Using the Sum of Real Type Functions to Encrypt Messages

Viktor Avramenko[1], Mykyta Bondarenko[2]

[1,2]*Sumy State University, Rymskogo-Korsakova st. 2., Sumy, 40007, Ukraine*

Abstract

This paper presents a symmetric key cryptosystem using the sum of real type functions which allows to increase the cryptographic strength. Both transmitter and receiver choose Key Functions with the same argument, the interval for setting the argument, and the step for changing it. The symbol of the transmitted message is encrypted in an array where each element is the sum of Key Functions with random amplitudes. This sum includes those Key Functions for which the corresponding bit is one. Decryption uses disproportion functions. The system is suitable for encrypting both discrete and continuous messages.

Keywords

Cryptosystems, disproportion functions, function of real variable, key functions, encryption, decryption, text messages

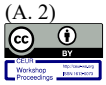
1. Introduction

Widely used cryptosystems are based on the set of integers. They implement symmetric and asymmetric encryption algorithms. In symmetric systems, the same key is used for both encryption and decryption. The most famous symmetric systems are AES [1] and GOST 28147-89 [2, 3]. To hack such a system, an enumeration of possible keys is required. The brute-force complexity is $O(2^k)$, where k is the key length in bits. For symmetric systems, if the communication channel is open, there is a problem of secure key transmission. This problem does not exist for asymmetric open key systems. In these systems, the most widely used algorithms are RSA and El-Gamal [4, 5]. The RSA algorithm is based on the computational complexity of the integer factorization problem. El-Gamal's algorithm is based on the difficulty of computing the discrete logarithm, especially over a group of points of an elliptic curve [6]. For breaking asymmetric cryptosystems, there are cryptanalysis methods which are faster than full search. This circumstance makes it necessary to use longer keys compared to keys in symmetric systems, but it's not promising due to the intensive development of the quantum computers [7], which will significantly affect the cryptographic strength of existing cryptosystems [8]. The ordinal brute force has complexity $O(2^k)$, meanwhile Grover's quantum algorithm [9] reduces it to $O(2^{k/2})$ [9].

Implementing quantum algorithms will also reduce the robustness of asymmetric systems. The RSA system uses the super polynomial computational complexity of the factorization of natural numbers. At the same time, there is a quantum algorithm whose complexity is polynomial $O(n^3)$ [10]. It means the cryptographic strength of asymmetric systems can be reduced as a result of the implementation of Shor's quantum algorithm for computing the discrete logarithm. In [11], Shor's algorithm is given for the group of points of an elliptic curve over the field GF(p) with complexity $O(n^3)$. Implementing quantum algorithms will also reduce the robustness of asymmetric systems. A method for increasing the crypto resistance of the system under these conditions is proposed in [12]. Along with the search for ways to hack cryptosystems, methods for detecting signals of means of secretly

III International Scientific And Practical Conference "Information Security And Information Technologies", September 13–19, 2021, Odesa, Ukraine

EMAIL: vv.avramenko@cs.sumdu.edu.ua (A.1);
nikbond97@gmail.com (A. 2)
ORCID: 0000-0002-6317-6711 (A. 1); 0000-0002-8849-7378 (A. 2)



obtaining information are also being developed [13].

The above analysis shows that one should look for other ways to create cryptosystems. In particular, to complicate the selection of keys using the simple enumeration method, one should switch from using integers to real ones. It is known [14] the set of real numbers has a higher cardinality compared to the set of natural numbers, so one can expect the cryptographic strength of a cryptosystem based on real numbers will be higher. The possibilities of creating cryptosystems using one or more functions of a real variable as keys are considered in [15-18].

So, in [15], characters from the ASCII code table are encrypted by the sum of 10 functions of a real variable, which are keys. Each key-function is preceded by a coefficient, which, depending on the character being encrypted, is equal to zero or one. The amplitudes of these functions are random for each new symbol. The resulting sum of the values of the functions is transmitted over the communication channel. On the receiving side, fragments of key functions are recognized, which are represented in the received encrypted signal. This allows you to decrypt the symbol transmitted at the current time using the disproportion functions [19-22].

