Cryptosystem Based on a Key Function of a Real Variable

Viktor Avramenko[1[0000-0002-6317-6711]] and Volodymyr Demianenko[2[0000-0002-1512-970X]]

[1,2]Sumy State University, Rimsky-Korsakov st., 2, Sumy, 40007, Ukraine
[1]avramenko1938@gmail.com, [2]vldemyan@gmail.com

Abstract. Using the function of a real variable in cryptosystems as a key allows you to increase its cryptographic strength, because it is more difficult to pick up such key. Therefore, the development of such systems is relevant. A cryptosystem with a symmetric key is offered. This key is some function of a real variable that satisfies some restrictions. It can be either continuous or discrete.

The transmitting and receiving parties select the key-function, the first transmitted character or the first transmitted value for the analog message, the function area of the key function, and the step of changing the function argument. A Disproportion over first-order derivative is used to encrypt an analog message.

The Cauchy problem is solving for decrypting this message. Discrete messages are encrypted using the first-order disproportionality integral function. Decryption is performed by the inverse transformation of the formula for integral disproportion.

Algorithms for encrypting and decrypting messages are presented. The ability to encrypt and decrypt text information, 2D graphic images, as well as analog messages are shown. The examples show the complexity to pick up the key function and the cryptographic strength of the proposed cryptosystem.

A cryptosystem, in which the function of a real variable is used as a key and as well as disproportion functions are used, is suitable for encryption of both discrete and continuous messages. To "crack" such a system, it is required to pick up the form of the key function and to find the values of its parameters with very high accuracy. That is, the system has high cryptographic strength.

Keywords: cryptosystems, disproportion functions, real variable functions, key-function, encryption, decryption, text messages, 2D images, analog messages.

1 Introduction

It is difficult to imagine the modern world without systems of secrecy of the transmitted information in both the military and business fields. Therefore, it is not surprising that they are constantly being improved. In addition, new ways of hiding information are emerging. The most widely used encryption algorithms are divided into symmet-

ric and asymmetric. In symmetric systems, encryption and decryption are performed using the same key. Such systems include AES [1], GOST 28147-89 [2, 3]. Hacking them requires enumeration of possible keys. The complexity of exhaustive search is estimated as O ($2k$), where k is the key length in bits.

In asymmetric systems (public key systems), RSA and El-Gamal algorithms are most widely used [4, 5]. The RSA algorithm is based on the computational complexity of the integer factorization problem. The reliability of the El-Gamal algorithm is based on the difficulty of computing the discrete logarithm. Of particular interest is the El-Gamal algorithm over the group of points of an elliptic curve [6].

For asymmetric algorithms, there are cryptanalysis methods that work faster than full search. Therefore, asymmetric algorithms have to use longer keys than keys in symmetric systems.

The disadvantages of both symmetric and asymmetric systems include the fact that they are based on a set of integers. This allows you to develop various methods of hacking such cryptosystems up to the implementation of a simple selection of keys. To increase cryptographic strength, you have to gradually increase the length of the keys.

However, due to the increasing capabilities of computer technology, the way to increase the length of keys is unpromising. Currently, great efforts are being made to create quantum computers [7]. It is expected that their use will significantly affect the cryptographic strength of existing cryptosystems [8]. The well-known Grover quantum algorithm [9] for restoring the key of a symmetric encryption algorithm from the message text and cipher text. Full search on a regular computer has complexity O ($2k$), where k is the key length. For a quantum computer, this complexity is 2 times less and amounts to O ($2k / 2$) [9]. That is, the efficiency of the key length is reduced by 2 times.

Quantum algorithms also pose a threat to systems with asymmetric encryption. The cryptographic strength of the RSA encryption system is based on the super-polynomial computational complexity of factoring natural numbers. However, there is a quantum algorithm, whose complexity is polynomial O (n^3) [10].

There is also a Shore quantum algorithm for computing the discrete logarithm. In [11], the Shor algorithm for a group of points of an elliptic curve over a field GF (p) with complexity O (n^3) is presented.

At the same time, it is known [12] that the set of real numbers has a greater power in comparison with the set of natural numbers. Therefore, the development of cryptosystems using real numbers is of practical interest. It can be expected that they may be more persistent in the event of an attempt to select a key by enumeration.

This paper presents one of the many possible options for such the cryptosystem.

2 Problem statement

It is necessary to develop algorithms for encrypting and decrypting analog and discrete messages using a function of a real variable as a key.

