UDC 517.9/519.6

S. Grybniak,
D. Dmitrishin, DSc, Prof.
Odessa Polytechnic National University, 1 Shevchenko Ave., Odessa, Ukraine, 65044; e-mail: dmitrishin@op.edu.ua

BASIC PRINCIPLES FOR CONSTRUCTING MIXING FUNCTIONS BASED ON THE SIMPLEST LINEAR AND NONLINEAR MAPPINGS

С. Грибняк, Д. Дмитришин. **Основні принципи конструювання функцій змішування на основі найпростіших лінійних і нелінійних відображень.** Стаття присвячена дослідженню можливості використання у комп’ютерній криптографії найпростіших принципів теорії нелінійних дискретних динамічних систем, які характеризуються своєю хаотичною поведінкою. Основна проблема використання хаотичних систем в комп’ютерних розрахунках полягає в тому, що число можливих станів в комп’ютері кінцеве. Тому комп’ютерні моделі хаосу є лише наближенням істинної хаотичної поведінки, а кожна траєкторія апроксимованої системи є періодичною. З математичної точки зору шифрування в інформаційних системах є перетворення простору скінченних повідомлень, що відповідає фазовому простору в теорії динамічних систем. Функція змішування визначає таке шифрування. Основними вимогами до функції змішування є відсутність колізій, тобто бієктивність відображення, хороші дифузійні властивості та крім того, зворотне перетворення не повинно бути складнішим за пряме. У статті показано, що є можливість можна використовувати дифузійні властивості нелінійних динамічних систем у просторах зі скінченною кількістю станів за допомогою найпростішого нелінійного відображення Тент. Для покращення дифузійних властивостей було використано суперпозицію нелінійного відображення Тент та лінійного відображення перестановки (у більш загальному випадку перетворення Хілла). Основними перевагами побудованих функцій є їхня простота реалізації, швидкість обчислень у задачах змішування та сильна криптографічна стійкість. Для цих функцій було проведено кореляційний аналіз, аналіз чутливості та аналіз величин періодів циклу, які поділяють простір на підмножини, що не перетинаються. У результаті підтверджуються очікувані хороші дифузійні властивості цих функцій змішування. Продемонстровано можливість застосування цих функцій для шифрування зображень.

Ключові слова: нелінійні дискретні системи, функція змішування, шифрування

S. Grybniak, D. Dmitrishin. **Basic principles for constructing mixing functions based on the simplest linear and nonlinear mappings.** This article is dedicated to exploring the possibility of using the simplest principles of nonlinear discrete dynamic systems theory in computer cryptography, which are characterized by their chaotic behavior. The main problem of using chaotic systems in computer calculations is that the number of possible states in a computer is finite. Therefore, computer models of chaos are only an approximation of the true chaotic behavior, and each trajectory of the approximated system is periodic. From a mathematical point of view, encryption in information systems involves transforming the space of finite messages, which is similar to the phase space in the theory of dynamical systems. The mixing function specifies such encryption. The main requirements for the mixing function are the absence of collisions, i.e., bijectivity of the mapping, good diffusion properties, and, in addition, the inverse transformation should not be more complicated than the direct one. The article demonstrates that it is possible to utilize the diffusion properties of nonlinear dynamical systems in spaces with a finite number of states by using the simplest nonlinear mapping, Tent. To enhance the diffusion properties, a superposition of the nonlinear Tent map and the linear permutation map (in the more general case of the Hill map) was used. The main advantages of the constructed functions are their simplicity of implementation, speed of calculations in mixing problems, and strong cryptographic persistence. Correlation analysis, sensitivity analysis, and analysis of the lengths of cycle periods that divide the space into non-overlapping subsets have been conducted for these functions. As a result, the expected good diffusion properties of these mixing functions are confirmed. The possibility of applying these functions to image encryption is demonstrated.

Keywords: nonlinear discrete systems, mixing function, encryption

Introduction

Cryptographic functions are used in information security tasks. There are a large number of diverse functions used in cryptography, based on various algorithms for their construction. The need to implement information protection in systems with limited computational resources leads to the need for the development of new cryptographic algorithms. The main properties of functions used in these algorithms are the properties of scattering and mixing [1].

From a mathematical point of view, encryption in information systems is often presented as the implementation of various transformations $G: X \to X$ on sets of finite messages $X = \{x_1, \dots, x_k\}$. Assuming that $x_1, \dots, x_k$ are different m-digit positive integers than encode possible messages, where $k < 10^m$ and $x_j < 10^m$, $j = 1, \dots, k$. In this case, transformations G must possess the property of absence of collisions, i.e. $G(x_j) = G(x_i) \Leftrightarrow x_j = x_i$. In addition, the inverse transformation G^{-1} should not be more difficult than the direct one. If $k = 10^m - 1$, then the function $G(x)$ is called a complete mixing function or a permutation function. In this case $X = \{1, 2, \dots, 10^m - 1\}$.

DOI: 10.15276/opu.2.66.2022.12

The practical implementation of mixing functions for large values of requires overcoming significant difficulties associated with describing and setting mixing parameters, developing mathematical and software support, and developing new mathematical models and methods.

Analysis of publications on the research topic

One of the most common mathematical methods used to construct mixing functions are nonlinear dynamics methods [2 – 5]. This can be explained by the fact that both in cryptography and in nonlinear systems, a nonlinear transformation of the space of the source information (phase space) is performed. On the one hand, this transformation is fully determined, on the other hand, it has aperiodic ergodicity and extreme sensitivity to initial conditions [6, 7], which allows for the generation of a pseudorandom chaotic sequence possessing characteristics that coincide with the basic requirements of cryptography. The structure of discrete-time dynamical systems is relatively simple, computer implementation is convenient, and computation speed is high. However, the values of the elements of the resulting chaotic sequence in phase space are distributed non-uniformly, i.e., insufficiently mixed, which leads to security risks [8].