In [16, 17], a variant is proposed when symbols for transferring binary codes are encoded with the help of three key functions of a real variable. "1", "0", "space", "new line" are encoded. Any other character is recognized as a new line. For unauthorized access to the intercepted message, you need to select the type and parameters of the key functions.

In [15-17], the disproportion functions over the first-order derivative were used. In this case, it is necessary to apply numerical methods for calculating the current values of the first derivatives. The need for these calculations led to the fact that the ciphertext significantly exceeded the length of the encrypted message.

A completely different encryption principle was proposed in [18]. One function of the real variable is used as the key. The disproportion function of the numerical representation of the encrypted process is calculated with respect to the key function. The obtained values of the disproportion function are an encrypted message and are transmitted over the communication channel. To avoid calculating the derivatives, the integral disproportions of the first order is used [23].

The cryptosystems [15-18] in the process of computer modeling have shown high cryptographic strength when trying to guess the parameters of keys functions, even if their form is known. To further complicate the work of cryptanalysts, the task is to develop a cryptosystem that could combine the advantages of the systems considered in [15-17] and the system [18]. So, it's necessary to develop the algorithms for encryption and decryption of analog and discrete messages, using several functions of a real variable as keys without the necessity to calculate derivatives.

2. Mathematical formulation of the problem

The message that is encrypted is a sequence of T numeric character codes from the ASCII table (or numeric values of the pixel brightness components in the case of a graphic image transmission). Each of them is encrypted using one-dimensional arrays of length N values. These arrays are obtained using one and the same step h of changing the argument of m Key Functions of the real variable. In this case, the value y(j, i) of the matrix y(T, N) has the form:

$$y(j,i) = \sum_{q=1}^{m} k_{qj}\, f_q(i), \qquad (1)$$

where:
j is the number of the character in the transmitted message;
$f_q(i) = f_q(ih)$, (i = 1, 2, … N > m), (q = 1, 2, … m) - an array of values of the q-th Key Function;
k_{qj} - coefficients that are generated during encryption of the j-th element and can be either equal to zero or represent random numbers which are unknown to the recipient.

Key Functions can be either continuous or discrete. These functions should be the same for the transmitting and receiving sides and have the same numbering. Also, the step h of changing the argument of the Key Functions should be the same. An encrypted message in the form of a matrix y(T,N) is transmitted over an open communication channel. The task is to decrypt the message using the matrix received at the receiving end. To solve it, the integral disproportion of the first order is used [23].

3. Disproportion functions

One of the first publications in which disproportion functions were proposed was [19]. In particular, the disproportion with respect to the n-th-order derivative of the function y(x) with respect to x is described by the expression:

$$@d_x^{(n)}y = \frac{y}{x^n} - \frac{1}{n!}\cdot\frac{d^n y}{dx^n}, \quad (2)$$

Here the @ symbol is chosen to denote the operation of calculating disproportion. The symbol "d" stands for "derivative". The order is indicated in parentheses. The left part (2) reads "et d n y with respect to x". The order $n \geq 1$ is an integer. If for any value of x, the function y(x) has the form $y = kx^n$, then disproportion (2) is equal to zero regardless of the value of the coefficient k.

For the case when n = 1,

$$@d_x^{(1)}y = \frac{y}{x} - \frac{dy}{dx}, \quad (3)$$

For defining the functions parametrically, when, $x = \varphi(t)$, $y = \psi(t)$, where t is a parameter, disproportion (3) is described by the expression

$$@d_{\varphi(t)}^{(1)}\psi(t) = \frac{\psi(t)}{\varphi(t)} - \frac{d\psi/dt}{d\varphi/dt}, \quad (4)$$

For $\psi(t) = k\varphi(t)$ disproportion (4) is equal to zero in the entire area of existence $x = \varphi(t)$, regardless of the value of k. In [19], the case was considered when

$$y(x) = k_1 f_1(x) + k_2 f_2(x) + \cdots + k_m f_m(x), \quad (5)$$

where $f_1(x)$, $f_2(x)$, … $f_m(x)$ are known functions; k_1, k_2, … k_m are coefficients whose values are unknown.