3 Literature review

Mostly modern cryptosystems use a block cipher. It operates with groups of bits of a fixed length - blocks. Their size can be from 64 to 256 bits.

The widely used block symmetric ciphers (BSS) are based on the use of several symmetric cryptographic transformations (elementary ciphers).

When constructing them, three main approaches are used:

- based on Feistel chains;
- IDEA-like ciphers;
- SPN (Substitution Permutation Network) structure. The AES system belongs to SPN based BSN [1]. In it, cryptographic conversions are performed in the simple substitution mode over 128-bit blocks (16 bytes), which has a block length of 128 bits. The key length has several options: 128, 192, 256 bits.

The number of cycles of conversion depends on the length of the key. It is, respectively, 10, 12 or 14 cycles.

The encryption algorithm GOST 28147-89 [2, 3] also belongs to the category of block ciphers, where two parts of the selected block of information are of equal size.

It is a classical symmetric encryption algorithm based on the Feistel network and is characterized by high cryptographic strength. However, this algorithm also has disadvantages:

1) In comparison with the byte-oriented algorithm, AES on 8-bit platforms GOST loses in speed by 4 times.

2) In the text of the standard GOST 28147-89, there are no clear criteria for the selection of replacement nodes. Quite often, fears are expressed that there are weak replacement nodes.

In 1978, three authors: Ronald Rivest, Adi Shamir, Leonard Adleman proposed the RSA algorithm [4, 5]. This algorithm was the first full-fledged public key algorithm. Encryption is carried out without the transfer of secret keys.

In addition to classical cryptosystems using integers as keys, systems based on functions of real variables were proposed [13]. Symbols from the ASCII code table are encrypted with a sum of 10 key functions with coefficients before them zero or one. The amplitudes of these functions during encryption of each new character are taken randomly. At the receiving end, using the disproportion functions [14-16], fragments of key functions that are present in the received encrypted signal are recognized. It allows to decrypt the character transmitted at the current time.

In [17, 18], a variant was proposed when, three key functions of a real variable are used to encrypt a binary codes. Next symbols are coded: "1", "0", "space", "the transition to a new row". Any other character is recognized as the transition to a new row. For unauthorized access to the intercepted message, you need to select the type and parameters of the key functions.

Both cryptosystems in the process of computer simulation showed high cryptographic strength when trying to select the parameters of key functions even if their type is known.

However, using the sum of key functions and the need to recognize them at the receiving end complicates the algorithm. In addition, the encrypted message is much

longer than the original. Therefore, the task is to develop a cryptosystem using only one key function of a real variable.

4 Mathematical statement of the problem

A message that is encrypted is a sequence of numerical character codes (or numerical values of the components of the brightness of the pixels in the case of a graphic image), which is described by a discrete function y_i, i = 0, 1, ..., N-1, where N is the number of characters in the message. It is replaced by a sequence of ciphers that are calculated using the key function, which can be either discrete f_i, i = 0, 1, ..., M ($M > N$), or continuous $f(x)$. Here $x = i \cdot h$, where i is the sequence number of the character in the message, and h is the step with which the argument x changes. This step should be the same for both sides of the message transfer. Disproportion functions are used for encryption.

5 Disproportion functions

Several types of the disproportion function are known: the disproportion over n-th order derivative, the disproportion over n-th order value, relative and sequential disproportions. Virtually all of them are characteristics of numerical functions. Below is a summary of those that are used in this work.

The disproportion over n-th order derivative of the function $y(x)$ with respect to x is described by the expression:

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

Here, the @ symbol is chosen to indicate the operation of calculating disproportion. The symbol "*d*" means "derivative". The order is indicated in parentheses. The left side of (1) reads "at *d n y* with respect to *x*".

If for any value of x, the function $y(x)$ has the form $y = kx^n$, then disproportion (1) is equal to zero regardless of the value of the coefficient k.

The disproportion over 1-st order derivative (n = 1) has the form:

$$@\, d_x^{(1)} y = \frac{y}{x} - \frac{dy}{dx} \tag{2}$$

In case of a parametric specification of functions, when $x = \varphi(t)$, $y = \psi(t)$, where t is a parameter, disproportion (2) takes the form:

$$@d^{(1)}_{\varphi(t)}\psi(t) = \frac{\psi(t)}{\varphi(t)} - \frac{d\psi/dt}{d\varphi/dt} \tag{3}$$

For $\psi(t) = k\varphi(t)$ disproportion (3) is equal to zero in the entire area of existence $x = \varphi(t)$, regardless of the value of k.

