Sums of Key Functions Generating Cryptosystems[1]

Nataliya Kalashnykova[1][0000-0002-2314-3106], Viktor V. Avramenko[4][0000-0002-9269-2432], and Viacheslav Kalashnikov[2,3,4][0000-0001-6747-580X]

[1]Universidad Autónoma de Nuevo León (UANL), av. Universidad S/N, San Nicolás de los Garza NL 66455, Mexico
[2]Tecnológico de Monterrey (ITESM), av. Eugenio Garza Sada 2501 Sur, Monterrey NL 64849, Mexico
[3]Central Economics and Mathematics Institute (CEMI), Nakhimovsky pr. 47, Moscow 117418, Russia
[4]Sumy State University (SumDU), Rimsky-Korsakov str. 2, Sumy 40007, Ukraine
kalash@tec.mx

Abstract. The paper develops an algorithm based on derivative disproportion functions (DDF) for modeling a cryptosystem for transmitting and receiving devices. The transmitted symbols are encoded with the aid of sums of at least two of those functions weighted with random coefficients. Some important properties of the derivative disproportion functions are also discussed. Numerical experiments demonstrate that the algorithm is quite reliable and robust.

Keywords: Cryptosystems, Derivative Disproportion Functions (DDF), Decoding Algorithms.

1 Introduction

In the modern competitive world, the significance and price of information are steadily growing up. Hence, the information is often encrypted when transmitted by various ways from a transmitter to a receiver in order to prevent it from unauthorized access. The latter necessity strongly ushers one to the use of cryptographic techniques within information systems, the most popular of which are the Data Encryption Standard (DES) [1], Advanced Encryption Standard (AES) [2], and the Rivest-Shamir-Adleman (RSA) cryptosystem [3]. However, the new super-computers and the technologies of network and neural computing that have appeared in the 2000th, conduct to the reevaluation of the previous cryptographic systems that had been thought to be highly reliable. Because of that, the development of new principles for the generation of new cryptosystems is very reasonable.

At present, the majority of cryptosystems exploit integer numbers for their keys. Moreover, the longer the key, the more hacking-proof is the cryptosystem because it becomes more difficult to fit the key by solving an appropriate factorization problem. The shift from integers to reals, or even more, to real functions makes the task of code

[1] This research was financially supported by the SEP-CONACYT Grants CB-2013-01-221676 and FC-2016-01-1938.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_23

breaking (hacking) much more complicated, which promises to enhance the cryptosystem's reliability (resistance property).

In this paper, we study such a new tool for classifying and declassifying both an analog signal and a signal in the form of a sequence of symbols from the specified alphabet [5]. Such a cryptosystem is built with the use of derivative disproportion functions (DDF) [5]–[6]. The input symbols are encrypted by the sum of real functions (keys) weighted with randomly selected coefficients. Owing to the derivative disproportion functions, one has a possibility of recognizing which functions had been involved to encrypt the signal. The latter permits the receiver to decode the encrypted symbols *even though* the randomly selected weights of the key functions are *unknown*. The latter fact allows the considered topic to fit well into the area of solving problems under *uncertainty*.

Moreover, the derivative disproportion functions (DDF) are accepted in a steadily growing list of areas of applications under uncertainty, such as the identification of quasi-stationary dynamic objects [7]-[8], and pattern recognition [9].

The rest of the paper is organized as follows. Sections 2 and 3 introduce the derivative disproportion functions (DDF) thus specifying the problem. The decoding algorithm is presented in Section 4, while Section 5 considers numerical examples and the results of numerical experiments. Sections 6 and 7 argue the proposed cryptosystem's robustness and the requirements to the disproportion key functions, respectively. The paper is finished with the conclusions (Section 8) and the list of references.

2 Derivative Disproportion Functions (DDF)

The brand-new ways of classifying information can be generated with the use of derivative disproportion functions (DDF). Disproportion functions related to the derivatives and to the values were developed and examined by the authors in the previous publications[5]–[6].

The derivative disproportion functions (DDF) are exercised in order to identify (label, tag) relevant real functions. The DDFs permit to quantitatively estimate the degree of a deviation of a numerical function from the specified functions (like, e.g., power function $y = k \cdot x^n$) for any fixed value of the argument, regardless of the associated parameters (like, for example, multiplier k for the power function). Here, $n \geq 1$is an integer.

