UDC: 004.056.55 https://doi.org/10.15407/jai2024.02.010

V. Avramenko[1], M. Bondarenko[2]
[1,2]Sumy State University, Ukraine
116, Kharkivska st., Sumy, 40000
[1]vv.avramenko@cs.sumdu.edu.ua
[2]nikbond97@gmail.com
[1]https://orcid.org/0000-0002-6317-6711
[2]https://orcid.org/0000-0002-8849-7378

ENCRYPTION OF MESSAGES BY THE SUM OF A REAL VARIABLE FUNCTIONS

Abstract. The article proposes a cryptosystem with symmetric keys, where the keys are functions of a real variable. These functions can be either continuous or discrete and must satisfy certain constraints. The number of key functions is determined by the number of binary digits that encode a character in the ASCII table. Each binary digit has its own key function. The cipher of a character is represented by a one-dimensional array of real numbers. These numbers are obtained by summing the key functions, which correspond to "1" in the ASCII code of the character. The amplitudes of these functions are random and unknown to the receiving party. Decryption is a multi-level process, in which integral disproportion functions are calculated at each level. To increase the cryptographic strength, the encryption/decryption process involves a permutation of the key-functions according to a secret scheme agreed upon by both parties. Computer simulation has demonstrated the high cryptographic resistance of the proposed system to the determination of the coefficients within the key functions, as well as to the rearrangement of the key functions themselves. It is shown that adjacent identical symbols in an encrypted message have different ciphers, which also complicates hacking the system.

Keywords: cryptosystems, disproportion functions, functions of real variables, key functions, encryption, decryption, text messages.

Introduction

Currently, symmetric and asymmetric cryptosystems are widely used. The most well-known symmetric systems are AES [1] and GOST 28147-89 [2]. Asymmetric systems employ algorithms such as RSA and ElGamal [3]. Both types of systems are based on a set of integers. This fact allows for various methods of cryptanalysis, including straightforward key-guessing approaches. The complexity of a brute-force attack is estimated as $O(2^k)$, where k is the key length in bits.

For breaking asymmetric cryptosystems, there are cryptanalysis methods that operate faster than brute force, necessitating the use of longer keys compared to those in symmetric systems. To enhance the strength of cryptosystems, it is essential to continually increase key lengths. However, this approach may not be sustainable due to the continuous advancement in computing capabilities. Particularly, the rapid development of quantum computers [4] is expected to significantly affect the resilience of existing cryptosystems [5]. This is evident from the example of recovering the key of a symmetric encryption algorithm from the plaintext and ciphertext. Grover's quantum algorithm reduces the complexity by half [6]. As a result, the effectiveness of the key length is reduced by 2 times.

The application of quantum algorithms will also reduce the stability of asymmetric systems. Indeed, the RSA algorithm is based on the computational complexity of the integer factorization problem. At the same time, there is a quantum algorithm whose complexity is polynomial $O(n^3)$ [7]. Also, the stability of asymmetric systems can be reduced as a result of the implementation of Shor's quantum algorithm for calculating the discrete logarithm. In [8], Shor's algorithm is presented for a group of points of an elliptic curve over the field GF(p), with complexity $O(n^3)$.

From the above analysis, it is clear that we should look for alternative ways to create cryptosystems. In particular, in order to complicate the selection of keys using a simple brute force method, it is proposed to switch from using integers to real numbers. It is known [9] that the set of real numbers has greater cardinality compared to the set of natural numbers, thus it can be expected that

the security 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 [10-13, 15]. This work proposes a cryptosystem variant that uses the sum of several key functions of a real variable. In contrast to [15], to increase cryptographic strength, the permutation of key functions during the encryption and decryption process is considered.

Problem Statement

The objective is to develop encryption and decryption algorithms that utilize multiple keys simultaneously. These keys are to be functions of a real variable. To increase the cryptographic strength of the system, it is essential to incorporate a mechanism that permutes the positions of the key functions during both the encryption and the decryption processes.