If $y(x)$ is a sum of known functions taken with unknown coefficients, then the disproportion functions allow us to calculate the values of these coefficients from the data obtained for the current value of the argument. This opportunity was used both for creating cryptosystems [13, 17, 18], and in solving a number of more general problems [15, 16].

For the case when the first derivative does not exist or is equal to zero on any interval, it is proposed to use the first-order integral disproportion [19, 20]. This disproportion of the function $y(x)$ with respect to $f(x)$ has the form:

$$@I^{(1)}_{f(x)}y(x) = \frac{\int_{x-h}^{x} y(x)dx}{\int_{x-h}^{x} f(x)dx} - \frac{y(x)}{f(x)}, \tag{4}$$

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 (4) are calculated using the trapezoidal formula, then for the same step h for $y(x)$ and $f(x)$, disproportion (4) takes the form:

$$@I^{(1)}_{f_i}y_i = \frac{y_{i-1} + y_i}{f_{i-1} + f_i} - \frac{y_i}{f_i} \tag{5}$$

6 Encryption and decryption of text and image messages

Text messages are a sequence of character codes, for example, from an ASCII table. That is, it is a sequence of integers. When transmitting color graphic images, the brightness components of pixel-integer numbers from 0 to 255 are transmitted. As a key function, you can take any function of a real variable. It can be either a continuous or a discrete function. However, due to the fact that the message is represented by a discrete function that does not have a first derivative, in any case, integral disproportion should be used for encryption (5).

If the key function is continuous, you need to calculate an array of its values, changing the argument from the initial x_{min} to the final x_{max} values in increments of h. The parameters x_{min}, x_{max}, h must be the same for the transmitting and receiving sides. When encrypting the characters from the ASCII table or pixel luminance components,

their numerical representations differ by one. In these cases, the step h of changing the argument must be equal to one.

In addition, the transmission and, accordingly, the reception must begin with a certain symbol known to the transmitting and receiving sides.

Therefore, the encryption algorithm is as follows:

1. The start character that is known to both sides must be entered.
2. Enter an array of values of key function.
3. Read from the file or enter from the keyboard a sequence of message characters and convert them to the numbers. In this case, for text symbols the ASCII table can be used. As a result, the array y_i, $i = 0, 1, \ldots N\text{-}1$ will be get.
4. Using the code of the given initial character y_0, calculate the disproportion $I_i = @I_{f_i}^{(1)} y_i$, $i = 1, 2, \ldots, N\text{-}1$ for each next character in accordance with (5) and transmit them over the communication channel.

7 Decryption algorithm for text and image messages

1. Enter the start character y_0 that is known for the transmitting side.
2. If the key function is discrete, enter the array of its values. If the key function is continuous, calculate the array of its values in accordance with the instructions above.
3. Read from the file the accepted disproportion values for each of the symbols I_i, $i = 1, 2, \ldots N\text{-}1$.
4. Use the known initial symbol y_0, and the accepted disproportions I_i, $i = 1, 2, \ldots N\text{-}1$, for calculating the estimated codes of the recovered message according to the formula (6).

$$y_i = \frac{(y_i - I_i \cdot (f_{i-1} + f_i)) \cdot f_i}{f_{i-1}} \tag{6}$$

Round off estimated codes to the nearest integers to get recovered codes.

5. Using the recovered codes found, reproduce the message in symbolic form (when transmitting a graphic image, reproduce the corresponding pixel attributes).

8 Examples of encryption and decryption of discrete messages

Example 1

An example of encryption and decryption of characters from the ASCII code table is provided.

The key function has the form:

$$f(x) = ae^{\beta x} + \sin(\beta x) + \ln(ax + \beta) \tag{7}$$

$x = i \cdot h$ is an argument;

i is the sequence number of the character in the encrypted message;

$h = b + c$ is the step of changing the argument.

Here $a = 0.1$, $\beta = 0.01$, $b = 0.65$, $c = 0.35$ are constants.

Since in the example the characters are encrypted from the ASCII table, in accordance with the above algorithm, the sum of the constants b and c is equal to one.

The sequence of numerical codes y_i, $i = 1, 2, \ldots N$-1, corresponding to the transmitted characters, is encrypted. When transmitting messages, let's the first character is "G". In the ASCII table, its code is 71 (that is, $y_0 = 71$).

