Journal homepage: https://itce.vn.ua/en

Information Technologies and Computer Engineering **Vol. 22, No. 3. 2025**

Article's History: Received: 30.07.2025 Revised: 30.10.2025 Accepted: 23.12.2025

UDC 004.31: 004.056.5 DOI: 10.31649/vitce/3.2025.77

Image encryption and distribution method based on LFSR and counters

Volodymyr Luzhetskyi*

Doctor of Technical Sciences, Professor
Vinnytsia National Technical University
21021, 95 Khmelnytske Shose Str., Vinnytsia, Ukraine
https://orcid.org/0000-0001-7466-7738

Mykyta Tsikhotskyi

Postgraduate Student
Vinnytsia National Technical University
21021, 95 Khmelnytske Shose Str., Vinnytsia, Ukraine
https://orcid.org/0009-0005-8101-3536

Abstract. In the conditions of processing large amounts of graphic data, the task arises of developing a reliable image encryption scheme with reduced computing costs. The purpose of the study was to develop a deterministic scheme for encrypting and evenly distributing vectorised images using a shift register with linear feedback and counters. Methods of research included converting a pixel matrix to a sequence of bytes using a row-wise traversal rule, splitting the index space into equal subranges, generating pseudo-random indexes based on shift register states, and using reversible counters. The results of statistical testing demonstrate the stable characteristics of the proposed image encryption method. Encrypted test images were also evaluated for attack resistance by determining correlation coefficients between the incoming image and the encrypted one. In particular, for coloured images with a size of 512 × 512, when divided into eight subranges, the number of pixel change rate reached 99.61%, and the unified average intensity of pixel change was 32.28%, which corresponds to the upper cluster of estimates of advanced methods. The entropy of encrypted data was close to the theoretical maximum of 7.999, and the correlation between neighbouring pixels was significantly reduced and approaches zero values. Image distribution and restoration was performed without errors. The algorithm was characterised by low computational costs. The practical significance of the study consisted in ensuring reproducibility of the distribution and high cryptographic stability using mathematically simple operations, pseudo-randomness, and expanding the image encryption space to the full volume, making the proposed approach suitable for systems requiring accurate recovery and operating under limited computational resources

Keywords: secret distribution; image recovery; permutation; substitution; pseudo-random number sequence generator; image pixel correlation

Introduction

Given the current pace of information technology development, the use of digital data is growing exponentially. In accordance with the development of technologies, the requirements for protecting data that is stored, transmitted, and edited are also growing. Currently, there are many different approaches and methods that allow protecting information in the form of files of different formats. But among the file types, there is a separate category – these are images for which the use of conventional encryption methods is not appropriate, given the structure and large amount of data. Images can often contain very sensitive data, especially when they are used in critical areas of human life – medicine, military affairs, public and private secrets, etc. (Eichelberg *et al.*, 2020). Thus, contemporary science faces an important task of developing methods for protecting images in its various states, which can ensure a sufficient level of confidentiality and integrity of graphic data. In view of this, the relevance of the research is to ensure a cryptographically stable process of image encryption and uniform distribution of fragments in systems with

Suggested Citation:
Luzhetskyi, V., & Tsikhotskyi, M. (2025). Image encryption and distribution method based on LFSR and counters. *Information Technologies and Computer Engineering*, 22(3), 77-88. doi: 10.31649/vitce/3.2025.77

*Corresponding author

limited computing resources, which is especially important for Internet of Things (IoT) devices, embedded systems, and mobile solutions.

Conventional image encryption approaches based on the classic symmetric AES and DES algorithms guarantee a high level of protection, but require significant computational costs, which makes them of little use for mobile devices or IoT platforms. Therefore, there was a need for lightweight methods that can provide image encryption at minimal cost, while preserving cryptographic properties. A. Ihsan & D. Nurettin (2023) proposed a scheme that combines an affine transformation, a linear-feedback shift register (LFSR), and an XOR operation – this reduces the overhead complexity of the algorithm by using simple operations.

One of the most common encryption techniques is permutation schemes, in which information is protected by changing the order of pixels without changing their values (Babenko *et al*., 2021). Such methods require less computational resources compared to substitution or chaotic ciphers, since instead of complex nonlinear transformations, it is implemented only by replacing indexes. In most studies, this method was used in isolation: in particular, Y.-J. Sun *et al*. (2021) proposed an efficient permutation scheme based on two-dimensional logistic mapping for image encryption, which provides high entropy and resistance to statistical attacks, but has an increasing time complexity in processing large images, which limits its application in systems with strict performance requirements.