Literature Review

The most known symmetric system AES implements a Substitution Permutation Network (SPN) [1]. Based on the Feistel network, the symmetric encryption algorithm GOST 28147-89 was developed [2, 3]. In 1978, the public key algorithm RSA was proposed [3]. Beyond cryptosystems that use integers as keys, other approaches to cryptosystem creation are known. For instance, in [14, Zhytomyr], a Fredholm integral cryptosystem of the first kind is proposed, illustrating that the essence of the encryption and decryption procedures is reduced to solving the direct and inverse problem described by the first kind Fredholm integral equations. Systems based on functions of real variables are also known [10-13]. For example, encryption of ASCII code table characters using the sum of 10 functions of real variables is considered in [10]. The sum of the function-key values obtained during the encryption of a character is transmitted over the communication channel. At the receiving end, fragments of the function-keys present in the received encrypted signal are recognized using disproportion functions [16-19]. Based on the recognition results, the symbol being transmitted at that moment is decrypted.

In [11, 12], the transmission of binary codes using three function-keys of real variables is examined. "1", "0", "space", and "newline" are encoded, with any other symbol recognized as a newline. To gain unauthorized access to an intercepted message, one must determine the form and parameters of the key-functions.

In [13], a principle of encryption using only one function of a real variable is proposed. The text to be encrypted is represented as a process of sequential transmission of numerical character codes. A disproportion function of the numerical representation of the encrypting process by the key function is calculated. The obtained disproportion function values constitute the encrypted message and are transmitted over the communication channel. In [15], attempts were made to determine the parameters of the key-functions, assuming their form is known. The results of computer modeling [10-13] showed high cryptographic resilience of these systems.

Mathematical Formulation of the Problem

The encrypted message consists of a sequence of T numerical codes of characters from the ASCII table. Each code is associated with a one-dimensional array of N elements. These arrays are generated by tabulating with a consistent step h in the argument change of m key-functions of a real variable. Each binary digit of the character code corresponds to its own key-function. The number m depends on the bit length of the code. The value of the element y(j,i) in the matrix y(T,N) is given as:

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

where j is the index of the character in the transmitted message;

$f_q(i)$ are the array values of the q-th key-function. Here $f_q(i) = f_q(ih)$, where i = 1,2,...N, q=1,2,...m, and N > m;

k_{qi} are the coefficients generated during the encryption of the j-th element.

The functions corresponding to a "1" in the character code are included in sum (1) with random coefficients. These coefficients are unknown to the recipient. For functions corresponding to a "0", the coefficients are equal to zero.

The key-functions can be either continuous or discrete. They must be identical for both the transmitting and receiving parties and have the same numbering. To increase cryptographic strength, a secret scheme for automatically changing this numbering should be implemented directly during the encryption or decryption process. Additionally, the step h of the argument change for the key-functions must be the same on both the transmitting and receiving sides.

The encrypted message is represented by a matrix y(T,N), whose elements are sequentially transmitted over an open communication channel. The task involves decrypting the message from the received matrix. To achieve this, the receiving side utilizes an integral disproportion function of the first order [20] to identify fragments of the key-functions present in the encrypted signal. If a fragment of a key-function is detected in the received signal, the corresponding bit is set to "1". For bits where key-functions are absent, the bit is set to "0". In this way, the transmitted symbol is decrypted.

Disproportion Functions

The following is a brief overview of disproportion functions. There are several disproportion functions. Let's consider one of them - the n-th order disproportion of the function y(x) with respect to x [17]. It is described by the expression:

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

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

For the case when n=1,

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

If functions x(t) and y(t) depending on the parameter t are considered, then disproportion (3) is described by the expression:

$$@\,d_{x(t)}^{(1)}y(t) = \frac{y(t)}{x(t)} - \frac{dy/dt}{dx/dt}, \qquad (4)$$

For y(t) = kx(t), the disproportion (4) is equal to zero in the entire area of existence x(t) regardless of the value of k.