It is shown that the disproportion functions allow calculating the values of the unknown coefficients in (5) from the data obtained for the current value of the argument. This opportunity is used to create cryptosystems [15-17].

In practice, often the first derivative of the function does not exist or is equal to zero on some interval. This excludes the possibility of using disproportions over the first-order derivative (2-4). In this case, it is advisable to use the integral disproportion of the first order [23]. This disproportion of the function y(x) with respect to f(x) has the form:

$$@I_{f(x)}^{(1)}y(x) = \frac{\int_{x-h}^{x} y(x)dx}{\int_{x-h}^{x} f(x)dx} - \frac{y(x)}{f(x)}, \quad (6)$$

where h - is the preset time interval. In the discrete representation of signals, this is a time quantization step.

In this case, y(x) and f(x) are represented by one-dimensional arrays. If the approximate values of the integrals in (6) are calculated using the trapezoid formula, then for the one and the same step h for y(x) and f(x), disproportion (6) takes the following form (7):

$$@I_{f_i}^{(1)}y_i = \frac{y_{i-1} + y_i}{f_{i-1} + f_i} - \frac{y_i}{f_i}, \quad (7)$$

4. Encrypting and decrypting messages

The transmitting and receiving sides must have the same system of m Key Functions of the real variable, their numbering, the interval of changing the argument and step h of its change. The number of elements N of the one-dimensional array corresponding to the encrypted character must also be set. These can be both characters from the ASCII table, and components of pixel brightness when transmitting color graphic images. Each of them is represented by an integer. The required number of Key Functions depends on the maximum value of this number. For example, to encrypt characters from the ASCII table, m = 8 Key Functions are required. They can be either continuous or discrete. If the Key Functions are continuous, it is necessary to calculate N elements of one-dimensional arrays of their values, changing the argument from the initial x_{min} to the final x_{max} value with a step h. When encrypting characters from the ASCII table or the pixel brightness, their numerical representations differ by one. In these cases, the step h of changing the argument must be equal to one.

An m-bit binary code corresponds to each encrypted character. Each bit in this code is associated with a specific number of the Key Function. If the bit is zero, the value of the corresponding Key Function is also zero. If the bit is equal to one, then a random value of the amplitude of the corresponding Key Function is played. The character to be encrypted is represented by the sum (1).

4.1. Encrypting messages

1. The following is a character encryption algorithm:
2. Calculate arrays of N > m values of Key Functions: $f_q(x)$, q = 1, 2… m.
3. Enter the encrypted j-th character and calculate its cipher in the form of values of the

one-dimensional array y(j, i), i = 1, 2... N according to (1).

4. Repeat this point for all characters of the message of length T.

5. A sequence of T arrays is an encrypted message transmitted over an open communication channel.

4.2. Decrypting messages

Pre-compute the arrays $f_q(i) = f_q(ih)$, (q = 1, 2, ... m), (i = 1, 2, ... N > m), of Key Functions and to receive T one-dimensional arrays y(j , i), j = 1, 2, ... T, i = 1, 2, ... N over the communication channel. Further, in order to simplify the description of the decryption process, an example is given when only three functions are used in the cryptosystem - the keys: $f_1(x)$, $f_2(x)$, $f_3(x)$. In this case m = 3. Accordingly, the j-th character of the message is encrypted as

$$y(j,i) = k_{1j}f_1(i) + k_{2j}f_2(i) + k_{3j}f_3(i), \quad (8)$$
$$i = 1, 2, \dots N > 3,$$

The process consists of m = 3 levels in accordance with the number of Key Functions.