Definition 1. The *derivative disproportion function (DDF) of order n* of the function $y = y(x)$ with respect to x ($x \neq 0$) is defined as follows:

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

In the particular case of $n = 1$ (order 1), Eq. (1) of the derivative disproportion is easily reduced to:

DOI: 10.1007/978-3-030-22750-0_23

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

As one could expect, for the linear function $y=kx$, its DDF of order 1 is *zero* for any value of the coefficient *k*. The symbol @ is chosen to designate the operation of determination of disproportion. The symbol “*d*” is selected to refer to the function’s derivative as the main object of disproportion calculated. Finally, the left-hand side of Eq. (2) reads “a*t d one y with respect to x*”.

If a function is reported in a parametric form, the *n*-th order derivative disproportion function (DDF) defined by Eq. (1) is determined by applying the rules of calculation of $d^n y/dx^n$ under the parametric dependence of *y* on *x*. In particular, the first-order derivative disproportion of the function defined parametrically as $y=\psi(t)$ and $x=\varphi(t)$ (where *t* is the parameter and $\varphi(t)\neq 0,\varphi'(t)\neq 0$ for all *t*) has the form

$$@d_x^{(1)}y=@d_{\varphi(t)}^{(1)}\psi(t)=\frac{y}{x}-\frac{y'_t}{x'_t}=\frac{\psi(t)}{\varphi(t)}-\frac{\psi'(t)}{\varphi'(t)}. \tag{3}$$

It is clear that if $\psi(t)=k\varphi(t)$ for some constant *k*, its derivative disproportion defined by Eq. (3) equals *zero* on the (shared) domain of the functions $y=\psi(t)$ and $x=\varphi(t)$.

Lemma 1. Each derivative disproportion function (DDF) of order *n* has the following properties*:*

1. Multiplying the function *y* by any scalar *m* results in multiplying its DDF by the same scalar.
2. The order *n* derivative disproportion function (DDF) of a sum (difference) of functions equals the sum (difference) of their DDFs.
3. For the linear function $y=kx$, its derivative disproportion of order 1 is zero for any value of the coefficient *k.*

Proof. It is readily verified by simple algebraic manipulations with the use of Definition 1.

Remark 1. In other words, the operator $@d_x^{(n)}$ defined on the space $C^n(\Omega)$ of *n* times continuously differentiable real functions is *linear* on this space.

3 The Problem’s Statement

Examine a communication system (channel) transmitting symbols (signals) encoded with a cryptosystem $\mathcal{K}$ based on the key functions $f_i=f_i(t)$, each defined on a (time) interval $t\in[0,T_i],T_i>0,i=1,\ldots,m$. The functions are assumed to be smooth and *n* times (continuously) differentiable. A symbol transmitted at the time moment *t* is encrypted with the sum of (at least two) key functions with possible time delays (shifts) $\tau_i \in [0, T_i], i = 1, \ldots, m.$

For example, if the transmitted symbol is encrypted by the (weighted) sum of two key functions f_p and $f_q, 1 \le p, q \le m,$ the signal transmitted through the communication channel has been encoded as

$$y(t) = k_p f_p(t + \tau_p) + k_q f_q(t + \tau_q), k_p > 0, k_q > 0. \tag{4}$$

We assume that an invader (intruder, hacker, etc.) who may have found unauthorized access to the channel is ***not aware*** of either the key functions f_i or their time delays (shifts) τ_i, or the coefficients $k_i, \ i = p, q.$

On the receiver's side of the communication system (channel), the complete list of key functions and their delays is known, but which of them (and with what weights) are involved in the received signal coded as in Eq. (4) is to be yet detected. The identification of these functions and their weights in Eq. (4) allows one to work out the received symbol $y(t)$.

The problem of identifying both the key functions and their weights in Eq. (4) is solved by the algorithm presented in the next section.

4 The Algorithm's Description

The problem in question is by no means easy to solve because the key functions and their weights can be detected only approximately (uncertainty environment). The received message $y(t)$ is unfolded in time, so the exact or approximate derivatives of this function are necessary. When one works with discrete data, e.g., $y(t) = (y(t_0), y(t_1), \ldots, y(t_{N-1}))^T$, then the desired approximate "derivative" of the (discrete) function $y(t)$ is estimated by a special approximation method, similar to that by Gregory-Newton (*cf.*, [4]).