On the other hand, permutation functions based on linear operators (perhaps with an additional modulus comparison operation) mix the elements of the phase space well [9]. Such permutation functions were first constructed and used by L. Hill [10, 11]. The main advantages of the Hill cipher are resistance to complete enumeration even with relatively small key sizes, simplicity, and speed of application. However, there is a significant disadvantage inherent in all cryptographic functions with pronounced linear operation properties: vulnerability to chosen plaintext attacks. In this regard, to increase cryptostrength, nonlinear operations must be added to linear ciphers. Combining linear operations, as in the Hill cipher, with nonlinear ones led to the creation of substitution-permutation networks (for example, the Feistel network [12]). Currently, methods for constructing mixing functions that use superpositions of linear and nonlinear operations are still actively used [13].

Purpose and Objectives of the Research

The main objective of this article is to demonstrate the possibility of constructing mixing functions based on simple linear and nonlinear mappings. Specifically, a permutation operator is proposed as a linear mapping, and the classical tent map is suggested as a nonlinear mapping. Other objectives include conducting correlation analysis of the constructed mappings, sensitivity analysis, cycle length analysis, exploring the possibility of generalizing algorithms for constructing mixing functions and applying them to image encryption.

Auxiliary results

Next, we will use the classical Tent mapping [14]:

$$f(x) = 2(1/2 - |x - 1/2|) = \begin{cases} 2x, x \le 1/2, \\ 2\,(1-x), x > 1/2 \end{cases},$$

$[0,1] \xrightarrow{f} [0,1]$, and mapping q -Tent (q - an odd number):

$$f_q(x) = 2(q/2 - |x - q/2|) = \begin{cases} 2x, x \le q/2, \\ 2(q-x)\ , x > q/2, \end{cases}$$

$$[0,q] \xrightarrow{f_q} [0,q]\ .$$

These mappings have identical properties (they differ from each other by extension/compression) and are interconnected by the relations:

$$f_1(x) = f(x)\,,\ \ f_q(x) = q\,f(x/q)\,.$$

Let us note some important properties of the Tent mapping.

Property 1. If $p_1 / q = f(p/q)$ (q is an odd number), then the numerator is uniquely restored p, namely, $p = p_1 / 2$, if $p_1 / 2$ is an even number, or $p = q - (p_1 / 2)$, if $p_1 / 2$ is an odd number.

Thus, if p is an even number, then the preimage is uniquely restored from the image.

Property 2. The set of proper fractions with even numerators and odd denominators is invariant under the tTent mapping; on this set, the Tent mapping is reversible.

Property 3. If p_1 and p_2 are even numbers, then $f_q(p_1) = f_q(p_2)$ if and only if $p_1 = p_2$.

As will be shown later, the use of mapping $f_q(x)$ is more convenient than mapping $f(x)$, firstly, from a computational point of view, and secondly, the fraction p/q can be reduced, and such a reduction is not desirable for the future algorithm.

Also note that the future algorithm requires that computational procedures using the awning mapping be performed without rounding, and hence exclusively with ordinary fractions. The point is that a chaotic system cannot be implemented on a computer because of the finiteness of the number of states. Each subsequent state of the system must not coincide with any previous state of the trajectory. Otherwise (for example, as a result of rounding) the trajectory turns into a cyclic orbit or degenerates into a stationary point, as in the tent mapping. Consider the sequence generated by the Tent mapping $\{f^{(n)}(x_0)\}$, Where $f^{(1)}(x) = f(x)$, $f^{(k)}(x) = f(f^{(k-1)}(x))$. We use the EXEL package. Pick a starting point $x_0 = 2/3$. Then theoretically it should be $f^{(k)}(x_0) = x_0$, $k = 1, 2, \dots$ However, the calculations give different results: $f^{(53)}(x_0) = 1$, $f^{(54)}(x_0) = 0$. The first impression is that the problem arose because the number $2/3$ is not represented as a terminating decimal. Then pick $x_0 = 0.4$. Theoretically we should obtain $f^{(1)}(x_0) = 0.8$, $f^{(2)}(x_0) = 0.4$ etc. However, as a result of calculations, we obtain $f^{(53)}(x_0) = 1$, $f^{(54)}(x_0) = 0$. Exactly the same results will be obtained in calculations in the MAPLE package (and in other similar packages) if we set $x_0 = \text{HFloat(evalf}(2/3))$ or $x_0 = \text{HFloat}(0.4)$. For any other initial values x_0, we will again receive $f^{(54)}(x_0) = 0$.

The reason for such incorrect calculations can be explained by the fact that the number from the segment $[0, 1]$ is represented in the computer in binary notation as a finite sum of numbers of the form $\alpha_j / 2^j$ ($\alpha_j \in \{0, 1\}$), i.e. are a binary rational number, it is equal to $A/2^m$ (A is an integer and $0 \le A \le 2^m$). The number m is called order. But then $f(A/2^m) = A_1/2^{m-1}$ (A_1 is an integer and $0 \le A_1 \le 2^{m-1}$), $f^{(2)}(A/2^m) = A_2/2^{m-2}$, … , $f^{(m+1)}(A/2^m) = 0$. On the computers where the iterated sequence was calculated, the maximum order turned out to be 53.

If the q-tent mapping is used on a subset of even natural numbers, then the permutation operator is defined on the set of vectors with components consisting of single-digit natural numbers. The linear permutation operator is defined by a permutation matrix consisting of zeros and ones, with exactly one unit in each row and column. This operator is invertible, and the inverse matrix is equal to the transpose of the original matrix. Thus, both the tent map operator and the permutation operator bijectively map the aforementioned sets onto themselves.

Main result

Let us introduce the following notation. Let there be sets of finite messages X. Let the message be some number $x \in X$, $x < 10^m$. Let's take this message as:

$$x = \xi_1 10^{m-1} + \xi_2 10^{m-2} + \dots + \xi_{m-1} 10 + \xi_m \ , \xi_j \in \{0, 1, \dots, 9\},\ j = 1, \dots, m.$$

Equivalent notation is $x = [\xi_1 \xi_2 \dots \xi_{m-1} \xi_m]$.