First level: It is necessary to calculate the array of disproportions (7) y(j,i) with respect to any of the Key Functions , for example, $f_1(i)$:

$$F_{01}(j,i) = @I^{(1)}_{f_1(i)} y(j,i) = \quad (9)$$
$$\frac{y(j,i-1) + y(j,i)}{f_1(i-1) + f_1(i)} - \frac{y(j,i)}{f_1(i)},$$

where i = 2, 3…N.

Also calculate the disproportions (7) of the remaining key functions with respect to $f_1(i)$:

$$F_{r1}(j,i) = @I^{(1)}_{f_1(i)} f_r(j,i) \quad (10)$$
$$= \frac{f_r(j,\ i-1) + f_r(j,i)}{f_1(i-1) + f_1(i)} - \frac{f_r(j,i)}{f_1(i)},$$

where r = 2, 3.

Considering that the disproportion of the function relative to itself is zero, we get:

$$F_{01}(j,i) = k_{2j}F_{21}(j,i) + k_{3j}F_{31}(j,i), \quad (11)$$

Second level: It is necessary to select any disproportion from right-hand of (11), for example $F_{21}(j, i)$. It is used to calculate next disproportions:

$$F_{0121}(j,i) = @I^{(1)}_{F_{21}(j,i)} F_{01}(j,i) \quad (12)$$
$$= \frac{F_{01}(j,\ i-1) + F_{01}(j,i)}{F_{21}(j,\ i-1) + F_{21}(j,i)} - \frac{F_{01}(j,i)}{F_{21}(i)},$$

$$F_{3121}(j,i) = @I^{(1)}_{F_{21}(j,i)} F_{31}(j,i) \quad (13)$$
$$= \frac{F_{31}(j,\ i-1) + F_{31}(j,i)}{F_{21}(j,\ i-1) + F_{21}(j,i)} - \frac{F_{31}(j,i)}{F_{21}(i)},$$

Taking into account that the disproportion of $F_{21}(j, i)$ with respect to $F_{21}(j, i)$ is equal to zero, we get:

$$F_{0121}(j,i) = k_{3j}F_{3121}(j,i), \quad (14)$$

Third level: The disproportion of $F_{0121}(j, i)$ with respect to $F_{3121}(j,i)$ is calculated in the following way

$$F_{01213121}(j,i) = @I^{(1)}_{F_{3121}(j,i)} F_{0121}(j,i) \quad (15)$$
$$= \frac{F_{0121}(j,\ i-1) + F_{0121}(j,i)}{F_{3121}(j,\ i-1) + F_{3121}(j,i)}$$
$$- \frac{F_{0121}(j,i)}{F_{3121}(i)} = 0,$$

It is equal to zero because, as can be seen from (14), there is a proportional relationship between $F_{0121}(j, i)$ and $F_{3121}(j, i)$. This fact allows calculating from (14) k_{3j} and k_{2j}, k_{1j} for the j-th message symbol.

$$k_{3j} = \frac{F_{0121}(j,i)}{F_{3121}(i)}, \quad (16)$$

$$k_{2j} = \frac{F_{01}(j,i) - k_{3j}\,F_{31}(j,i)}{F_{21}(i)}, \quad (17)$$

$$k_{1j} = \frac{y(j,i) - k_{2j}f_2(i) - k_{3j}f_3(i)}{f_1(i)}, \quad (18)$$

Depending on which of these coefficients are nonzero and which are equal to zero, the j-th message symbol is decrypted. In practice, it must be taken into account that there are calculation errors.

Therefore, it is necessary to compare the disproportion (15) calculated at the last level in modulus not strictly with zero, but with an approximate number ε. For example, it could be $\varepsilon=10^{-4}$. In this case, if $| F_{01213121}(j, i) | <= \varepsilon$, then it should be assumed that it is zero.