The initial first version of our algorithm is technically quite burdensome, and due to the space restriction, we present here its description only for $m = 3$ (the complete version can be found in [6] and other publications of the authors).

The main idea of the general algorithm is as follows: if the key functions' delays (shifts) $\tau_i, i = 1 \ldots, m$, are known, we may represent the received message $y(t)$ as the sum of *all* key functions with (yet unknown) weights k_i :

$$y(t)=\sum_{i=1}^{m}k_i f_i\left(t+\tau_i\right). \tag{5}$$

Next, we have to identify these weights at the present moment t. The coefficients will be equal to zero for those functions that are ***not really*** involved in the encoded signal Eq. (5).

As we mentioned above, the algorithm will be described only for the case $m=3$. In general, the algorithm consists of m steps (that is, 3 in our case).

Step 1. Select randomly one of the key functions, for instance, the first one $f_1=f_1\left(t+\tau_1\right)$. By making use of Eq. (3), estimate the derivative disproportion function (DDF) value for the signal $y(t)$ and denote it as $F_{01}(t):=@d_{f_1}^{(1)}y(t)$. Besides, the DDF values $F_{21}(t)$ and $F_{31}(t)$ are calculated for the key functions $f_2\left(t+\tau_2\right)$ and $f_3\left(t+\tau_3\right)$ ***with respect to*** $f_1\left(t+\tau_1\right)$. Owing to the linearity of operator @ on the space $C^n\left(\Omega\right)$ (*see*, Remark 1), Eq. (5) yields (for $m=3$):

$$\begin{aligned}F_{01}(t)\equiv @d_{f_1}^{(1)}y(t)=\frac{y(t)}{f_1\left(t+\tau_1\right)}-\frac{y'(t)}{f'_1\left(t+\tau_1\right)}=k_1\cdot 0+k_2\left[\frac{f_2\left(t+\tau_2\right)}{f_1\left(t+\tau_1\right)}-\frac{f'_2\left(t+\tau_2\right)}{f'_1\left(t+\tau_1\right)}\right]+\\ +k_3\left[\frac{f_3\left(t+\tau_2\right)}{f_1\left(t+\tau_1\right)}-\frac{f'_3\left(t+\tau_2\right)}{f'_1\left(t+\tau_1\right)}\right]=k_2\,@d_{f_1}^{(1)}f_2\left(t+\tau_2\right)+k_3\,@d_{f_1}^{(1)}f_3\left(t+\tau_2\right)\equiv\\ \equiv k_2F_{21}(t)+k_3F_{31}(t).\end{aligned} \tag{6}$$

Here, the first term on the right-hand side of the upper line of Eq. (6) is zero due to Assertion 3 of Lemma 1.

Step 2. Again, pick up randomly one of the remaining DDFs $F_{21}(t)$ and $F_{31}(t)$; let it be, for instance, $F_{21}(t)$. Now, we calculate the derivative disproportions of the functions $F_{01}(t)$ and $F_{31}(t)$ with respect to $F_{21}(t)$; denote them as $F_{0121}(t)$ and $F_{3121}(t)$, respectively.

Applying the operator of the derivative disproportion of order 1 to both sides of Eq. (6), then making use of its linearity and Assertion 3 of Lemma 1, one easily comes to the equalities

$$F_{0121}(t)\equiv\frac{F_{01}(t)}{F_{21}(t)}-\frac{F'_{01}(t)}{F'_{21}(t)}=k_2\cdot 0+k_3\left[\frac{F_{31}(t)}{F_{21}(t)}-\frac{F'_{31}(t)}{F'_{21}(t)}\right]\equiv k_3F_{3121}(t). \tag{7}$$

Step 3. The relationship given by Eq. (7) shows the linear dependence of the function F_{0121} on the function F_{3121}. Again, on the ground of Assertion 3 of Lemma 1, we conclude that the DDF $F_{01213121}(t)$ of the function F_{0121} with respect to F_{3121} is *zero* for all feasible t:

$$F_{01213121}(t) \equiv @d^{(1)}_{F_{3121}} F_{0121}(t) = \frac{F_{0121}(t)}{F_{3121}(t)} - \frac{F'_{0121}(t)}{F'_{3121}(t)} = k_3 - k_3 = 0.$$