Number $x = [\xi_1 \xi_2 \dots \xi_{m-1} \xi_m]$ is corresponded to the vector $\overline{\mathbf{x}} = \{\xi_1, \xi_2, \dots, \xi_{m-1}, \xi_m\}$. This correspondence is one-to-one. Let's denote it as $\overline{\mathbf{x}} = \phi(x)$, and the inverse $x = \phi^{-1}(\overline{\mathbf{x}})$. We introduce the permutation operator $\pi_{\mathbf{M}} : X \to X$,

$$\pi_{\mathbf{M}}(x) = \phi^{-1}(\mathbf{M}\,\overline{\mathbf{x}}) = \phi^{-1}(\mathbf{M}\,\phi(x)),$$

where $\mathbf{M}$ – $m \times m$ dimension permutation matrix.

Obviously, the inverse operator is also a permutation operator: $\pi_{\mathbf{M}}(x) = \phi^{-1}(\mathbf{M}^*\phi(x))$, where $\mathbf{M}^*$ – transposed to $\mathbf{M}$ matrix.

Based on nonlinear operators $f_q(x)$, $f_q^{(T)}(x)$ we introduce nonlinear operators $(1/2) f_q(2x)$, $(1/2) f_q^{(T)}(2x)$, Where $q = 2 \cdot 10^m - 1$. These operators map one-to-one $X \to X$. Back to them $(1/2)(f_q)^{-1}(2x)$, $(1/2)(f_q^{(T)})^{-1}(2x)$.

The mixing algorithm is defined by the formula $y = G(x, T, M)$, where the mixing function:

$$G(x,T,\mathbf{M})=\phi^{-1}(\mathbf{M}\,\phi((1/2)\,f_q^{(T)}(2x)))=\pi_{\mathbf{M}}((1/2)\,f_q^{(T)}(2x))\,.$$

Inverse function:

$$G^{-1}(x,T,\mathbf{M})=(1/2)(f_q^{(T)})^{-1}(2\phi^{-1}(\mathbf{M}\phi(x)))=(1/2)(f_q^{(T)})^{-1}(2\pi_{\mathbf{M}}(x))\,.$$

Also the mixing function.

The mixing function can also be considered as an encryption function, in which the parameter T and matrix $\mathbf{M}$ are the keys.

Naturally, the superposition of mixing functions is a mixing function, i.e. $G(G(x,T_1,\mathbf{M}_1),T_2,\mathbf{M}_2)$, depending on two parameters T_1 and T_2, and two matrices $\mathbf{M}_1$ and $\mathbf{M}_2$, i.e. cryptographic strength can increase with superposition.

Correlation analysis

To perform a correlation analysis, we will conduct an experimental test. Let's put $m=4$,

$$T_1=372\,,\ T_2=251\,,\ \mathbf{M}_1=\begin{pmatrix}0&0&1&0\\0&0&0&1\\0&1&0&0\\1&0&0&0\end{pmatrix},\ \mathbf{M}_2=\begin{pmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{pmatrix}.$$

Consider three mixing functions:

$$G_1(x)=(1/2)\,f_q^{(T_1)}(2x)\,;\ G_2(x)=G(x,T_1,\mathbf{M}_1)\,;\ G_3(x)=G(G(x,T_1,\mathbf{M}_1),T_2,\mathbf{M}_2)\,,$$

where $q=2\cdot 10^m-1=19999$.

The first function uses exclusively non-linear transformation q-tent; in the second, a linear mixing transformation is added; in the third one, the superposition of the previous function with different parameter values is used. For every function $G_k(x)$ construct mappings $x=\{i/10^m\}_{i=1}^{10^m}\to$ $y=\{(1/10^m)G_k(i)\}_{i=1}^{10^m}$, $k=1,2,3$, and analyze the statistical indicator of linear dependence, namely, the correlation coefficient:

$$\sigma_{xy}^{(k)}=(1/10^m)\left(\sum_{i=1}^{10^m}((i/10^m)-\overline{X})(y_i-\overline{Y})\right)\Big/(\sigma_x\sigma_y)\,,$$

where $\overline{X}=\overline{Y}=(1/10^m)\sum_{i=1}^{10^m}(i/10^m)=(1/10^m)\sum_{i=1}^{10^m}(1/10^m)G_k(i)=0.5$.

Calculate $\sigma_y^{(k)}=0.28864626$, $k=1,2,3$, then the correlation coefficients

$$\sigma_{xy}^{(1)}=-0.00014894...,\ \sigma_{xy}^{(2)}=-0.00014533...,\ \sigma_{xy}^{(3)}=0.00039123....$$

It can be seen that there is no linear relationship between the sequences $x=\{i/10^m\}_{i=1}^{10^m}$ and $y=\{(1/10^m)G_k(i)\}_{i=1}^{10^m}$, however, a significant dependence (nonlinear) in the first two cases can be observed on the charts (Fig. 1).

Sensitivity analysis

Let there be two messages $x_0,y_0\in X$, whose numeric representations differ by only one character. Consider successive transformations of these messages under the action of the mixing function $G_k(x)$: $x_{n+1}=G_k(x_n)$, $y_{n+1}=G_k(y_n)$, $k=1,2,3$, $n=0,1,2\ldots$

Task: Calculate the mean and standard deviation of the sequence $\{x_i-y_i\}_{i=1}^{\infty}$.

Consider the mean of the sequence $\{x_i-y_i\}$, i.e. function $S_n=(1/(n+1))\sum_{i=0}^{n}(x_i-y_i)$. We write the recurrence relation $S_n=(n/(n+1))S_{n-1}+(1/(n+1))(x_n-y_n)$, $n=0,1,2\ldots$, $S_0=x_0-y_0$. Let's take, for example, $x_0=[0001]=1$, $y_0=[0002]=2$. For every function $G_k(x)$ calculate the elements of the sequence $\{S_i\}_{i=1}^{\infty}$ (Fig. 2, 3 and 4).

