1 error, it is permissible $d_{min} = 2 * 1 + 1$, where t is the number of corrected errors.

But for the implementation of the proposed module, 9 is enough, and the maximum number in the binary representation will be 9. Therefore it is necessary to use a 2-shortening of the code.

2-shortening of the code (crossing out information symbols) is as follows: Any code can be 2-shortened by deleting information symbols, and the minimum distance of a 2-shortened code will not exceed the minimum distance of the source code, if not to ensure of a special choice of crossed-out symbols. Practically all cyclic codes, as a rule, are 2-shortened by crossing out consecutive positions, since this simplifies their implementation [3]. In this case, the minimum code distance will be greater than$d_{\min}$ ofthe source code. It is necessary to cross out two consecutive positions in the higher digits, since for coding $2^8 = 256$, we do not need them. Thus, the block length will be 13 symbols, the number of control bits $r = 13 - 9 = 4 * 1$,and the minimum code distance satisfies the condition $d_{min} \geq 2t + 1$.

Let I – the information word, and R – the control digits. First, it is necessary to get a symbol to match the number in the ASCII table and present it in binary form. In object-oriented programming languages, this is easy to do if you return the integer value of a variable. For binary representation, it is necessary to enter variables $i1, i2, i3 \ldots, i9$.

Determine the number of control digits and parity groups. There should be so many control digits so that their number would satisfy the expression $2^r > 9$. Thus, for an information word of length 9 symbols it will be 4 control digits: $2^4 = 16$. That means that the coded information word, protected from one error, will have 13 digits, 9 of them are information one and 4 - control ones.

Give each of the digits its own number - from 1 to $i + r$. Since the condition $2^r > i + r$, is satisfied, each number can be represented by an r-digit binary number. Parity control groups are formed as follows. Each digit of a code word is included in so many groups, as many units in its binary code. Each *i*-th control group includes those digits in which there is one in the *i*-th position. The least significant digit is the control one. The control digits will be those ones that have only one unit in the binary record. Thus, the control will be the digits numbered: 1,2,4,8.

The operation based on 2 module is used, to calculate the control groups:

$$r_1 = i_1 \oplus i_2 \oplus i_4 \oplus i_5 \oplus i_7 \oplus i_9;$$
$$r_2 = i_1 \oplus i_3 \oplus i_4 \oplus i_6 \oplus i_7;$$
$$r_3 = i_2 \oplus i_3 \oplus i_4 \oplus i_8 \oplus i_9;$$
$$r_4 = i_5 \oplus i_6 \oplus i_7 \oplus i_8 \oplus i_9;$$

After that, the code word is transmitted:

$$I = (i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9, r_1, i_2, i_3, i_4).$$

The word received in the receiver is denoted by:

$$I' = (i'_1, i'_2, i'_3, i'_4, i'_5, i'_6, i'_7, i'_8, i'_9, r'_1, r'_2, r'_3, r'_4).$$

After receiving the wordI' a general parity check is performed. That means the received word is compared with the transmitted one. If it gives 0, then there are either no errors, or there is a double error. This requires verification.

The calculation of the error syndrome:

$$B = I \oplus I_1;$$
$$S_1 = i_1 \oplus i_2 \oplus i_4 \oplus i_5 \oplus i_6 \oplus i_8 \oplus r_1 \oplus B;$$
$$S_2 = i_1 \oplus i_3 \oplus i_4 \oplus i_6 \oplus i_7 \oplus r_2 \oplus B;$$
$$S_3 = i_2 \oplus i_3 \oplus i_4 \oplus i_8 \oplus i_9 \oplus r_3 \oplus B;$$
$$S_4 = i_5 \oplus i_6 \oplus i_7 \oplus i_8 \oplus i_9 \oplus r_4 \oplus B;$$

If it is zero, then there are no errors. If it is non-zero, then a double error occurs and this message cannot be used because the information is distorted. If the total check is odd, then there is a single error, which place is determined by 4 checks on the syndromes.

Error syndrome $S = (S_1, S_2, S_3, S_4)$ allows to find out the digit in which the error occurred. And correct the error by performing the inverse of the variable in the specified digit.

Check the working capacity of the program, manual calculation of error correction is performed and compared with the work of the program. Let the source word is $I = 511$; In binary representation $I = 111111111$. Denote r – the control digit; a i – the information digit, where i = 1, 2, 3, 4…, 9.