A promising area is also the use of chaotic systems for building complex multi-level encryption algorithms. Thus, Z. Liu *et al.* (2025) proposed a method for encrypting coloured images based on a discrete wavelet transform and a hyperchaotic dynamical system that combines several stages of permutation and diffusion. However, the high algorithmic complexity of the method in ($O(N \log N)$) limits the effectiveness of its application for high-resolution images.

Another area of research concerns the use of LFSR as sources of pseudorandom sequences (PRS) in image encryption schemes. LFSRs are characterised by high performance and simple hardware implementation, which makes them attractive for low-resource devices and IoT solutions. Appropriate hardware evaluations and comparisons of implementations on the Field-Programmable Gate Array (FPGA) confirm the advantages of LFSR in terms of logic element usage and performance (Dridi *et al*., 2023). In a number of applications where pseudo-random sequence (PRS) generators built using LFSR have known cryptanalysis vulnerabilities and limitations on the entropy and correlation properties of sequences compared to some chaotic generators, which requires additional measures (for example, combining multiple LFSR, nonlinear adders, or cascading with other PRS generators). Because of such trade-offs, hybrid approaches are emerging in practice – combining LFSR (as a fast hardware component) with cryptographic blocks and chaotic modules (for example, using DNA as a source of chaos and combining it with Advanced Encryption Standard (AES) components) – which increase cryptographic resistance, but significantly increase hardware and time complexity, making them less suitable for devices with limited resources. Such algorithms were presented in the papers by R. Ettiyan & V. Geetha (2023) and K. El Kinani *et al*. (2025).

An important task in encryption schemes is to divide the encrypted image into fragments so that no fragment itself contains enough information to restore the original, and reconstruction became possible only if the necessary set of parts is available. Practical implementations of block encryption often combine block permutations with chaotic diffusion, which provides flexible segmentation and localised access control, but requires the preservation of service metadata about the size and order of blocks and related access rights, which complicates exchange protocols and increases network load, one of these implementations was proposed by N. Wang *et al*. (2024). Contemporary approaches also experiment with dynamic geometric fragmentation schemes (square, L-shaped, and other geometric block extractions), which reduce the number of fragments or increase the degree of scrambling, but increase the computational complexity of the optimal splitting and recovery stage. P. Oikonomou *et al*. (2025) proposed a square image distribution method. A separate group of methods consists of approaches that use the fuzzy logic apparatus or fuzzy numbers to introduce additional uncertainty into the partitioning/distribution parameters; such methods can increase resistance to direct analysis of fragments, but do not always guarantee an even distribution of the amount of data between parts and require careful adjustment of the fuzzy rules. Y. Umadevi *et al*. (2022) proposed one of these methods of image hiding using fuzzy logic.

Thus, despite the presence of a wide range of image encryption methods, there are not enough solutions in the literature that would provide a deterministic pixel rearrangement within the entire image and simultaneously allow evenly distributing the resulting fragments without the need to save auxiliary metadata. The purpose of the study was to improve the encryption process and splitting the image into parts, with an emphasis on reducing the structural complexity of the algorithm. To achieve this goal, the following tasks were set: formalisation of the image vectorisation procedure; development of an encryption algorithm using LFSR and reversible counters; construction of a mechanism for evenly dividing the vector into n fragments and reverse image recovery. Simultaneously, the algorithm must be deterministic and meet the general requirements for evaluating the quality of encryption. The scientific originality lies in the integration of efficient LFSR components and counters to form a deterministic encryption algorithm that reduces the structural complexity of the algorithm and performs pixel rearrangement throughout the image.

Materials and Methods

The methodological component of the study was aimed at formalising and implementing an encryption algorithm and evenly distributing images into parts. It included

several stages: development of a mathematical model, software implementation, selection of a test set of images, performance evaluation based on the criteria of correct recoverability, cryptographic quality, and time complexity. The proposed image distribution algorithm contained four main methods: splitting the image into a vector; using counters and LFSR; the image (vector) encryption process itself; splitting the encrypted image into parts and the reverse recovery process.

For convenient and efficient image processing, it was proposed to perform the process of dividing pixels into a vector. Image I was defined as a matrix of pixels of size $h \times w$, where h – height, w – width. For a greyscale image, each element $I_{x,y} \in \{0, 1,..., 255\}$, where $x \in \{0,..., w\text{-}1\}$, $y \in \{0,..., h\text{-}1\}$. For a coloured image, each pixel had a triple value $I_{x,y} = (R_{x,y}, G_{x,y}, B_{x,y})$, where R – red intensity, G – green intensity, B – blue intensity, and each channel took on a value with {0, 1,..., 255}.