In practice, problems often arise when y(x) has the form:

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

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

It has been shown that disproportionality functions make it possible to calculate the values of unknown coefficients in (5) using data obtained for the current value of the argument [17]. This feature is used to decrypt messages [13-15].

Functions are often considered for which the first derivative does not exist or equals zero over some interval. This precludes the use of disproportion based on derivatives in (2-4). In such cases, the first-order integral disproportion can be employed [20]. This disproportion of the function y(x) with respect to f(x) is described by the following expression:

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

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

The functions y(x) and f(x) are represented by the elements of one-dimensional arrays x(ih) and y(ih), where i =

0, 1, 2, ..., N. If the approximate values of the integrals in equation (6) are calculated using the trapezoidal rule, then the disproportion (6) takes the following form:

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

Encryption and Decryption of Messages

Initially, it is essential to ensure that both the transmitting and receiving sides possess the same set of m key-functions of a real variable and their numbering. The interval for changing the argument of these functions is determined by the step h of its change and the number of elements N>m of the one-dimensional array corresponding to the encrypted symbol. The step h is defined by the requirements for the accuracy of reproducing the key-functions. This ensures that all conditions are met for the arrays of key function values to be identical for both parties.

Message Encryption

1. Calculate the elements of arrays from N > m values of key functions $f_q(x)$, where q = 1, 2, …, m.

2. Input the j-th character to be encrypted and calculate its cipher as values of the one-dimensional array y(j, i), where i =1, 2, …, N, following equation (1). Repeat this step for all characters in the message of length T. During this process, the key functions are permuted according to a specific scheme based on the number of cycles.

3. The encrypted message, consisting of a sequence of T arrays, is transmitted over an open communication channel.

Message Decryption

Calculate the arrays $f_q(i) = f_q(ih)$, where i = 1, 2, ... N (>m), and q = 1, 2, ... m, for the key functions and receive T one-dimensional arrays y(j, i) via a communication channel, where j = 1, 2, ... T, and i = 1, 2, ... N.

Let's consider the decryption process using an example in which only four key functions are used in the cryptosystem: $f_1(x)$, $f_2(x)$, $f_3(x)$, $f_4(x)$, meaning that m = 4. Consequently, the cipher for the j-th character appears as follows:

$$y(j,i) = k_{1j} f_1(i) + k_{2j} f_2(i) + k_{3j} f_3(i) + k_{4j} f_4(i), \qquad (8)$$

where i =1, 2, …, N > 4.

To decrypt it, it is necessary to determine the values of the unknown coefficients k_1, k_2, …, k_m in equation (8). The process consists of m = 4 levels, corresponding to the number of key functions. At each level, the disproportion (7) $F_{l,n}(j,i)$ is calculated, where l is the level, and n is the index of the disproportion calculation at that level.

First Level

It is necessary to calculate the disproportion (7) of y(j, i) using any of the key functions, for example, using $f_1(i)$:

$$F_{1,1}(j,i-1) = @I_{f_1(i)}^{(1)} y(j,i) = \frac{y(j,i-1) + y(j,i)}{f_1(i-1) + f_1(i)} - \frac{y(j,i)}{f_1(i)}, \qquad (9)$$

where j = 1, ... N.

Also, calculate the disproportions (7) for the remaining key functions in equation (8) with respect to $f_1(i)$:

$$F_{1,r}(j,i) = @I_{f_1(i)}^{(1)} f_r(j,i) = \frac{f_r(j,i-1) + f_r(j,i)}{f_1(i-1) + f_1(i)} - \frac{f_r(j,i)}{f_1(i)}, \qquad (10)$$

where j = 1, …N; r = 2, 3, 4.