Calculate r_1, r_2, r_3, r_4:

$$r_1 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 0;$$
$$r_2 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 1;$$
$$r_3 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 1; \quad r_4 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 1.$$

Simulate the occurrence of an error in the i_1 digit. B – parity check.

$$r4' = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 1, \quad r3' = 1 \oplus 1 \oplus 1 \oplus 1 = 0,$$
$$r2' = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 1, \quad r1' = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 1.$$

Calculate error syndromes:

$$S4 = i1 + i2 + i4 + i5 + i6 + i8 + r1' = 1,$$
$$S3 = D1 + D3 + D4 + D6 + D7 + P2' = 1,$$
$$S2 = D2 + D3 + D4 + D8 + D9 + P3' = 0,$$
$$S1 = D5 + D6 + D7 + D8 + D9 + P4' = 0,$$
$$S = (S1, S2, S3, S4) = 0011.$$

In our case, the syndrome is 0011, therefore, an error is in the discharge i_1, n accordance with this, the decoder sends a signal to the output, and the inverter corrects the error. This scheme can be implemented in hardware. The word modeling is used with keys, the module two addition operation implements the XoR, element, light bulbs were used as indicators of tracking the result, and 74 series of chips were used as a decoder: 7451 to determine the syndrome and 7439 to correct errors.

c. Testing the encryption module using cyclic codes

Now it is necessary to combine the encryption algorithm with the cyclic shortened BCH code. It is necessary to generate keys, to add a correction cyclic code after receiving an ASCII code, to encrypt the message. Check errors and decrypt the received message, according to the result of the check. So it is necessary to add a corrective error code. If a single error is made during the transmission of the encrypted text, it will correct it, if there is a double error, the program will inform that this message cannot be used, if there are no errors, then the result of the work will not differ from the result of the encryption module without error correction.

The program receives a number of the letter in ASCII, after that a binary representation of this number is received, and control groups and error syndromes are calculated. The encoded text must be decoded upon receipt by the receiver.

Thus, the module performs not only data encryption, but also protects the encrypted text from errors when sending a message based on the cyclic code

III. RESULTS

Based on the proposed method, the software module has been developed. In Listing 1, an example of work with two errors in the transmission of the word «Hello»: is presented: in the digits 1 and 2, the total parity check is 0, this means that there are no errors or two, then the syndrome is calculated, because the syndrome is not zero, then there are 2 errors.

Listing 1 - Double error test

```
Key generation….
<2251,    2537>   -    Open key
Type the text: hello
Information word:001101000 1011
Accepted word   :111101000 1011
B=0
S=1000
2 errors detected:
Information word:000000000 0000
Accepted word   :110000000 0000
B=0
S=0110
2 errors detected :
```

Consider the result of the work of the program code in Listing 2 in case when there are no errors. Let the message be sent: «HelloPetr pas_4644».

Listing 2 - Test of the program code without errors during message transmission

```
Key generation….
<2096,    6059>   -    Open key
Type the text
Hello Petr pas_4464
№ symbol=No error

Text ASCII Encrypted text ASCII  Decrypted text
 H      72          3512       72          H
 e     101          1010      101          e
 l     108          1487      108          l
 l     108          5480      108          l
 o     111          4151      111          o
        32          3020       32
 P      80          2200       80          P
 e     101          2375      101          e
 t     116          5933      116          t
 r     114          3241      114          r
        32           282       32
 p     112          3241      112          p
 a      97          2617       97          a
 s     115          2254      115          s
 _      95          2617       95          _
 4      52          4012       52          4
 4      52          1930       52          4
 6      54          4264       54          6
 4      52          5164       52          4
                     595        0
Press any key to continue . . .
```

In Listing 3, an example of transmitting of a symbol g with an error is considered; the error in the discharge will be manually modeled: i_1, consider 1 symbol, since modeling errors in the text is a laborious process and the output data of a large sample will not be visual for presentation. When

receiving a message, the total parity check was equal to one, that means that the word has been transmitted with one error, then the error syndrome is calculated, the symbol is corrected in the discharge.

Listing 3 - Single error test

```
Key generation….
<2251,    2537>   -    Open key
Type the text
g
Information word:001100111 1011
Accepted word   :101100111 1011
B=1
S=0011
Discharge error i1:Corrected word:001100111 1011
Information word:000000000 0000
Accepted word   :100000000 0000
B=1
S=0011
Discharge error i1:Corrected word:000000000 0000

Text ASCII Encrypted text ASCII  Decrypted text
_____________________________________________
g     103       1176       103          g
        0       2151         0
Press any key to continue
```

The software module performs the RSA asymmetric encryption and noise resistant coding using the BCH code, no operation errors were detected, thus the encrypted message is protected from single errors, we have a correctly decoded message at the output. Three cases have been tested: no errors, double error, single error. In all cases, the module works correctly.