The value of ε is determined during testing of the cryptosystem. Theoretically, this disproportion is equal to zero for all i = 2, 3, ... N, but, taking into account the calculation errors, it is recommended to do calculations using formulas (16-18) for i, at which the modulus of disproportion (15) is minimal.

4.3. An example of encrypting and decrypting characters from an ASCII table

Eight Key Functions are used (m =8):

1. $f_1(x) = 1000 \sin((\alpha_1 - \beta_1)x) \cos(w\beta_1 x)$
2. $f_2(x) = 1000 \exp(0.1\alpha_2 x) \sin(w\beta_2 x) \cos((\alpha_2 + \beta_2)x)$
3. $f_3(x) = 1000 \exp(-\alpha_3 x) \sin(w\beta_3 x)$
4. $f_4(x) = 1000 \cos((\alpha_1 x - \beta_1)x) \sin(w\beta_1 x)$
5. $f_5(x) = 1000 \exp(0.1\sin(\alpha_2 x)) \sin(w \cos(\beta x)) \cos((\alpha_2+\beta_2) x)$
6. $f_6(x) = 1000 \sin(-\cos(\alpha_3 x)) \cos(w \sin(\beta_3 x))$
7. $f_7(x) = 1000 \sin(wx + \alpha_1) \exp(-\beta_1 x^2)$
8. $f_8(x) = 1000 \cos(w\gamma x^2)$

where $\alpha_1 = 1$, $\alpha_2 = 0.12$, $\alpha_3 = 0.5$, $\beta_1 = 0.1$, $\beta_2 = 1.5$, $\beta_3 = 0.7$, $\gamma = 0.5$, $w = 400$ are constants.

A sequence of numbers corresponding to the transmitted characters from the ASCII code table is encrypted. Each character is encoded by a sum of Key Functions

$$y(x) = k_1 f_1(x) + k_2 f_2(x) + k_3 f_3(x) + k_4 f_4(x) + k_5 f_5(x) + k_6 f_6(x) + k_7 f_7(x) + k_8 f_8(x), \quad (19)$$

where:

x = ih – argument;

h = 1 – step of changing the argument.

i – is the ordinal number of the element of the one-dimensional array for each of the Key Functions, as well as the array $y_0, y_1, ..., y_{N-1}$, which is the character cipher;

N – a number of elements of each one-dimensional array. Based on the requirement of N > m, the amount of array elements N = 16.

Table 1 shows the transmitted characters in the upper horizontal line. The corresponding ciphers are given in the form of arrays arranged vertically. The decrypted characters are located horizontally on the bottom line.

It is obvious that the received message matches the transmitted one. It should be noted that the ciphers (arrays) of the adjacent symbols `t` are completely different.

The codes of the other adjacent identical symbols in the message are given in Table 2. The above results indicate that the ciphers of the adjacent identical symbols in the message differ from each other. This circumstance greatly complicates the "hacking" of the cryptosystem. In order to "crack" the message, it is required to select the form of eight Key Functions and the values of their parameters.

Table 1

Encrypted and decrypted characters “Hello”

y	‘H’	‘e’	‘l’	‘l’	‘o’
0	-323.36050	-1096.0141	-872.47149	37.134528	-112.93721
1	257.702939	167.391848	1051.01033	532.400561	427.740614
2	57.298613	175.907791	-408.37541	-216.26334	-116.65218
3	-165.32821	126.358160	-324.75198	-162.19800	-197.22270
4	-186.82906	-394.77504	-929.02530	-439.94548	-449.01146
5	-163.70378	-392.33753	-1059.2853	-385.85981	-170.75848
6	37.446166	299.880685	370.310135	74.675455	44.746439
7	-110.70494	426.787248	-218.90900	-238.68403	-314.75860
8	-2.714026	-59.278796	115.564604	-2.371129	-165.23427
9	9.436954	152.916970	116.697501	-23.361501	281.220083
10	42.465400	-412.02347	-203.82595	84.424717	150.029168
11	-24.295297	615.002251	349.117675	-42.371452	-231.55086
12	101.570285	95.754178	543.764259	227.452661	-122.68102
13	195.422364	132.219196	855.514365	432.359247	754.345004
14	-11.067358	-507.14921	-252.27430	-29.696351	228.457327
15	214.445079	264.682389	659.731238	467.369970	-63.039751