Given the discrete nature of the message, integral disproportion (5) is used to encrypt it.

Decryption is carried out using (6) and according to the above algorithm.

Table 1 shows the transmitted characters, their ciphers, and the corresponding decrypted characters.

Table 1. Codes of transmitted characters and the results of their decryption.

Character number	Source character	Character code	Decrypted character	Character number	Source character	Character code	Decrypted character
1		-0,33745		19	@	-14,9626	@
2	C	18,56566	C	20		17,18209	
3	o	35,05402	o	21	#	-0,50861	#
4	n	23,38864	n	22		2,084013	
5	t	45,28726	t	23	$	-1,0013	$
6	r	75,66021	r	24		2,205811	
7	o	215,5424	o	25	%	-1,38787	%
8	l	3638,486	l	26		2,342513	
9		1392,007		27	^	-21,4272	^
10	e	-49,5932	e	28		22,22681	
11	x	36,34984	x	29	&	-1,64792	&
12	a	60,50844	a	30		2,295671	
13	m	7,687781	m	31	*	-2,84342	*
14	p	11,55305	p	32		3,393497	
15	l	13,07326	l	33	(	-2,13477	(
16	e	11,94166	e	34		2,607738	
17	!	44,92703	!	35	)	-2,34347	)
18		2,001215					

It can be seen from it that the cipher of each character is a real number with both an integer and a fractional part. Encrypted and decrypted characters match.

In order to "crack" a message, you need to select the type of the key function and the values of its parameters.

The following is an example that illustrates the cryptographic strength of the system to obtain a key, even if somehow it was possible to find out the type of key-function. Suppose that the above sequence of characters is encrypted using function

(7), and decrypted using the same type of function, but the constant *a* instead of the value 0.1 during decryption is chosen incorrectly and is 0.1005.

Table 2 shows the results, from which it is seen that even such a slight deviation of the parameter of the key function does not allow to decrypt transmitted characters correctly.

Table 2. Codes of transmitted characters and the results of their decryption with an incorrectly selected key function.

Character number	Source character	Character code	Decrypted character	Character number	Source character	Character code	Decrypted character
1		-0,33745		19	@	-14,9626	D
2	C	18,56566	B	20		17,18209	
3	o	35,05402	n	21	#	-0,50861	M
4	n	23,38864	l	22		2,084013	Ґ
5	t	45,28726	q	23	$	-1,0013	-
6	r	75,66021	n	24		2,205811	Г
7	o	215,5424	g	25	%	-1,38787	T
8	l	3638,486	5	26		2,342513	П
9		1392,007	ќ	27	^	-21,4272	,
10	e	-49,5932	X	28		22,22681	®
11	x	36,34984	”	29	&	-1,64792	
12	a	60,50844	f	30		2,295671	e
13	m	7,687781	(	31	*	-2,84342	s
14	p	11,55305	r	32		3,393497	ъ
15	l	13,07326		33	(	-2,13477	A
16	e	11,94166	L	34		2,607738	q
17	!	44,92703	+	35	)	-2,34347	«
18		2,001215	2				

A positive feature of the proposed encryption system is that the cipher of one and the same symbol is not repeated and depends on its number in the message.

Table 3 shows the repeated characters, their ciphers, and decryption results.

Table 3. Results of decryption of the sequence of identical characters.

Character number	Source character	Character code	Decrypted character	Character number	Source character	Character code	Decrypted character
1	a	42,89534	a	23	%	-1,07004	%
2	a	55,98126	a	24	%	0,230793	%
3	a	223,1301	a	25	%	0,211779	%
4	a	-2722	a	26	%	0,195132	%
5	a	193,6388	a	27	%	0,180462	%

6	a	47,19846	a	28	%	0,167453	%
7	a	21,12258	a	29	%	0,155856	%
8	a	12,02764	a	30	%	0,145461	%
9	a	7,807816	a	31	%	0,136099	%
10	a	5,504747	a	32	%	0,127629	%
11		31,8935		33		1,189454	
12	A	-12,0762	A	34	9	-5,17547	9
13	A	1,71819	A	35	9	0,164057	9
14	A	1,413497	A	36	9	0,154966	9
15	A	1,186178	A	37	9	0,146586	9
16	A	1,011796	A	38	9	0,138837	9
17	A	0,874885	A	39	9	0,131652	9
18	A	0,765272	A	40	9	0,124967	9
19	A	0,67603	A	41	9	0,118735	9
20	A	0,602313	A	42	9	0,112905	9
21	A	0,540641	A	43	9	0,10744	9
22		9,210041		44		4,816542	

From Table 3 it can be seen that the ciphers of the same characters differ from each other, but at the same time, their decryption is error-free.