Transformation of matrix I in vector V was as follows: for greyscale images, the vector was $V = (V_1, V_2,..., V_N)$, where $N = h \times w$, $V_k = I_{x,y}$ for $k = y \cdot w + x$; for coloured images, there were two options for converting the image matrix I into vector V.

Option 1. Three pixel components were selected sequentially. $V = (V_1, 1, V_{1,2}, V_{1,3}, V_{2,1}, V_{2,2}, V_{2,3},..., V_{N,1}, V_{N,2}, V_{N,3})$, where $V_{k,1} = R_{x,y}$, $V_{k,2} = G_{x,y}$, $V_{k,3} = B_{x,y}$ for $k = y \cdot w + x$.

Option 2. A separate vector was formed for each pixel component, and then these vectors were combined into a single vector. $V = (V^R, V^G, V^B)$ where $V^R = (V_1^R, V_2^R,..., V_N^R)$, $V_k^R = R_{x,y}$, $V^G = (V_1^G, V_2^G,..., V_N^G)$, $V_k^G = G_{x,y}$, $V^B = (V_1^B, V_2^B,..., V_N^B)$, $V_k^B = B_{x,y}$.

These options provided for the selection of pixels in their natural display in files, namely by row-major order. The following is a formalisation of the encryption algorithm for the vector obtained after reading the image. This representation provides an unambiguous reflection of the two-dimensional image structure in a linear form, which allows correctly applying further cryptographic transformations to the vector and guarantees reproducibility of all stages of the algorithm.

Image encryption algorithm

C.E. Shannon (2001) showed that the encryption procedure can be represented as a combination of two basic transformations – permutation (diffusion) and substitution (confusion). A common approach to encrypting messages is to break them down into blocks and implement a permutation of message characters within only each block, and not within the entire message. The researchers suggest rearranging pixels not within the image blocks, but within the entire image. This increases the resistance to cryptanalysis, since the number of possible pixel permutations in the entire image is significantly greater than the number of possible permutations in the block.

As part of the development of the image encryption algorithm, two options for implementing sequential substitution and permutation were considered:

1) First substitution, then permutation (Fig. 1). Elements of the input vector are processed sequentially and, accordingly, substitution and permutation operations are performed sequentially at each step. In this approach, the permutation rule is determined by the function π:

$$\pi^{SP} = \begin{pmatrix} 0,1,...,N-1 \\ p_0,p_1,..., p_{N-1} \end{pmatrix}, \quad (1)$$

where S – substitution; P – permutation; p – element number in the encrypted vector according to the permutation rule.

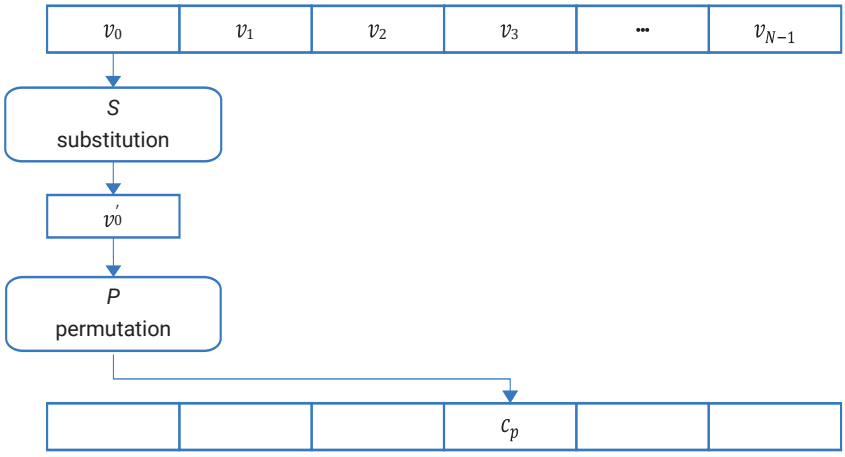


Figure 1. SP-type encryption procedur

Source: compiled by the authors

2) First the permutation, then the substitution (Fig. 2). In the second version of the encryption algorithm implementation, the input vector elements are first selected according to the permutation rule and written to the next current position (starting from zero) after applying the substitution operation. The function π has a slightly different reflection in this implementation:

$$\pi^{PS} = \begin{pmatrix} p_0,p_1,..., p_{N-1} \\ 0,1,...,N-1 \end{pmatrix}. \quad (2)$$