By substituting y(j,i) from equation (8) into equation (9) and considering that the disproportion (7) of the function $f_1(i)$ with respect to itself is zero, we obtain:

$$F_{1,1}(j,i) = k_{2j}F_{1,2}(j,i) + k_{3j}F_{1,3}(j,i) + k_{4j}F_{1,4}(j,i), \quad (11)$$

Second Level

From the right-hand side of equation (11), select any component, for example, $F_{1,2}(j,i)$. Based on this, calculate the disproportions:

$$F_{2,1}(j,i) = @I^{(1)}_{F_{1,2}}(j,i)F_{1,1}(j,i) = \frac{F_{1,1}(j,i-1) + F_{1,1}(j,i)}{F_{1,2}(j,i-1) + F_{1,2}(j,i)} - \frac{F_{1,1}(j,i)}{F_{1,2}(i)}, \quad (12)$$

$$F_{2,r}(j,i) = @I^{(1)}_{F_{1,2}}(j,i)F_{1,r}(j,i) = \frac{F_{1,r}(j,i-1) + F_{1,r}(j,i)}{F_{1,2}(j,i-1) + F_{1,2}(j,i)} - \frac{F_{1,r}(j,i)}{F_{1,2}(i)}, \quad (13)$$

where $j = 1, \ldots N$; $r = 3, 4$.

Considering that the disproportion of $F_{1,2}(j,i)$ with respect to $F_{1,2}(j,i)$ is zero:

$$F_{2,1}(j,i) = k_{3j}F_{2,3}(j,i) + k_{4j}F_{2,4}(j,i), \quad (14)$$

Third Level

From the right-hand side of equation (14), select $F_{2,3}(j,i)$. Calculate the disproportions:

$$F_{3,1}(j,i) = @I^{(1)}_{F_{2,3(j,i)}}F_{2,1}(j,i) = \frac{F_{2,1}(j,i-1) + F_{2,1}(j,i)}{F_{2,3}(j,i-1) + F_{2,3}(j,i)} - \frac{F_{2,1}(j,i)}{F_{2,3}(i)}, \quad (15)$$

$$F_{3,2}(j,i) = @I^{(1)}_{F_{2,3(j,i)}}F_{2,4}(j,i) = \frac{F_{2,4}(j,i-1) + F_{2,4}(j,i)}{F_{2,3}(j,i-1) + F_{2,3}(j,i)} - \frac{F_{2,4}(j,i)}{F_{2,3}(i)}, \quad (16)$$

Given that the disproportion of $F_{2,3}(j,i)$ with respect to itself is zero:

$$F_{3,1}(j,i) = k_{4j}F_{3,2}(j,i), \quad (17)$$

Fourth Level

Calculate the disproportion (7) of $F_{3,1}(j,i)$ with respect to $F_{3,2}(j,i)$:

$$F_{4,1}(j,i) = @I^{(1)}_{F_{3,2(j,i)}}F_{3,1}(j,i) = k_{4j}\left(\frac{F_{3,2}(j,i-1) + F_{3,2}(j,i)}{F_{3,2}(j,i-1) + F_{3,2}(j,i)} - \frac{F_{3,2}(j,i)}{F_{3,2}(i)}\right) = 0, \quad (18)$$

The reason for equation (18) equating to zero is that, as evident from equation (17), there exists a proportional relationship between $F_{3,1}(j,i)$ and $F_{3,2}(j,i)$.

From equation (17), find k_{4j}:

$$k_{4j} = \frac{F_{3,1}(j,i)}{F_{3,2}(j,i)}, \quad (19)$$

From equations (14), (11), and (8), calculate the remaining coefficients:

$$k_{3j} = \frac{F_{2,1}(j,i) - k_{4j}F_{2,4}(j,i)}{F_{2,3}(j,i)}, \quad (20)$$

$$k_{2j} = \frac{F_{1,1}(j,i) - k_{3j}F_{1,3}(j,i) - k_{4j}F_{1,4}(j,i)}{F_{1,2}(j,i)}, \quad (21)$$

$$k_{1j} = \frac{y(j,i) - k_{2j}f_2(i) - k_{3j}f_3(i) - k_{4j}f_4(i)}{f_1(i)}, \quad (22)$$

Decryption of the j-th character in the message depends on which of these coefficients are non-zero and which are zero. If a coefficient associated with a key function is non-zero, the corresponding bit in the character code is set to "1." Bits for which the coefficients are zero are set to "0." In this way, the decryption of the character currently being transmitted is carried out.