Table 2

Encrypted and decrypted characters ‘A’

y	‘A’	‘A’	‘A’	‘A’	‘A’
0	-597.135343	-762.540347	-609.489179	-1245.052456	-917.400855
1	9.473961	-6.667141	-2.760240	-37.310266	-17.407304
2	274.695430	368.229254	291.933312	625.796927	451.736269
3	15.193014	87.216540	60.428145	237.898008	138.849052
4	-123.497929	-217.377766	-165.579209	-438.953490	-291.370618
5	-58.830278	-107.584825	-81.548078	-221.367775	-145.669069
6	-8.280530	-11.911812	-9.337867	-21.332769	-14.999968
7	199.488556	342.700766	261.876769	683.403833	456.291669
8	-76.316724	-131.227388	-100.265637	-261.818697	-174.769533
9	-60.158608	103.377062	78.993047	-206.183725	137.653657
10	-125.104506	-215.011307	-164.292499	-428.868345	-286.313719
11	214.641127	368.892513	281.874942	735.803375	491.224813
12	-14.047252	24.142272	18.447384	-48.154847	-32.148341
13	6.530344	11.223364	8.575899	22.386440	14.945262
14	-223.052958	-383.349601	-292.921754	-764.640006	-510.476209
15	169.891087	291.983039	223.107534	582.397666	388.810617

Below is an example that illustrates the resistance of the system obtaining keys, even if somehow it was possible to find out the forms of Key Functions. Suppose that the above sequence of characters is encrypted using functions (7), and decrypted using the same kind of functions, but the constant w was guessed incorrectly. Instead of w = 400 was used w = 399.999 during decryption. In this case, the disproportion at the last eighth level in absolute value exceeds the permissible deviation ε from zero. That is, decryption is impossible. Only if w = 399.9999, the message may be decrypted. This result shows that even such a slight deviation of one of the parameters of the Key Functions does not allow decryption of the transmitted character.

5. Requirements for Key Functions

1. The Key Functions must be of real type.
2. They can’t be constant and must not take zero values.
3. When using the key function, there should be no situation where division by a number

close to zero occurs, which leads to an unacceptable calculation error. For this purpose, it is recommended to test the cryptosystem for the entire alphabet of characters that will be used in messages.

4. Check that the sum of two or more key functions does not coincide with any other of the key functions.

5. It is recommended to include all parameters in the expression for each key function. In this case, a change in the value of any parameter leads to a change in all key functions, but not one or several of them only.

6. Before sending an encrypted message, first check what the decrypted message looks like in order to avoid errors that may occur as a result of not taking into account the previous points.

6. Conclusions

A cryptosystem with symmetric keys is proposed. These keys are real variable functions that satisfy the above constraints. They can be either continuous or discrete. The number of functions is equal to the number of binary digits used to encrypt a character, for example, in an ASCII table. Each of the functions corresponds to a certain binary digit. The symbol of the transmitted message is encrypted with a one-dimensional array. The elements of this array represent the sum of Key Functions with random amplitudes. This sum includes those Key Functions, for which the corresponding binary digit is equal to one.

Decryption is performed using disproportion functions. The possibility of encryption and decryption of text information is shown. The given examples show the complexity the guessing Key Functions and the cryptographic strength of the proposed cryptosystem. So, for example, a real-type constant, which equals 400 during encryption, to break the system by brute-force, you need to select with an accuracy of 10^{-4}, but there can be any number of such constants. It is very difficult to find all the constants of the real type at the same time with high precision and thus hack the system, even with well-known formulas of functions - keys.