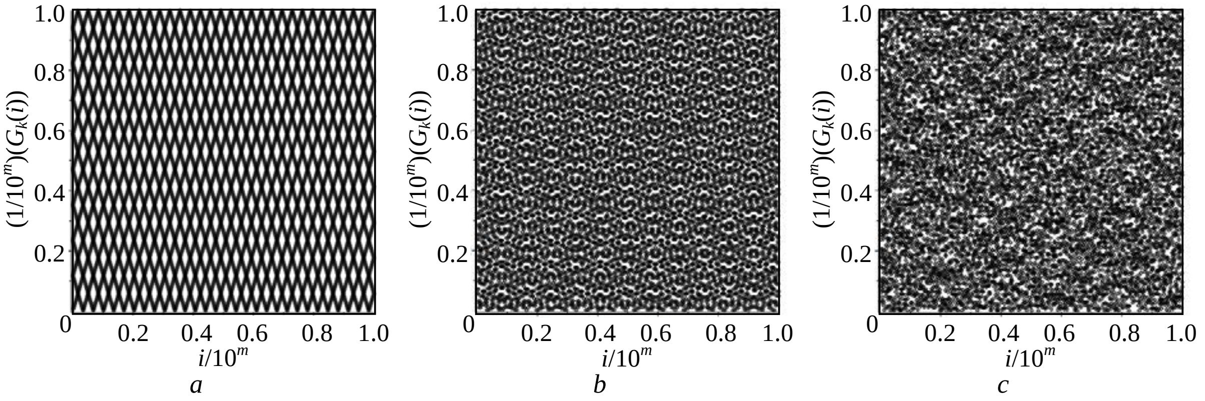

Fig. 1. Mapping charts $x=\{i/10^m\}_{i=1}^{10^m} \to y=\{(1/10^m)G_k(i)\}_{i=1}^{10^m}$: $a-k=1$; $b-$ $k=2$; $c-$ $k=3$

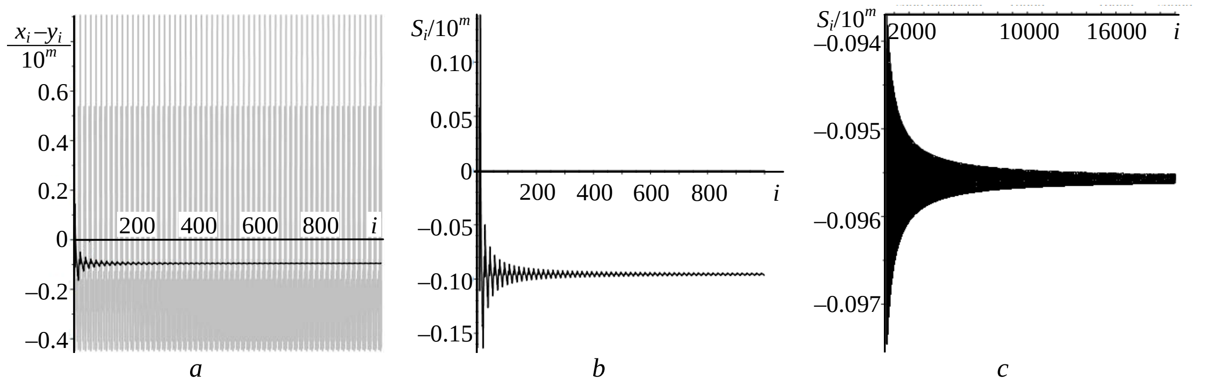

Fig. 2. Mapping charts: $a-$ $\{(x_i-y_i)/10^m\}_{i=1}^{1000}$ (grey colour); $b-$ $\{S_i/10^m\}_{i=1}^{1000}$;
$c-$ $\{S_i/10^m\}_{i=500}^{20000}$ for mixing function $G_1(x)$

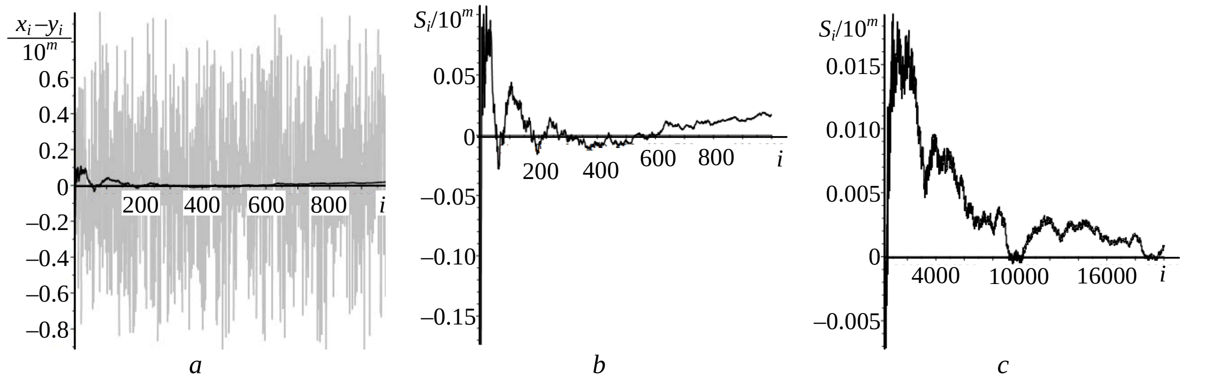

Fig. 3. Mapping charts: $a-$ $\{(x_i-y_i)/10^m\}_{i=1}^{1000}$ (grey color); $b-$ $\{S_i/10^m\}_{i=1}^{1000}$;
$c-$ $\{S_i/10^m\}_{i=500}^{20000}$ for mixing function $G_2(x)$

Consider the variance and standard deviation of the sequence $\{x_i-y_i\}$, i.e. function $D_n=\frac{1}{n}\sum_{i=0}^{n}(x_i-y_i-S_i)^2$ and $\sigma_n=\sqrt{D_n}$. We write the recurrence relation $D_n=\frac{n-1}{n}D_{n-1}+\frac{1}{n}(x_n-y_n-S_n)^2$, $D_0=0$. For the same initial values $x_0=[0001]$, $y_0=[0002]$ and for each function $G_k(x)$ we calculate the elements of the sequence $\{\sigma_i\}_{i=1}^{\infty}$ and (Fig. 5 and 6).