However, it is important to consider the presence of computational errors. Therefore, the disproportion calculated at the final level may not be exactly zero, but may differ from zero by a small number ε. For instance, if $|F_{4,1}(j,i)| < \varepsilon = 10^{-4}$, then it should be considered as zero. The specific value of ε can be determined during the testing of the cryptosystem.

It should also be noted that theoretically, the disproportion at the final level is zero for all i = 2, 3, ... N. In practice, taking into account calculation errors, it is recommended to do calculations using formulas (17-19) for the array element number i, at which the disproportion module (16) is minimal.

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

An eight-bit character code requires m=8 functions - keys:

$$\begin{aligned}
f_1(x) &= 1000(\alpha_1 \sin(\mathrm{pow}(\cos(w\beta_1 x),2)) + \alpha_2 \cos(\sin(\beta_2 x)) + \exp(\sin(\alpha_3 x + \cos(\beta_3 x^2)))),\\
f_2(x) &= 1000(\beta_3 \cos(w\beta_1 \sin(\alpha_2 x)) + \beta_2 w \sin(\exp(\alpha_3 \cos(\alpha_1 x)))),\\
f_3(x) &= 1000(\beta_3 \cos(\alpha_1 \exp(0.01\alpha_3 x)) + \beta_2 \sin(w\alpha_2 \sin(\beta_1 x)),\\
f_4(x) &= 1000(\alpha_2 \sin(w\beta_1 \exp(-0.03\alpha_3 x)))\alpha_1 \sin(\beta_2\beta_3 x))),\\
f_5(x) &= 1000((\alpha_2 - 10x)\beta_2 \cos(\sin(\beta_1 x)x) + \beta_3 w \sin(\alpha_1 x + \alpha_3))),\\
f_6(x) &= 1000((100 - \beta_3 x)\exp(\alpha_3 \sin(\alpha_1 x)) + (\beta_1 + 20x)\sin(w\beta_2 \cos(\alpha_2 x))),\\
f_7(x) &= 1000(\alpha_3 \sin(\beta_1 x)(\sin(\beta_2 x) + \cos(w\alpha_2 x)\beta_3(-\alpha x^2))),\\
f_8(x) &= 1000((\alpha_1 + \beta_3)\cos(\alpha_3 w \sin(\beta_1 x)\alpha_2 \cos(\beta_2 x))),
\end{aligned} \quad (23)$$

where $\alpha_1 = 10$, $\alpha_2 = 0.12$, $\alpha_3 = 0.5$, $\beta_1 = 0.1$, $\beta_2 = 12$, $\beta_3 = 0.7$, $w = 500$, - are constants.

In the example, the symbol is encoded by the sum of functions – keys

$$y(x) = k_1f_1(x) + k_2f_2(x) + k_3f_3(x) + k_4f_4(x) + k_5f_5(x) + k_6f_6(x) + k_7f_7(x) + k_8f_8(x), \quad (24)$$

where x = ih is an argument; h = 1 is a step of changing the argument; i is the serial number of the element of the one-dimensional array $y_0, y_1, \ldots, y_{N-1}$, as well as each of the functions - keys.

N is the number of elements of each one-dimensional array. Given the condition N > m, in the example, N is taken to be 16.

In Table 1, the transmitted characters are displayed in the top horizontal row. The corresponding ciphers are presented as vertically arranged arrays, $y_0, y_1, \ldots, y_{15}$. The

decrypted characters are located in the bottom horizontal row. Table 1 shows the results of encrypting and decrypting the word "Hello." The transmitted characters are shown again in the top horizontal row, with their corresponding ciphers displayed as vertically arranged arrays. The decrypted characters are positioned in the bottom row, horizontally.