Now, one can use relationships of Eq. (6) and Eq. (7) in the converse order and compute the desired values of the weights k_i. Indeed, first from Eq. (7), one readily obtains

$$k_3 = \frac{F_{0121}}{F_{3121}}; \tag{8}$$

the latter, in its turn, combined with Eq. (6) implies:

$$k_2 = \frac{F_{01} - k_3 F_{31}}{F_{21}}. \tag{9}$$

Finally, by substituting the just found weights k_2 and k_3 in Eq. (5), one deduces the value of k_1:

$$k_1 = \frac{y(t) - k_2 f_2(t+\tau_2) - k_3 f_3(t+\tau_3)}{f_1(t+\tau_1)}. \quad (10$$

The algorithm stops after having decoded the received message $y(t)$ by having identified the (unknown) weights associated with the participating key functions. All the weights related to the idle (non-participating) key functions are ***zero***.

Remark 2. As one can smoothly infer, the explicit list of basic key functions and their possible delay (shift) values τ_i is ***indispensable*** for the implementation of this simplified version of the decoding algorithm. The more sophisticated procedures that may be needed to decipher the received message in the lack of such important information, that is, under more profound uncertainty, are described in [6].

5 Numerical Examples and Experiments

In order to illustrate the cryptosystem's operation, let us consider the binary coding in the form of an arbitrary sequence of symbols "0", "1", space "_", and a transition to the new line (paragraph return) "\". For this model, only three real key functions are employed. The symbols being transmitted are encoded by the (weighted) sum of at least two of these functions multiplied by random factors (weights). The time delays (shifts) of the standard functions with respect to the current time *t* are assumed to be *zero*. The communication system (TV or radio channel) can transmit only binary code symbols. Therefore, if there appears any other symbol apart from those listed above, it is perceived as a paragraph return.

To develop the numerical methods calculating the approximate derivatives, it is necessary to control the signal *y*(*t*) within the interval containing at least 10 (discrete) points of the time variable *t*. In fact, the number of points in this interval may vary (the greater

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_23

this number of points, the higher the cryptosystem's stability (resistance)), but in our case, it is selected constant and equals 75 (*cf.*, again, [4]).

In order to simulate the operations of the cryptosystem, the following three functions are employed as key functions:

$$f_1(t) = 100\,sin\left(\left(\alpha_1 - \beta_1\right)t\right)cos\left(15\beta_1 t\right);$$
$$f_2(t) = 100\,exp\left(-0.1\alpha_2 t\right)sin\left(10\beta_2 t\right)cos\left(\left(\alpha_2 + \beta_2\right)t\right);$$
$$f_3(t) = 100\,exp\left(-0.1\alpha_3 t\right)sin\left(400\beta_3 t\right),$$

where $\alpha_1 = 1$; $\alpha_2 = 0.12$; $\alpha_3 = 0.5$; $\beta_1 = 0.1$; $\beta_2 = 1.2$; $\beta_3 = 0.7$.

The weights k_1, k_2, and k_3, with which the key functions encode the signal $y(t)$ by Eq. (5) before its transmission, have been selected randomly by making use of a pseudo-random generator with the uniform distribution from *zero* to 10 (for each symbol). However, only when encoding a symbol '1', $y(t)$ includes the entire (weighted) sum of all three key functions and therefore, their coefficients k_1, k_2, and k_3 are not equal to zero. When encrypting '0', we put $k_1 = 0$, and while encoding a space, we set $k_3 = 0$. Finally, if another symbol or the paragraph return is encoded, then $k_2 = 0$.

At any given time moment, the receiver tries to identify the involved key functions and calculate the unknown weights (coefficients) $k_i,\ i = 1, 2, 3$, by making use of the formulas from Eq. (8) – Eq. (10). Thereafter, the received message is decoded.

When an arbitrary text is encrypted with the application of derivative disproportion functions, it is always recommended to introduce at least two random letters before the transmission of the binary code.

Figures 1, 2, and 3 show the diagrams of the signal $y(t)$ transmitted via the communication channel. Various examples of the cryptosystem operation when the same symbols are transmitted, as well as when the binary symbols are alternated, were treated.