It should also be noted that the codes of the same adjacent symbols are not repeated, which can be seen from Table 2. This also increases the cryptographic strength of the system.

7. References

[1] National Institute of Standards and Technology, Specification for the ADVANCED ENCRYPTION STANDARD (AES) (2001). doi: 10.6028/NIST.FIPS.197.

[2] GOST 28147-89. Sistemy obrabotki informacii. Zashhita kriptograficheskaya. Algoritm kriptograficheskogo preobrazovaniya [Information processing systems. Cryptographic protection. Cryptographic Transformation Algorithm], 1990.

[3] A. N. Lebedev, Kriptografiya s «otkrytym klyuchom» i vozmozhnosti ee prakticheskogo primeneniya [Cryptography with "public key" and the possibilities of its practical application], Zashhita informacii. Konfident 2 (1992)

[4] R. Rivest, A. Shamir, I. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM, 1978, 21(2):120-126. doi:10.1145/359340.359342.

[5] I. D. Gorbenko., Y. I. Gorbenko, Prykladna kryptolohiya [Applied Cryptology], Fort, NURE, Kharkiv, 2012, p. 878.

[6] D. R. Hankerson, S. A. Vanstone and A. J. Menezes., Guide to elliptic curve cryptography, Springer, New York, 2003, p. 311.

[7] T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, J. L. O'Brien, Quantum Computing, Nature, 464 (2010) 45—53. doi: 10.1038/nature08812.

[8] P. G. Klucharev, Kvantovyj komp'yuter i kriptograficheskaya stojkost' sovremennyx sistem shifrovaniya [Quantum computer and cryptographic strength of modern encryption systems], Herald of the Bauman Moscow State Technical University, Series Natural Sciences 2 (2007).

[9] I.K. Grover, Quantum Mechanics Help in Searching for a Needle in a Haystack, Phys. Rev. Lett. 79, 325 (1997): 326-328. doi: 10.1103/PhysRevLett.79.325.

[10] P. Shor, Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, in: Proceedings of the 35th Annual Symposium of Foundations of Computer Science, 1994. doi: 10.1137/S0097539795293172.

[11] J. A. Proos, Shor's discrete logarithm quantum algorithm for elliptic curves, Faculty of Mathematics University of Waterloo, Waterloo, 2003, p. 35.

[12] S. Yevseiev, R. Korolyov, A. Tkachov, O. Laptiev, I. Opirskyy, O. Soloviova. Modification of the algorithm (OFM) S-box, which provides increasing crypto resistance in the post-quantum period, International Journal of Advanced Trends in Computer Science and Engineering, 2020, volume 9, pp. 8725-8729. doi:10.30534/ijatcse/2020/261952020.

[13] O. Barabash, O. Laptiev, V. Tkachev, O. Maystrov, O. Krasikov, I. Polovinkin, The Indirect method of obtaining Estimates of the Parameters of Radio Signals of covert means of obtaining Information, International Journal of Emerging Trends in Engineering Research, 2020, volume 8, pp. 4133 – 4139. doi:10.30534/ijeter/2020/17882020.

[14] A. N. Kolmogorov, S. V. Fomin, Ehlementy teorii funkcij i funkcional'nogo analiza [Elements of function theory and functional analysis], Science, Moscow, 1972.

[15] V. V. Avramenko, M. I. Zabolotny, A Way of Data Coding, 2009. Patent No. 42957, Filled March 16th, 2009, Issued July 27th, 2009.

[16] V. V. Kalashnikov, V. V. Avramenko, N. I. Kalashnikova and Kalashnikov Jr. V.V., A Cryptosystem Based Upon Sums of Key Functions, International Journal of Combinatiorial Optimization Problems and Informatics, 2017, volume 8, pp. 31-38.