Table 1. Encryption and decryption of the message “Hello”

y_i	H	e	l	l	o
y_0	-3462.92	-464930	-1.94948e06	-1.67087e06	79665.1
y_1	-27945.8	-1.93085e06	2.32983e+06	-618047	-57354.2
y_2	48274.2	49597.3	-10053.6	-540688	-1.33456e06
y_3	-63910.1	-95844.5	-111907	714190	1.39113e+06
y_4	4843.91	-521719	-283149	-621670	-1.41922e06
y_5	180555	-530555	-397695	237254	396756
y_6	-547757	-115632	-69730.9	291654	563131
y_7	1.12602e+06	-1.12165e06	-486727	-388156	-958982
y_8	-1.84834e06	-1.20104e06	-1.29669e06	506483	1.32904e+06
y_9	2.62386e+06	405547	1.3986e+06	-513376	-1.74415e06
y_{10}	-3.23145e06	1.54594e06	-1.00874e06	-179453	794323
y_{11}	3.49927e+06	1.96214e+06	3.86163e+06	804064	724870
y_{12}	-3.20739e06	-1.57378e06	-4.31761e06	-637326	-368713
y_{13}	2.21232e+06	-820759	3.096e+06	1.40039e+06	1.05473e+06
y_{14}	-3462.92	2.23014e+06	-2.02298e06	-1.81204e06	-1.81368e06
y_{15}	-27945.8	-136888	2.17465e+06	774521	824983
y_i	H	e	l	l	o

The decrypted characters are the same as the encrypted ones. It is also clear that the ciphers of adjacent ‘l’ symbols differ from each other. In order to demonstrate that such a difference is natural, Table 2 shows the results of encryption/decryption of a sequence of identical symbols.

Table 2. Encryption and decryption of the five identical characters ‘A’

y_i	A	A	A	A	A
y_0	44355.1	79061.9	50605.3	4.27791e+06	-358.087
y_1	14263.4	57186.5	37156.4	6.57533e+06	-5675.86
y_2	65693.3	52062.4	32191.3	6.13868e+06	-9857.87
y_3	-34604.5	30122.3	20878.8	4.57351e+06	-11894.1
y_4	16899.4	16660.3	10429.4	6.95281e+06	-7584.02
y_5	152282	17372.7	6696.34	4.03545e+06	4207.17
y_6	-451022	-20600	452.905	6.8154e+06	8004.11
y_7	946393	104883	39588.5	5.65589e+06	-10091.4
y_8	-1.51504e06	-92083	-13525.1	5.0007e+06	2564.52
y_9	2.20616e+06	246139	93365.8	6.97047e+06	5317.41
y_{10}	-2.66021e06	-170969	-29852.3	3.98658e+06	2912.24
y_{11}	2.95031e+06	329825	125294	6.968e+06	-1618.25
y_{12}	-2.62671e06	-159244	-23183.1	5.14571e+06	16031.9
y_{13}	1.87308e+06	227270	91296.6	5.46946e+06	10967.3
y_{14}	-360135	16120.2	21775.1	6.89191e+06	1641.11
y_{15}	-1.58311e06	-109592	-22924.5	4.02238e+06	-1348.6
y_i	A	A	A	A	A

The results indicate that the ciphers of identical symbols adjacent in a message differ from each other. Each time, the same symbol receives a completely different cipher. This significantly complicates the "cracking" of the cryptosystem. Additionally, to break the cryptosystem, it is not only necessary to somehow discover the expressions for the eight key functions but also to determine the values of the coefficients they contain. Below is an example illustrating the difficulty of determining these coefficients. Table 3 shows

the decryption results when the value of one of the seven coefficients is incorrectly selected. Instead of w = 500, the value 499.9995 was selected during decryption.