This once again shows that for hacking it is necessary to find the type of the key function, and then select the values of its parameters with high accuracy.

Example 2

The encryption of a graphic message is considered - a 2D image using the key function of a real variable

$$f(q) = ae^{\sin((b+c)q+a)}, \tag{8}$$

where q is the sequence number of the pixel;

$a = 100$, $b = 0.65$, $c = 0,35$.

The image of a seagull on the seashore is encrypted. According to the algorithm, transmission begins with a pixel known to the receiving side. Pixel brightness is also represented by integers with a step equal to one. Therefore, according to the algorithm in this case, it is required that the sum of a and b is equal to one.

Table 4 shows the disproportions of the brightness components for the first 23 pixels.

Table 4. Components brightness pixels' codes.

pixel number	disproportion of component A	component red disproportion	component green disproportion	component blue disproportion
1	-0,17260194	-0,052654658	-0,079113617	-0,109100438
2	0,249815554	0,122038436	0,147606293	0,183288106
3	-0,25138517	-0,066806224	-0,080056209	-0,073489858
4	-1,54785813	-0,671606862	-0,746100819	-0,895078346
5	-2,11017572	-1,141977451	-1,198335871	-1,339798239
6	0,638962347	0,336698458	0,322878349	0,344784243

7	0,904695026	0,493147485	0,536886675	0,603130017
8	0,35818895	0,185034305	0,203422857	0,224748673
9	-0,0619523	-0,079999226	-0,074320205	-0,069612985
10	-1,08142099	-0,6158245	-0,699207099	-0,762820098
11	-2,35243212	-1,569025352	-1,733111253	-1,953901199
12	-0,07899416	-0,025833335	-0,031099612	-0,000303777
13	1,059101779	0,630917816	0,701524602	0,778998655
14	0,482163337	0,302525074	0,337339747	0,360361396
15	0,079311391	0,042700382	0,052726876	0,071145629
16	-0,68481115	-0,484134182	-0,526938708	-0,575032293
17	-2,19903004	-1,601314543	-1,722045604	-1,879955093
18	-0,9877602	-0,663217085	-0,68687686	-0,706872244
19	1,094772111	0,607081863	0,704520639	0,846261488
20	0,632160636	0,56848388	0,574362813	0,580312445
21	0,193576492	0,084571962	0,09345652	0,077852313
22	-0,3793663	-0,320611934	-0,339514003	-0,33505087
23	-1,80013937	-1,01073232	-1,126174205	-1,185141129

It can be seen from it that the ciphers are real numbers with both an integer and a fractional part, the nature of the changes of which is difficult to predict.

Figure 1 shows the original (left) and decrypted (in the middle) images according to the disproportions received at the receiving end, known to the first pixel and the key function.

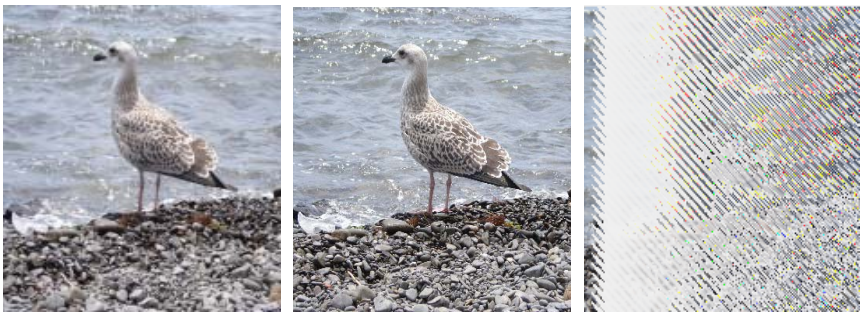

Fig. 1 Original image (left), decrypted (in the middle), decrypted with an incorrectly selected key-function parameter (right).

Figure 1 on the right shows the result when instead of $a = 100$, when decrypting the image, the value $a = 99.9999$ was mistakenly selected. Despite a slight difference from the value of this parameter during encryption, it was not possible to “crack” the image. The same result if $b + c = 0.999999$ is selected instead of unity.