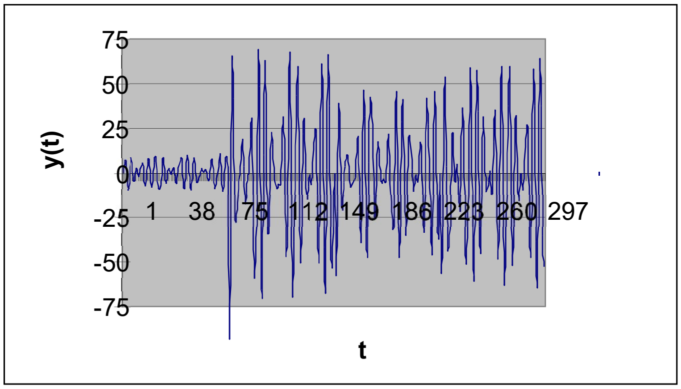

Fig. 1. The signal corresponding to the serial transmission of four symbols '0'.

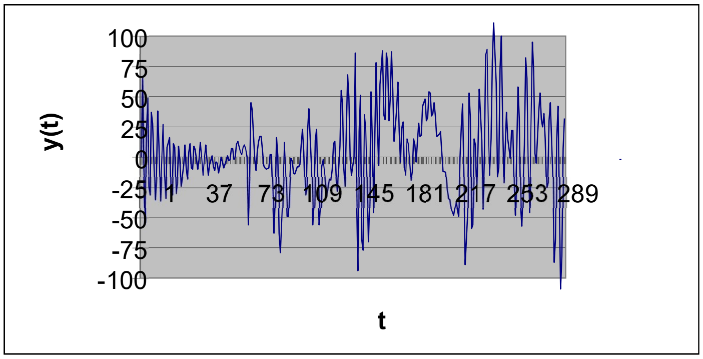

Fig. 2. The signal corresponding to the serial transmission of four symbols ‘1’.

Besides, the case when ASCII- codes of symbols A, B, C, D, O are transmitted, was tested. The corresponding codes were as follows:

01000001 01000010 01000011
01000100 01001111.

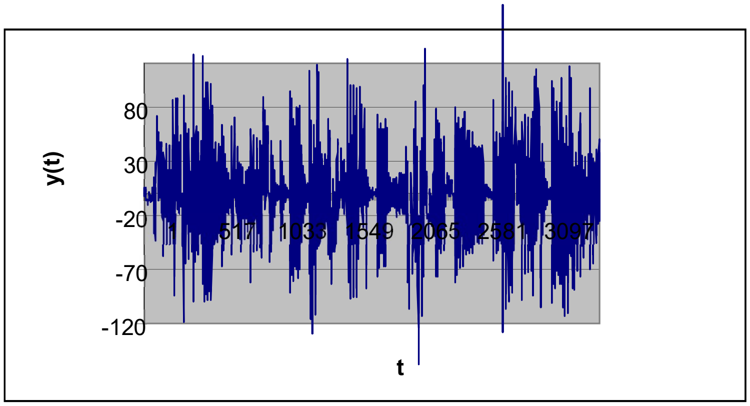

Fig. 3. The signal corresponding to the serial transmission of ASCII- codes of symbols A, B, C, D, O.

The cryptosystem’s operation can be illustrated by the last (third) example. The following message was encrypted:

01000001 01000010 01000011

01000100

01001111

The following message was obtained after decoding:

DOI: 10.1007/978-3-030-22750-0_23

01000001 01000010 01000011

01000100

01001111

According to the operational algorithm, each of the letters led to the appropriate paragraph return.

In all cases, the received message was deciphered exactly as what was transmitted. At the same time, as it can be seen from the figures, it is quite difficult to reveal a message born by the transmitted signal through the communication channel unless the decoding algorithm is applied.

6 Robustness of the Cryptosystem

The cryptosystem's robustness (stability) depends on the choice of the key functions as well as on their total number. The more components are involved in the signal, the more difficult becomes the task of deciphering (in case it's been intercepted as a result of a hacker attack). Obviously, it is necessary ***not only*** to identify the type and the number of key functions ***but also*** to fit the weights involved.

How difficult it is to fit their values can be judged from the fact that in the given example, it suffices to apply $sin\left(9.9999\beta_2 t\right)$ instead of the present $sin\left(10\beta_2 t\right)$ in $f_2(t)$, or to select 400.0001 instead of the current 400 in $f_3(t)$, so that the code word consisting of four consecutive 0's is "decoded" as four 1's. This simple example confirms that any attempts on part of a hacker to "guess" the coded word by an exhaustive search for the coefficients (weight) even after having detected the key functions used, is almost always doomed to fail.