Table 3. Comparison of the decryption with correct and incorrect coefficient w

Original message	Decrypted message with a coefficient w = 500	Decrypted message with a wrong coefficient w = 499.9995
T	T	...
h	h	l
e	e	...
(Whitespace)	(Whitespace)	...
i	i	i
m	m	l
p	p	s
o	o	o
r	r	r
t	t	w
a	a	a
n	n	n
t	t	w
(Whitespace)	(Whitespace)	(Whitespace)
m	m	э
e	e	...
s	s	...
s	s	...
a	a	...
g	g	...
e	e	э

When w = 500, the decryption result matches the encrypted message. However, even a deviation of 0.0005 makes decryption impossible. And this is just one of several coefficients. To further complicate the cracking of the cryptosystem, it is also proposed to perform a permutation of the key functions according to a specific scheme, which must be implemented by both parties and serves as an additional element of secrecy.

The example involves a scheme of six permutations of key functions. Each permutation occurs after a certain number of encryption/decryption cycles. The number of cycles can be either consistent or vary. The scenario is modeled where a third party has become aware of the key functions, which they decided to use to intercept the message, however, the scheme of their permutations remained unknown. Table 4 shows the decryption results when encryption was performed with permutations and decryption was conducted without them.

Table 4. Comparison of the decryption with and without permutations of functions

Original message	Decrypted with permutations of functions	Decrypted without permutations of functions
A	A	A
B	B	B
C	C	C
D	D	€
E	E	(Whitespace)
F	F	(Whitespace)
G	G	…
H	H	(Whitespace)
I	I	%

J	J	%
K	K	-
L	L	1
M	L	1
N	N	5
O	O	v
P	P	…
Q	Q	)
R	R	(Whitespace)
S	S	9
T	T	I
U	U	Y
V	V	Y
W	W	№
X	X	I
Y	Y	Y
Z	Z	‡

The results indicate that in this case the secret scheme of permutations of key functions does not allow reading the encrypted message.

Limitations on Key Functions

In [15], the main requirements for key functions are outlined:

1. The function must be defined over the set of real numbers.

2. The function should not be constant and must not assume zero values.

3. When using a key function, avoid situations where division by a number close to zero occurs, as this can lead to significant computational errors. To prevent this, it is advisable to thoroughly test the cryptosystem with all the characters that will be used in the messages.

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

5. It is recommended to include all parameters in the expression for each key function. In this case, changing the value of any parameter leads to a change in all key functions, rather than just one or a few.

6. Before sending an encrypted message, first check how the decrypted message appears to avoid errors that might arise from not adhering to the previous guidelines.

Conclusions

Encryption and decryption algorithms have been developed using functions of a real variable as keys. To increase cryptographic strength, a permutation of key functions is carried out during the encryption/decryption process. Computer simulation has shown that this permutation works effectively. The lack of information about the permutation prevents unauthorized decryption of messages, even when the key functions are known.

Computer simulation has also demonstrated how difficult it is to determine the values of the coefficients within the key functions, not to mention the need to identify the form of each one. The complexity of cracking the system is further increased because identical symbols do not produce identical ciphers.

References

1. FIPS. (2001). Specification for the ADVANCED ENCRYPTION STANDARD (AES). *In Federal Information Processing Standards Publication*. https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf

2. Courtois, N. T. (2012). Security Evaluation of GOST 28147-89 in *View of International Standardisation. In Cryptologia (Vol. 36, Issue 1).* https://doi.org/10.1080/01611194.2011.632807

3. Rivest, R. L., Shamir, A., Adleman, L. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2). https://doi.org/10.1145/359340.359342

4. Ladd, T. D., Jelezko, F., Laflamme, R., Nakamura, Y., Monroe, C., O'Brien, J. L. (2010). Quantum computers. *Nature*, 464(7285), 45–53. https://doi.org/10.1038/nature08812

5. Ostrianska, Y. V., Yesina, M. V., Gorbenko, I. D. (2022). Analysis of views of the European Union on quantum-post-quantum limitations. *Radiotekhnika*, 210. https://doi.org/10.30837/rt.2022.3.210.06