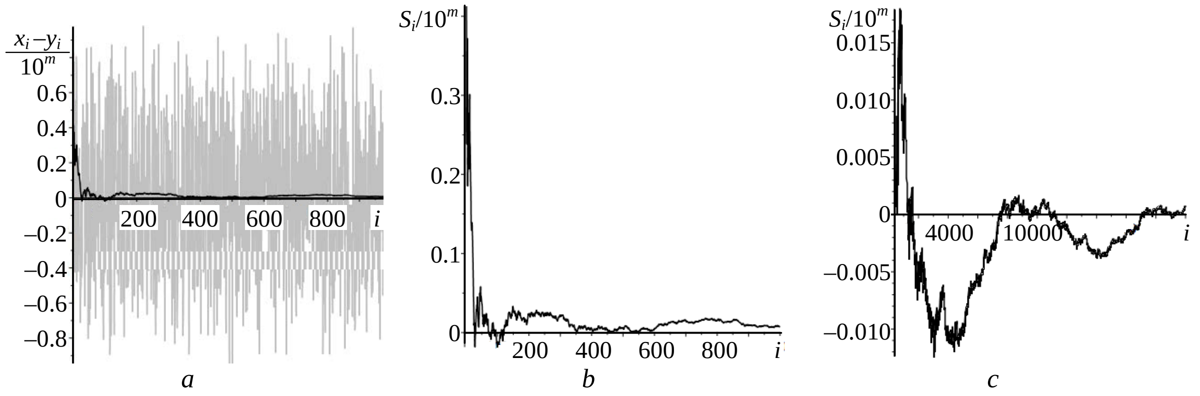

Fig. 4. Mapping charts: a – $\{(x_i - y_i)/10^m\}_{i=1}^{1000}$ (grey color); b – $\{S_i/10^m\}_{i=1}^{1000}$; c – $\{S_i/10^m\}_{i=500}^{20000}$ for mixing function $G_3(x)$

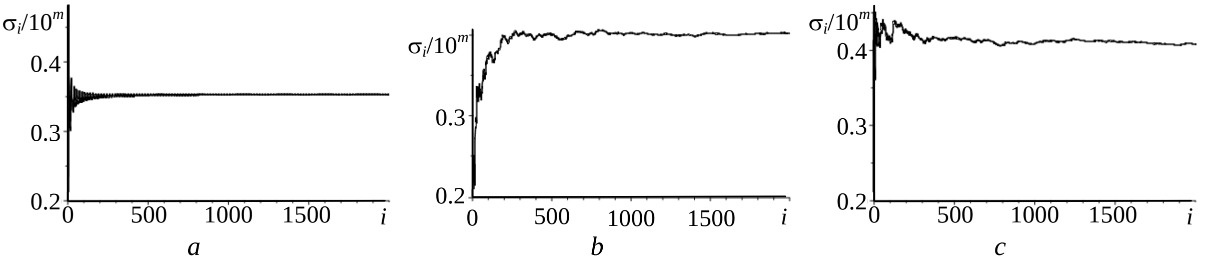

Fig. 5. Mapping charts $\{\sigma_i/10^m\}_{i=0}^{2000}$ for mixing function $G_k(x)$: a – $k=1$; b – $k=2$; c – $k=3$

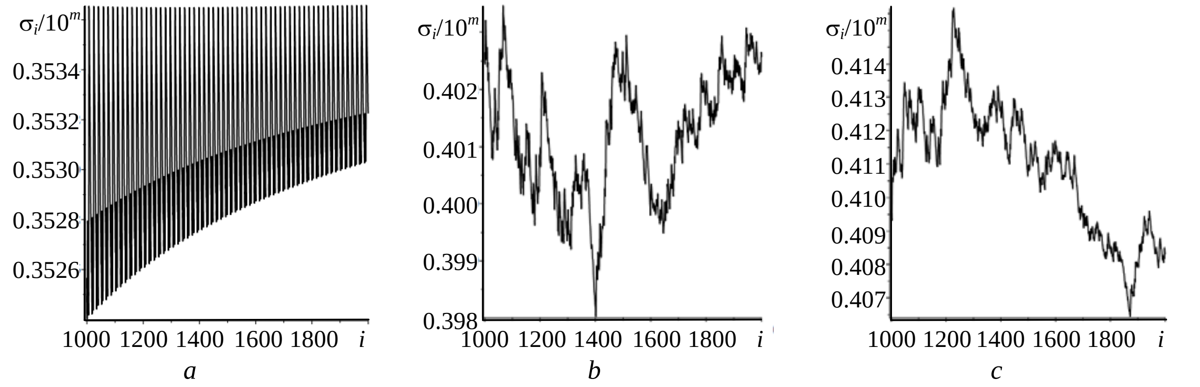

Fig. 6. Mapping charts $\{\sigma_i/10^m\}_{i=1000}^{2000}$ for mixing function $G_k(x)$: a – $k=1$; b – $k=2$; c – $k=3$

Similar results are obtained if other close values of the digital representations of the original messages are chosen. Based on the results of the experimental analysis, it can be concluded that the function $G_1(x)$ mixes poorly, and there is a clear preservation of dependence between the numerical values of messages in the process of mixing when the initial values were close The mean differences and standard deviations tend to a constant as the number of mixing increases. Conversely, for the mixing functions $G_2(x)$, $G_3(x)$ the mean differences and standard deviations behave without a pronounced trend (Fig. 6).

Analysis of period lengths

The mixing function $G(x)$ maps a finite set X into yourself. This means that for any $x \in X$ iterative sequence $\{G^{(i)}(x)\}_{i=0}^{\infty}$ (considered, $G^{(0)}(x) = x$) will be periodic, with a period of $10^m - 1$ at most.

When constructing mixing functions, it is important to know how the set is partitioned into subsets defined by periodic orbits. It is important to know the number of these sets and their sizes, i.e. minimum, average and maximum lengths of periodic orbits, and the number of orbits. The presence of a large number of orbits with short lengths is dangerous, because it leads to the formation of patterns.

It is most preferable to have a single long orbit passing through almost all space. An ensemble of single-type orbits of medium length may also be suitable.

Let us again consider the function $G_3(x) = G(G(x, T_1, \mathbf{M}_1), T_2, \mathbf{M}_2)$ with previously introduced parameter values $T_1, T_2, \mathbf{M}_1, \mathbf{M}_2$. And let us set the task: to find all subsets of the set X, defined by periodic orbits.

Numerical analysis shows that the mapping $G_3(x)$ has three fixed points, four periodic orbits of length 2, one orbit of lengths 3, 4, 6, 16, 17, 19, 319, 9604 each. Table 1 lists the points generating the corresponding periodic orbits.