Another instance: the replacement of $\alpha_1 = 1$ with $\alpha_1 = 0.99$ in $f_1(t)$ has resulted in the distorted reception of the sole line 11000000011010000001101000000000000100000000000 without breaks in contrast to the three-line original message boasting with spaces as well.

The cited examples show that it is quite difficult just to fit the weights by a simple guess, to say nothing about the necessity to determine the number of functions and to fit their types.

It should be also noted that the same symbol is encoded differently depending on its position (location). Besides, one should pay attention to the fact that in this case, the frequency analysis cannot be applied for unauthorized access and decoding.

All the above-mentioned facts show that the cryptosystems based on the (weighted) sum of real key functions are sufficiently resistant to hacking (cryptographically secure).

7 Requirements for the Key Functions

1. Each (one real variable) key function has to be real-valued and sufficiently smooth (*n* times continuously differentiable).

2. The key function and its derivatives up to order *n* must *not* be constant.

3. The key functions should *not* asymptotically approach a constant value within its domain (e.g., like the function $x^{-\alpha}$, $\alpha > 0$, for the large values of *x*).

4. The collection of key functions must be selected so that to exclude the possibility that the value of one function at some point be negligible (too small by its absolute values) as compared to the values of other functions at the same point; that is, every function has to make a quite significant "contribution" to the (weighted) sum of all key functions.

5. The key functions cannot be identical.

8 Concluding Remarks

We develop a cryptosystem where (one-variable) real functions are used as the keys. An example is presented to illustrate the operation of such a system where symbols are encrypted by the (weighted) sum of the key functions with random coefficients. The decoding is conducted with the aid of the first order derivative disproportion functions (DDF) calculated for the received signal and the key functions.

For a practical application of such cryptosystems, one should bear in mind that in the process of computing the weights during deciphering, a division by small numbers, or a ratio of two numbers both close to zero may happen. This can lead to certain information distortion. Therefore, the encrypted message must be decoded before it is transmitted via a communication channel. If necessary, the message should be encrypted once again with other coefficients (weights) generated randomly for every key function.

Acknowledgments

The authors would like to express their profound gratitude to the two anonymous referees whose valuable critical and constructive comments and proposals have helped a lot in improving the paper.

This work was financially supported by the SEP-CONACYT grants CB-2013-01-221676 and FC-2016-01-1938, as well as by Tecnológico de Monterrey (ITESM), Campus Monterrey, Strategic Group for Optimization and Data Science.

References

1. U.S. Department of Commerce/National Institute of Standards and Technology Data Encryption Standard (DES) Federal Information, Processing Standards Publication, 46-3, October 25 (1999). http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf
2. Federal Information Processing Standards Publication 197, November 26 (2001), Specification for the ADVANCED ENCRYPTION STANDARD (AES). http://crsc.nist.gov/publications/fips/fips197/fips197.pdf
3. Rivest, R., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public – key cryptosystems. Communications of the ACM 21(2), 120–126 (1978). DOI: 10.1145/359340.359342
4. Khan, I.R., Ohba, R., and Hozumi, N.: Mathematical proof of closed-form expressions for finite difference approximations based on Taylor series. Journal of Computational and Applied Mathematics 150(3), 303–309 (2003).
5. Avramenko, V.V., Zabolotny, M.I.: A Way of Data Coding, Patent UA H04L 9/00 №42957, Ukraine (2009).
6. Avramenko, V.V., Karpenko, A.P.: Recognition of fragments of given standards in an analyzed signal with the aid of disproportionality functions. Transactions of Sumy State University (SumDU), 34(1), 96–101 (2002).
7. Kalashnikov, V.V., Avramenko, V.V., Kalashnykova, N.I., Kalashnikov, V.V.-Jr.: A cryptosystem based upon sums of key functions. International Journal of Combinatorial Optimization Problems and Informatics*,* 8(1), 31–38 (2017).
8. 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**.** Procedia Computer Science*,* 108(C), 2100–2109 (2017).
9. Kalashnikov, V.V., Avramenko, V.V., Kalashnykova, N.I.: Derivative disproportion functions for pattern recognition. In: Junzo Watada, Shing Chiang Tan, Pandian Vasant, Eswaran Padmanabhan, Lakshmi C. Jain (Eds.), *Unconventional Modelling, Simulation, and Optimization of Geoscience and Petroleum Engineering, Chapter 7*. Springer-Verlag, Berlin-Heidelberg, 95 – 104 (2018).