6. Grover, L. K. (1997). Quantum mechanics helps in searching for a needle in a haystack. *Physical Review Letters*, 79(2). https://doi.org/10.1103/PhysRevLett.79.325

7. Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5). https://doi.org/10.1137/S0097539795293172

8. Proos, J., Zalka, C. (2003). Shor's discrete logarithm quantum algorithm for elliptic curves. *Quantum Information and Computation*, 3(4). https://doi.org/10.26421/qic3.4-3

9. Bagaria, J. (2010). Set theory. *In The Princeton Companion to Mathematics*. https://doi.org/10.4324/9781315167749-28

10. Avramenko, V. V., Zabolotnyy, M. I. (2009). Sposib shyfruvannya danykh (Pat. 42957 U Ukrayina, MPK6 H 04 L 9/00.). *Derzhavne pidpryyemstvo "Ukrayins'kyy instytut promyslovoyi vlasnosti" (UKRPATENT)*. https://essuir.sumdu.edu.ua/bitstream-download/123456789/9879/1/getdocument.pdf

11. Kalashnikov, V. V., Avramenko, V. V., Kalashnykova, N. I., Kalashnikov, V. V. (2017). A Cryptosystem Based Upon Sums of Key Functions. *International Journal of Combinatorial Optimization Problems and Informatics*, 8(1), 31–38. https://ijcopi.org/ojs/article/view/5

12. Kalashnykova, N., Avramenko, V., Kalashnikov, V. (2019). Sums of Key Functions Generating Cryptosystems (pp. 293–302). https://doi.org/10.1007/978-3-030-22750-0_23

13. Avramenko, V., Demianenko, V. (2020). Cryptosystem based on a key function of a real variable. *CEUR Workshop Proceedings*, 2608. https://doi.org/10.46932/sfjdv2n2-113

14. Hryshchuk, R., Hryshchuk, O. (2019). A GENERALIZED MODEL OF FREDHOLM'S CRYPTOSYSTEM. *Cybersecurity: Education Science Technique*, 4, 14–23. https://doi.org/10.28925/2663-4023.2019.4.1423

15. Avramenko, V., Bondarenko, M. (2021). Using the Sum of Real Type Functions to Encrypt Messages. *CEUR Workshop Proceedings*, 3200.

16. Avramenko, V. V. (2000). Harakteristiki neproporcional'nosti chislovyh funkcij i ih primenenie pri reshenii zadach diagnostiki. *Visnyk Sums'koho Derzhavnoho Universytetu*, 16, 12–20. https://essuir.sumdu.edu.ua/bitstream-download/123456789/1824/1/5201C993d01.pdf

17. Avramenko, V. V., Kalashnykova, N. I., Kalashnikov, V. V., Watada, J. (2023). Derivative of disproportion functions for pattern recognition. *In Unconventional Methods for Geoscience, Shale Gas and Petroleum in the 21st Century*. https://doi.org/10.3233/AERD230022

18. Kalashnikov, V. V., Avramenko, V. V., Slipushko, N. Y., Kalashnykova, N. I., Konoplyanchenko, A. E. (2017). Identification of Quasi-Stationary Dynamic Objects with the Use of Derivative Disproportion Functions. *Procedia Computer Science*, 108. https://doi.org/10.1016/j.procs.2017.05.266

19. Avramenko, V., Moskalenko, A. (2019). Operative recognition of standard signals in the presence of interference with unknown characteristics. *CEUR Workshop Proceedings*, 2353. https://doi.org/10.32782/cmis/2353-5

20. Karpenko, A. P. (2000). Integral'nye kharakteristiki neproportsional'nosti chislovykh funktsii i ikh primenenie v diagnostike. *Visnyk Sums'koho Derzhavnoho Universytetu*, 16, 20–25. https://essuir.sumdu.edu.ua/bitstream-download/123456789/10931/1/4_Karpenko.pdf

The article has been sent to the editors 27.05.24.
After processing 15.06.24.
Submitted for printing 28.06.24.