This indicates the cryptographic strength of the encryption system and how difficult it is to get a key function by enumeration.

9 Examples of encryption and decryption of the analog message

The case is considered when the message $y(x)$ and the key function $f(x)$ are continuous, smooth, having a first derivative.

The transmitting and receiving parties agree on the key function, the initial value of the argument x_0, the step of its change h and the initial value of the transmitted message $y(x_0)$.

In practice, most often, there is a need to encrypt signals from moving objects.

In this case, the signal is represented by a function of time $y(t)$. To encrypt it using the key function $f(t)$, disproportion (3) should be calculated, which in this case has the form:

$$z(t) = @\, d^{(1)}_{f(t)} y(t) = \frac{y(t)}{f(t)} - \frac{dy/dt}{df/dt} \tag{8}$$

In this case the time change step h is selected from the representation condition $y(t)$ with the necessary accuracy.

The decrypted values are calculated from equation (9) obtained from (8):

$$y'(t) = f'(t)\left(\frac{y(t)}{f(t)} - z(t)\right) \tag{9}$$

The decrypted message is found from (9) by solving the Cauchy problem using the known $y(t_0)$, h and $z(t)$ values obtained via the communication channel (8).

Example 3. Analog message encryption

For simulation, suppose that an analog message is described by the following expression:

$$y(t) = k \cdot \exp((a - t)\cos(bt))\sin^2(bx) \tag{10}$$

The key function has the form:

$$f(t) = a \cdot \exp(bt) + \sin(bt) + \ln(at + b)\,, \tag{11}$$

where $a = 1$, $b = 10$, $k = 100$ are constants.

The time t varies from 0.1 to 4 in increments of $h = 0.0025$.

The derivatives were calculated numerically using the values of three points.

The results of the cryptosystem simulation are shown in table 5.

Table 5. Transmitted and decrypted values of a numerical function.

account number	transmitted value	decrypted value
1	0,16938	0,173057
2	0,674778	0,68059
3	1,51054	1,51856
4	2,66904	2,67926
5	4,14073	4,15308
6	5,91425	5,93181
7	7,97656	7,99589
8	10,313	10,3338
9	12,9076	12,9296
10	15,743	15,7656
11	18,8006	18,8234
12	22,061	22,0834
13	25,5038	25,5252
14	29,1082	29,1278
15	32,8525	32,8735
16	36,715	36,7328
17	40,6737	40,6874
18	44,7066	44,7188
19	48,7916	48,7979
20	52,9072	52,9066
21	57,0322	57,0268
22	61,1458	61,1315
23	65,228	65,2037
24	69,2595	69,2267
25	73,2218	73,1769
26	77,0975	77,0394
27	80,8701	80,7977
28	84,5241	84,4363
29	88,0453	87,9424
30	91,4207	91,3003

Obtained results indicate that even with a rough calculation of the derivatives, the decryption error in this example is a fraction of a percent. This error can be reduced by decreasing the step h, and by applying more accurate methods of numerical differentiation, for example, the Newton-Stirling method [21].

10 Constraints on the key-function of a real variable

When implementing the proposed cryptosystem, it is necessary to take into account the restrictions imposed on the key function.

1. The function must be defined on the set of real numbers.
2. The function should not be constant and not take zero values.
3. When using the key function, a situation should not arise when the number is divided by a number close to zero, which leads to the appearance of an unacceptable calculation error. To this end, it is recommended to test the cryptosystem for the entire alphabet of characters that will be used in messages.
4. Before sending an encrypted message, first check what it looks like after decryption. This will avoid mistakes that may occur because of not taking into account the previous paragraphs.

It should also be noted that in the system there is an effect similar to the "avalanche effect" in the AES cryptosystem. It consists in the fact that behind an incorrectly decrypted character the remaining message is decrypted incorrectly.

11 Conclusions

A cryptosystem with a symmetric key is offered. For the first time, a function of a real variable is used as a key, which satisfies some restrictions. It can be either continuous or discrete. Encryption is performed using disproportion functions. The specific examples show the possibility of encryption and decryption of text information, 2D-graphic images, as well as analog messages. Examples are also given that demonstrate the cryptographic strength of the proposed cryptosystem.
In the practical application of the proposed cryptosystem, the above limitations on the key function should be taken into account.