Table 1

Periods of orbits and lists the points generating the corresponding periodic orbits

Period orbits	**1**	**2**	**3**	**4**	**6**	**16**	**17**	**19**	**319**	**9604**
Generative dot	6234 9666 9964	2122 5427 6111 9653	5913	1737	0109	0037	1192	0089	0016	0001

Note that the two longest orbits contain 9923 points, which is more than 99 % of the total number of points in the set X. Such a distribution of mapping cycles testifies to the good diffusion properties of the mixing function, built on the simplest basic principles. On Fig. 7 shows graphs of functions $\{(1/10^m) G_3^{(i)}(x)\}_{i=0}^{9999}$ for different starting points.

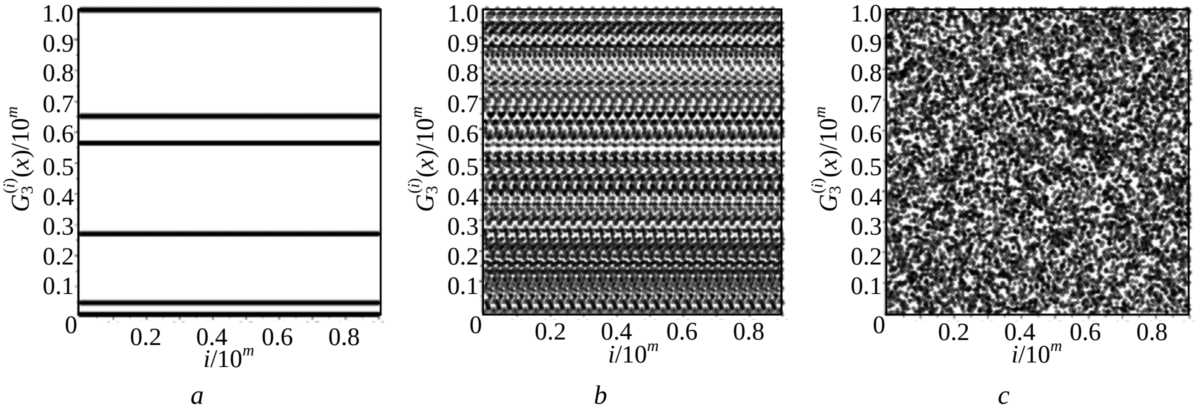

Fig. 7. Mapping charts $\{(1/10^m) G_3^{(i)}(x)\}_{i=0}^{10^m}$: a – $x = [0109]$ ($x = 109/10^m$, $T = 6$); b – $x = [0016]$ ($x = 16/10^m$, $T = 319$); c – $x = [0001]$ ($x = 1/10^m$, $T = 9604$)

Modification options for the mixing mapping

When constructing a mixing function:

$$G(x, T, \mathbf{M}) = \phi^{-1}(\mathbf{M}\,\phi((1/2) f_q^{(T)}(2x))) = \pi_{\mathbf{M}}((1/2) f_q^{(T)}(2x)),$$

permutation matrix $\mathbf{M}$ with dimensions $m \times m$ was used. There are $m!$ such matrices. Note that when $m \ge 25$ $m\,! > 10^m$, and when $m \ge 7$ $(m\,!)^2 > 10^m$. Thus, when $m \ge 25$ hacking the permutation function would require brute force, i.e. exhaustive search of all possible options. And to break the superposition of two mixing functions, exhaustive search is required when $m \ge 7$. However, the mixing properties of a linear operator can be further enhanced using Hill's algorithm, namely, instead of a permutation matrix, take any $m \times m$ matrix, ensuring only that the value of its determinant is not divisible by 2 or 5. Let A is such a matrix. Then the operator:

$$\pi_{\mathbf{A}}(x) = \phi^{-1}(\mathbf{A}\bar{x}(\mathrm{mod}10)) = \phi^{-1}(\mathbf{A}\,\phi(x)(\mathrm{mod}10)),$$

maps X into X unambiguously [10]. The number of suitable matrices is more than $10^{m(m-1)}$. On the one hand, only single-digit numbers can be used as elements of the matrix $\mathbf{A}$ (since calculations are done modulo 10), on the other hand, matrices with determinants equal to zero, even, or divisible by five should be discarded. It should be noted that $\mathbf{M}\,\overline{\mathbf{x}}(\text{mod}10)=\mathbf{M}\,\overline{\mathbf{x}}$, so the Hill-modified algorithm is a generalization of the above algorithm.

Another way to modify the algorithm is related to dividing the original message into blocks and applying different shuffling functions to different blocks. Moreover, after the first shuffling, it is possible to perform a second division and shuffle again, which generalizes the application of the superposition operation of shuffling functions. For example, let $m=16$. First, we divide the message into blocks 4-8-4 and apply three mixing functions to each block. Then we divide the resulting message into blocks 8-8 and again apply a pair of new mixing functions to each block. In total, there are possible options for using permutation matrices $(4\,!)^2(8\,!)^3>10^{16}$, for breaking the mixing operator built using mixing functions, a complete enumeration of options is required.

Mixing images

Let's apply the mixing function to encrypt images. Let's take the classical image "Lena" (Fig. 8, *a*), which consists of 561×2^{10} pixels. Let's build 561×2^{10} matrix that encodes the image. Matrix elements are calculated to 16 decimal places. Round off the matrix elements to two decimal places, thus obtaining a truncated image (Fig. 8, *b*). Then round the matrix elements to one decimal place (Fig. 8, *c*). In the last case a significant loss of information is noticeable, the quality of the image has significantly deteriorated.