IV. DISCUSSION

The developed software encryption module will solve the problem of protecting information for small companies and private users, as well as save a large amount of resources, since one module solves the problem of confidentiality and noise stability. After analyzing the results of the program, it can be concluded that a small encryption exponent should be used to improve the encryption module, which will increase the encryption speed using RSA. However, if the number of encryption exponents is small enough, then there is a risk that there will be *e* subscribers with the same value of encryption exponents. It is necessary for each user to use a separate encryption key value. The higher the digit capacity of the open exponent, the harder it is to crack the algorithm.

V. CONCLUSION

The software encryption module with a cyclic BCH code has been developed on the basis of the proposed method. RSA encryption algorithm, and error correction code have been synthesized. Thus, the module allows not only to encrypt a message, but also to protect it from distortion when sending a message. Testing is described and examples of the module working capacity are given for all possible cases when sending messages.

The software module performs RSA asymmetric encryption and jam resistant coding using the BCH code, no operation errors have been detected, thus the encrypted message is protected from single errors, there is a correctly decoded message at the output.

REFERENCES

[1] El KasmiAlaoui, M.S., Nouh, S., Marzak, A. Two New Fast and Efficient Hard Decision Decoders Based on Hash Techniques for Real Time Communication Systems (2019) Advances in Intelligent Systems and Computing, 756, pp. 448-459. DOI: 10.1007/978-3-319-91337-7_40

[2] Dinh, H.Q., Singh, A.K., Kumar, P., Sriboonchitta, S. On the structure of cyclic codes over the ring Z2s[u]/〈uk〉 (2018) Discrete Mathematics, 341 (8), pp. 2243-2275. DOI: 10.1016/j.disc.2018.04.028

[3] Sun, Z., Zhu, S., Wang, L. The symbol-pair distance distribution of a class of repeated-root cyclic codes over Fpm (2018) Cryptography and Communications, 10 (4), pp. 643-653. Цитирован(ы) 1 раз. DOI: 10.1007/s12095-017-0249-2

[4] Martinez-Moro, E., Otal, K., Özbudak, F. Additive cyclic codes over finite commutative chain rings (2018) Discrete Mathematics, 341 (7), pp. 1873-1884. DOI: 10.1016/j.disc.2018.03.016

[5] Vinocha, O.P., Bhullar, J.S., Gupta, M. Squaring construction for repeated-root cyclic codes (2010) World Academy of Science, Engineering and Technology, 65, pp. 1002-1004.

[6] Vega, G. A family of six-weight reducible cyclic codes and their weight distribution (2015) Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9084, pp. 184-196. DOI: 10.1007/978-3-319-18681-8_15

[7] Luo, T., He, B., Zhang, W., Maskell, D.L. A novel

[8] two-stage modular multiplier based on racetrack memory for asymmetric cryptography (2017) IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD, 2017-November, pp. 276-282. DOI: 10.1109/ICCAD.2017.8203789

[9] Yun-Peng, Z., Xia, L., Qiang, W. Asymmetric cryptography algorithm with Chinese remainder theorem (2011) 2011 IEEE 3rd International Conference on Communication Software and Networks, ICCSN 2011, article № 6014606, pp. 450-454. DOI: 10.1109/ICCSN.2011.6014606

[10] Wang, H., Song, Z., Niu, X., Ding, Q. Key generation research of RSA public cryptosystem and Matlab implement (2013) Proceedings of 2013 International Conference on Sensor Network Security Technology and Privacy Communication System, SNS and PCS 2013, article № 6553849, pp. 125-129. DOI: 10.1109/SNS-PCS.2013.6553849

[11] Mohit, P., Biswas, G.P. Modification of symmetric-key des into efficient asymmetric-key des using RSA (2016) ACM International Conference Proceeding Series, 04-05-March-2016. DOI: 10.1145/2905055.2905352

[12] Mansour, A., Davis, A., Wagner, M., Bassous, R., Fu, H., Zhu, Y. Multi-asymmetric cryptographic RSA scheme (2017) ACM International Conference Proceeding Series, article № a9, . DOI: 10.1145/3064814.3064820

[13] Logachev O.A., Salnikov A.A., Yashchenko V.V. Boolean functions in coding theory and cryptology - Moscow: ICNMO, 2004. – 470 p.(In Russian).