References

1. Federal Information Processing Standards Publication 197 November 26, 2001 Specification for the ADVANCED ENCRYPTION STANDARD (AES).
2. GOST 28147-89. Sistemy` obrabotki informaczii. Zashhita kriptograficheskaya. Algoritm kriptograficheskogo preobrazovaniya. – Vved. 01.01.90.
3. Lebedev, A.N.: Kriptografiya s «otkry`ty`m klyuchom» i vozmozhnosti ee prakticheskogo primeneniya. Zashhita informaczii, vol. 2 (1992).
4. Rivest, R., Shamir, A., Adleman, I.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, vol. 21(2), pp. 120-126. (1978). doi:10.1145/359340.359342.
5. Gorbenko, I.D., Gorbenko, Yu.I.: Prykladna kryptologiya. Pidruchnyk. KhNURE, Fort, 878 p. Kharkiv (2012).
6. Hankerson, D.R., Vanstone, S.A., Menezes, A.J.: Guide to elliptic curve cryptography. XX, 311p. Springer, New York (2003).
7. Ladd, T. D., Jelezko, F., Laflamme, R., Nakamura, Y., Monroe, C., O'Brien, J. L.: Quantum Computing, Nature, vol. 464, pp. 45-53 (2010).
8. Klyucharev, P.G.: Kvantovy`j komp`yuter i kriptograficheskaya stojkost` sovremenny`kh sistem shifrovaniya. Vestnik MGEN im. N.E`.Baumana, ser. «Estestvenny`e nauki», vol. 2 (2007).

9. Grover, I.K.: Quantum Mechanics Help in Searching for a Needle in a Haystack. Phys. Rev. Lett., vol. 78(2), pp. 326-328 (1997).
10. Shor, P.: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. Prceedings of the 35th Annual Sympotion of Foundations of Computer Science (1994).
11. Proos, J.A.: Shor's discrete logarithm quantum algorithm for elliptic curves. p. 35. Waterloo. Ont: Faculty of Mathematics University of Waterloo (2003).
12. Kolmogorov, A.N., Fomin, S.V.: E`lementy` teorii funkczij i funkczional`nogo analiza. Nauka, Moscow (1972).
13. Avramenko, V.V., Zabolotny, M.I.: A Way of Data Coding. Patent UA H041. 9/00 №42957, Ukraine (2009).
14. Avramenko, V.V.: Characteristic properties of disproportionality functions and their application to solving diagnoses problems. Transactions of Sumy State University (SumDU), vol. 16, pp. 24-28, Ukraine (2000).
15. Kalashnikov, V.V., Avramenko, V.V., Kalashnykova, N.I.: 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). Chapter 7.
16. Kalashnikov, V.V., Avramenko, V.V., Slipushko, N.Yu., Kalashnykova, N.I., Konoplyanchenko, A.E.: Identification of quasi-stationary dynamic objects with the use of derivative disproportion functions. 108(C): 2100–2109. Procedia Comput. Sci. (2017).
17. Kalashnikov, V.V., Avramenko, V.V., Kalashnikova, N.I., Kalashnikov-Jr., V.V.: A cryptosystem based on sums of key functions. International Journal of Combinatiorial Optimization Problems and Informatics, vol. 8, No.1, pp. 31-38 (2017).
18. Kalashnikova, N.I., Avramenko, V.V., Kalashnikov, V.V.: Sums of Key Functions Generating Cryptosystems. In: J. M. F. Rodrigues et al. (Eds.): Computational Science – ICCS 2019. ICCS 2019, Chapter 23, Lecture Notes in Computer Science, vol. 11540, pp. 293-302. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22750-0_23 .ISBN 978-3-030-22749-4; Online ISBN 978-3-030-22750-0
19. Karpenko, A.P.: Integral'nye harakteristiki neproporcional'nosti chislovyh funkcij i ih primenenie v diagnostike. Vestnik SumDU, vol. 16, pp. 20-25, Ukraine (2000).
20. Avramenko, V., Moskalenko, A.: Operative Recognition of Standard Signals in the Presence of Interference with Unknown Characteristics, Proceedings of the Second International Workshop on Computer Modeling and Intelligent Systems (CMIS-2019), Zaporizhzhia, Ukraine, April 15-19 (2019).
21. Rao, S. B.: Numerical Methods: With Program in Basic, Fortran, Pascal & C++. Hyderabad, Universities Press, (2004).