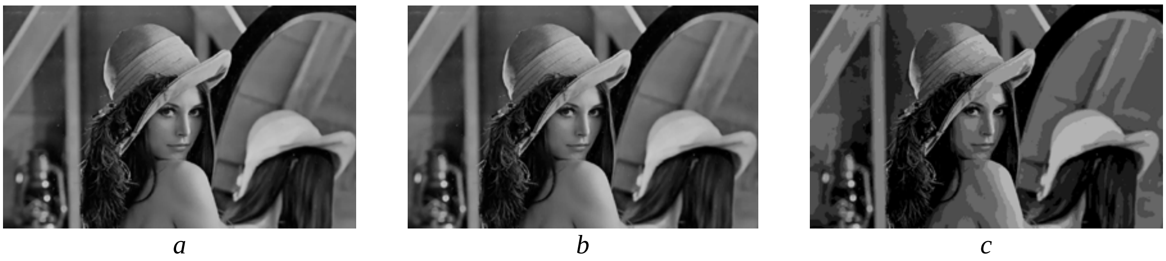

Fig. 8. Lena image: *a* – original, *b* – truncated (to two decimal places), *c* – truncated (to one decimal place)

Multiply the elements of these matrices by 100 and represent them as one-dimensional arrays containing $561\cdot2^{10}$ two-digit numbers. Or as arrays containing $561\cdot2^{11}$ single digit numbers. Let's split the arrays into blocks of eight numbers. There are $561\cdot2^{8}$ blocks of these in each array. For each element of a block, i.e. eight-digit numbers, apply the mixing function. We will get a new set of blocks of two arrays, from which we will compose matrices, and output images that are encoded by these matrices. We will get two mixing images of truncated (to two decimal places and to one decimal place) "Lena" images (Fig. 9, *a* and 10, *a*).

The same mixing function was applied to each block of both arrays. Parameters of the mixing function: $T_1=373$, $T_2=251$, $m_1=\{2,4,1,3,8,7,6,5\}$, $m_2=\{8,3,7,2,6,1,4,5\}$. Mixing matrices M_1 and M_2 are encoded by sequences m_1, m_2 (the ordinal number of the element in the sequence determines the row number, and the value of this element determines the column number, at the intersection of which a one stands in the permutation matrix). These mixing images are reversible without loss of information, i.e. the inverse images coincide with the original truncated images.

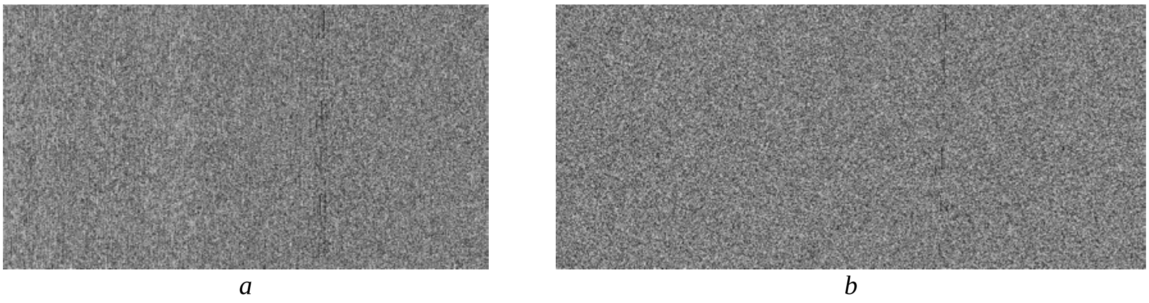

Fig. 9. Mixing the "Lena" image (truncated to two decimal places):
a – with splitting into blocks of 8 elements; *b* – with splitting into blocks of 24 elements

To improve the quality of mixing, let's split the arrays into blocks consisting of twenty-four numbers. There are $187 \cdot 2^8$ such blocks in each array. For each block element (a twenty-four-digit number), we apply the mixing function with the following parameters: $T_1 = 573$, $T_2 = 751$,

$$m_1 = \{13, 24, 20, 19, 9, 10, 2, 4, 23, 22, 1, 21, 16, 3, 18, 17, 8, 7, 15, 14, 6, 5, 12, 11\},$$

$$m_2 = \{16, 17, 8, 9, 18, 3, 10, 7, 14, 2, 23, 6, 11, 12, 1, 22, 21, 4, 13, 24, 5, 20, 19, 15\}.$$

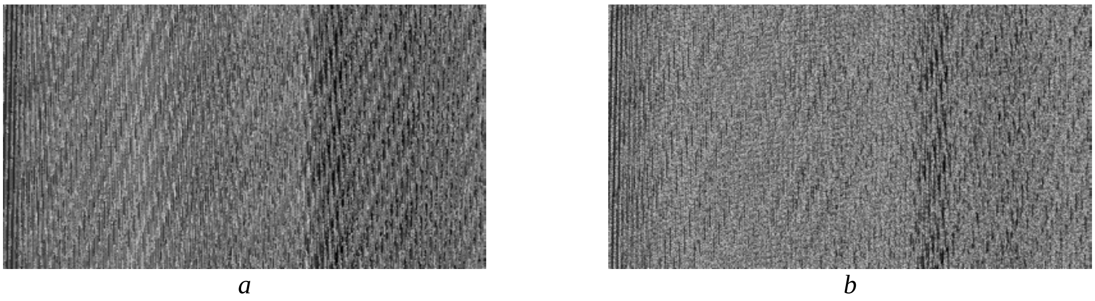

Fig. 10. Mixing of the "Lena" image (truncated to one decimal place):
a – with splitting into blocks of 8 elements, *b* – with splitting into blocks of 24 elements

In Figures 9 and 10 you can see how the mixing quality improves as the size of the array partitioning blocks corresponding to the matrices encoding the original image increases.

Conclusions

In this article, the possibility of using simple principles of dynamic systems theory in computer cryptography has been studied. The main problem with using chaotic systems in computer calculations is that the number of possible states in a computer is finite. Therefore, computer models of chaos are only an approximation of true chaotic behavior. Such an approximation can only convey the property of the original system in the initial iterations. Due to the sensitivity of the system to initial conditions, the trajectories of the original and approximated systems diverge very quickly, with each trajectory of the approximated system becoming periodic.

However, the use of the diffusion properties of nonlinear dynamic systems in spaces with a finite number of states is possible. Moreover, even with the simplest nonlinear mapping, the Tent map, it is possible to construct mixing functions that uniquely map the state space to itself, with good diffusion properties. Superpositions of the nonlinear Tent mapping and the permutation linear mapping (more generally, the Hill mapping) were used for this purpose. For the constructed functions, a correlation analysis, sensitivity analysis, and most significantly, an analysis of cycle lengths that divide the state space into non-intersecting subsets were performed in the case of spaces with a finite number of states. As a result, the expected good diffusion properties of these mixing functions were confirmed. The possibility of using these functions for image encryption was demonstrated. These functions can also be applied in local networks, simplifying mixing and encryption algorithms, which should significantly reduce computation time without losing network cryptostability. Based on the obtained mixing functions, it is planned to construct cryptographic hash functions in the future.