[17] N. I. Kalashnikova, V. V. Avramenko, V. Kalashnikov. Sums of Key Functions Generating Cryptosystems, in: ICCS 2019, Chapter 23, Lecture Notes in Computer Science, vol. 11540, Springer, Cham, 2019. doi: 10.1007/978-3-030-22750-0_23.

[18] V. V. Avramenko, V. Demianenko, in: CEUR Workshop Proceedings, 2020, 2608, pp. 661-674. doi: 10.15588/1607-3274-2020-2-8.

[19] V. V. Avramenko, Characteristic properties of disproportionality functions and their application to solving diagnoses problems, Transactions of Sumy State University, SSU, Sumy, 2000, №16, pp. 24-28.

[20] V. V. Kalashnikov, V. V. Avramenko, N. I. Kalashnykova, Derivative disproportion functions for pattern recognition, in: Watada, J., Tan, S.C., Vasant, P., Padmanabhan, E., Jain, L.C. (eds.) Unconventional Modelling, Simulation, and Optimization of Geoscience and Petroleum Engineering, pp. 95–104. Springer, Heidelberg, 2018.

[21] V. V. Kalashnikov, V. V. Avramenko, N. Y. Slipushko, N.I. Kalashnykova, N.I., A. E. Konoplyanchenko, Identification of quasi-stationary dynamic objects with the use of derivative disproportion functions, Procedia Comput. Sci., (2017) 108(C): 2100–2109.

[22] V. V. Avramenko, A. Moskalenko, Operative Recognition of Standard Signals in the Presence of Interference with Unknown Characteristics, in: Proceedings of the Second International Workshop on Computer Modeling and Intelligent Systems (CMIS-2019), Zaporizhzhia, 2019.

[23] A. P. Karpenko, Integral'nye harakteristiki neproporcional'nosti chislovyh funkcij i ih primenenie v diagnostike [Integral characteristics of the disproportionality of numerical functions and their application in diagnostics], Vestnik Sumskogo gos. un-ta. 16(2000): 20-25

[24] Vitalii Savchenko, Oleksandr Laptiev, Oleksandr Kolos, Rostyslav Lisnevskyi, Viktoriia Ivannikova, Ivan Ablazov. Hidden Transmitter Localization Accuracy Model Based on Multi-Position Range Measurement. 2020 IEEE 2nd International Conference on Advanced Trends in Information Theory (IEEE ATIT 2020) Conference Proceedings Kyiv, Ukraine, November 25-27. pp.246 –251

[25] Valentin Sobchuk, Volodymyr Pichkur, Oleg Barabash, Oleksandr Laptiev, Kovalchuk Igor, Amina Zidan. Algorithm of control of functionally stable manufacturing processes of enterprises. 2020 IEEE 2nd International Conference on Advanced Trends in Information Theory (IEEE ATIT 2020) Conference Proceedings Kyiv, Ukraine, November 25-27. pp.206 –211.

[26] Oleksandr Laptiev, Oleh Stefurak, Igor Polovinkin, Oleg Barabash, Savchenko Vitalii, Olena Zelikovska. The method of improving the signal detection quality by accounting for interference. 2020 IEEE 2nd International Conference on Advanced Trends in Information Theory (IEEE ATIT 2020) Conference Proceedings Kyiv, Ukraine, November 25-27. pp.172 –176.

[27] Mashkov V.A. and Barabash O.V. Self-Testing of Multimodule Systems Based on Optimal Check-Connection Structures. Engineering Simulation. Amsterdam: OPA, 1996. Vol. 13, pp. 479 – 492.

[28] S. Toliupa, N. Lukova-Chuiko, O. Oksiuk. Choice of Reasonable Variant of Signal and

Code Constructions for Multirays Radio Channels. Second International Scientific-Practical Conference Problems of Infocommunications. Science and Technology. IEEE PIC S&T 2015. pp. 269 – 271.