Література

1. Shannon C. E. Communication Theory of Secrecy Systems. *The Bell System Technical Journal*. 1949. Vol. 28, Iss. 4. P. 656–715.
2. A Novel Asymmetric Hyperchaotic Image Encryption Scheme Based on Elliptic Curve Cryptography / H. Liang, G. Zhang, W. Hou, P. Huang, B. Liu, S. Li. *Appl. Sci.* 2021. 11(12), 5691. DOI: https://doi.org/10.3390/app11125691.
3. Talhaoui M.Z., Wang X., Midoun M.A. A new one-dimensional cosine polynomial chaotic map and its use in image encryption. *Vis Comput*. 2021. 37. 1757–1768. DOI: https://doi.org/10.1007/s00371-020-01936-z.
4. A Novel Hybrid Secure Image Encryption Based on Julia Set of Fractals and 3D Lorenz Chaotic Map / F. Masood, J. Ahmad, S.A. Shah, S.S. Jamal, I. Hussain. *Entropy*. 2020. *22*(3). 274. DOI: https://doi.org/10.3390/e22030274.

5. Alvarez G. Some basic cryptographic requirements for chaos-based cryptosystems. *International J. of Bifurcation and Chaos*. 2006. Vol. 16(8). P. 2129–2151.
6. Devaney R.L. An Introduction to Chaotic Dynamical Systems. Second edition. New York : Addison-Wesley Publ. Co., 1993. 363 p.
7. Ott E., Grebodgi C., Yorke J.A. Controlling chaos. *Phys. Rev. Lett.* 1990. 64. 1196–1199. DOI: https://doi.org/10.1103/PhysRevLett.64.1196.
8. Biham E. Cryptanalysis of the Chaotic-Map Cryptosystem. Advances in Cryptology. EUROCRYPT'91; ed. DW Davies. LNCS 547. Berlin : Springer-Verlag, 1991. 532 p.
9. Bauer F.L., Friedrich Ludwig. Decrypted Secrets: Methods and Maxims of Cryptology. 4th edition. New York, USA : Springer, 2006. 474 p.
10. Hill L.S. Cryptography in an Algebraic Alphabet. *The American Mathematical Monthly*. 1929. Vol. 36, No.6. P. 306–312.
11. Hill L.S. Concerning Certain Linear Transformation Apparatus of Cryptography. *The American Mathematical Monthly*. 1931. Vol.37. P. 135–154.
12. Feistel H. Cryptography and Computer Privacy. *Scientific American.* 1973. Vol. 228, No. 5. P. 15–23.
13. Teh J.S., Alawida M., and Sii Y.C. Implementation and practical problems of chaos-based cryptography revisited. *Journal of Information Security and Applications*. 2020. Vol. 50. Article 102421.
14. Devaney R. L. An Introduction to Chaotic Dynamical Systems. 2nd Edition. C.R.C. Press, 1989. 360 p.

References

1. Shannon, C. E. (1949). Communication Theory of Secrecy Systems. *The Bell System Technical Journal, 28*, 4, 656–715.
2. Liang, H., Zhang, G., Hou, W., Huang, P., Liu, B., & Li, S. (2021). A Novel Asymmetric Hyperchaotic Image Encryption Scheme Based on Elliptic Curve Cryptography. *Appl. Sci., 11*(12), 5691. DOI: https://doi.org/10.3390/app11125691.
3. Talhaoui, M.Z., Wang, X., & Midoun, M.A. (2021). A new one-dimensional cosine polynomial chaotic map and its use in image encryption. *Vis Comput*, *37*, 1757–1768. DOI: https://doi.org/10.1007/s00371-020-01936-z.
4. Masood, F., Ahmad, J., Shah, S.A., Jamal, S.S., & Hussain, I. (2020). A Novel Hybrid Secure Image Encryption Based on Julia Set of Fractals and 3D Lorenz Chaotic Map. *Entropy, 22*(3). 274. DOI: https://doi.org/10.3390/e22030274.
5. Alvarez, G. (2006). Some basic cryptographic requirements for chaos-based cryptosystems. *International J. of Bifurcation and Chaos*, *16*(8), 2129–2151.
6. Devaney, R.L. (1993). An Introduction to Chaotic Dynamical Systems. Second edition. New York: Addison-Wesley Publ. Co.
7. Ott, E., Grebodgi, C., & Yorke, J.A. (1990). Controlling chaos. *Phys. Rev. Lett., 64*, 1196–1199. DOI: https://doi.org/10.1103/PhysRevLett.64.1196.
8. Biham, E. (1991). *Cryptanalysis of the Chaotic-Map Cryptosystem. Advances in Cryptology*. EUROCRYPT'91; ed. DW Davies. LNCS 547. Berlin: Springer-Verlag.
9. Bauer, F.L., & Friedrich Ludwig. (2006). *Decrypted Secrets: Methods and Maxims of Cryptology*. 4th edition. New York, USA: Springer.
10. Hill, L.S. (1929). Cryptography in an Algebraic Alphabet. *The American Mathematical Monthly, 36*, 6. 306–312
11. Hill, L.S. (1931). Concerning Certain Linear Transformation Apparatus of Cryptography. *The American Mathematical Monthly*, *37*, 135–154.
12. Feistel, H. (1973). Cryptography and Computer Privacy. *Scientific American, 228,* 5, 15–23.
13. Teh, J.S., Alawida, M., & Sii, Y.C. (2020). Implementation and practical problems of chaos-based cryptography revisited. *Journal of Information Security and Applications*, *50,* 102421.
14. Devaney, R. L. (1989). *An Introduction to Chaotic Dynamical Systems.* 2nd Edition. C.R.C. Press.

Грибняк Сергій Сергійович; Sergii Grybniak, ORCID: https://orcid.org/0000-0001-6817-8057
Дмитришин Дмитро Володимирович; Dmytro Dmitrishin, ORCID: https://orcid.org/0000-0002-2291-2364

Received October 02, 2022

Accepted November 29, 2022