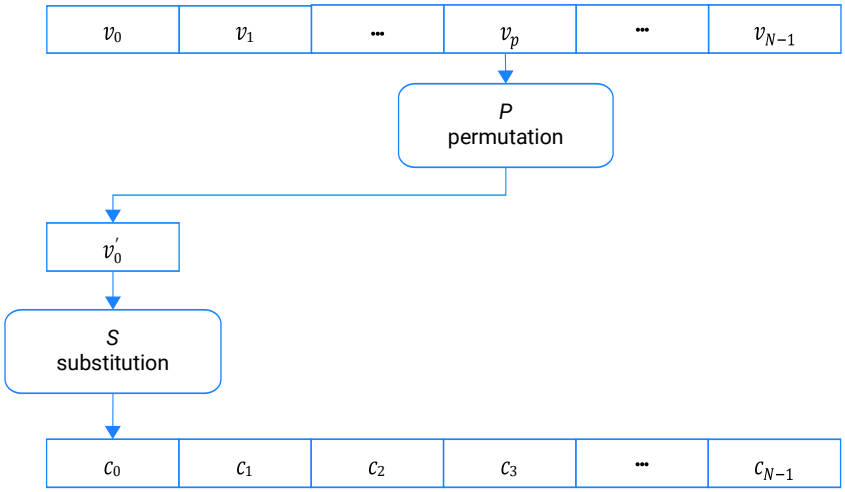


Figure 2. PS-type encryption procedure

Source: compiled by the authors

In the figures above, elements of the vector $V = \{v_0, v_1, v_2,..., v_{N-1}\}$ define elements of the image input vector; $V' = \{v'_0, v'_1, v'_2,..., v'_{N-1}\}$ – elements of the vector after performing substitution/permutation operations, $C = \{c_0, c_1, c_2,..., c_{N-1}\}$ – elements of the encrypted vector. Further, to present the proposed encryption method and its further testing, the first encryption method was chosen, namely, first performing a substitution, and then a permutation.

In this paper, it was proposed to perform substitution according to the following rule:

$$S^{SP}_{K_2}(V) = (V_i + Z(t)) \, mod \, 256;$$
$$S^{PS}_{K_2}(V') = (V'_i + Z(t)) \, mod \, 256, \quad (3)$$

where $Z(t) = (aX(t) + bY(t) + c) \bmod 256$. Here a, b, c – odd numbers that are determined by the secret key. $X(t) = (R(t) \ll 8)$ & 0xFF, where $\ll$ – means a cyclic shift to the left by 8 positions of the register status code $R(t)$. $Y(t) = (R^*(t) \ll 8)$ & 0xFF, where $R^*(t) = (R(t) \ll 8)$. This substitution acts as byte-by-byte data masking.

To implement pixel rearrangement within the entire image, it is necessary to generate pseudorandom numbers that will correspond to the new pixel position numbers p in the image. Pseudo-randomness of the sequence of numbers is ensured by using a secret key. V.A. Luzhetskyi & I.S. Horbenko (2013) proposed a generalised approach to constructing such number generators in the range from 0 to N-1. It was adapted to the conditions for solving the problem of rearranging pixels in the entire image.

Range of numbers from 0 to N-1 is split into l subranges. If $d = \frac{N}{l}$ is an integer, then each of the subranges consists of d-numbers. Each subrange is defined by the minimum and maximum values of the number:

$$D_1 = [d^1_{min}, d^1_{max}]; \quad (4)$$

If d is not an integer, then (l-1) subranges consist of $\left]\frac{N}{l}\right[$ numbers, and l-th subrange contains $N - (l-1)\left]\frac{N}{l}\right[$ numbers.

In the process of generating a pseudo-random sequence of numbers, numbers are selected from a specific subrange, the number of which is determined by the code generated by the shift register with linear feedback. The initial state of the shift register is set by the secret key K_1. The subrange number will be set r-bit binary code. With this in mind, the number of subranges is $l = 2^r$.

The selection of numbers from the subrange is carried out in a deterministic way. There are two possible options. The first is to form a sequence of numbers starting with d^1_{min}, increasing the number at each step by one. The second option is to form a sequence of numbers starting with d^1_{max}, reducing the number by one at each step. These operations are proposed to be implemented on the basis of a reversible counter CT. The counter of each subrange generates numbers from 0 to $d-1$ or from $d-1$ to 0. The availability of two options provides additional resistance to tampering, as the initial and final states can be defined separately for each counter. The initial value of the counter state is determined by the secret key component $K_2 = \{k_{2,j}\}$, where $j = 0,1,...,l-1$. Table 1 shows an example of the initial states of counters for a key $K_2 = \{0,1,1,0\}$.

Table 1. Example of the initial states of counters

K_2			
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$
	1		0
$CT_0 := 0$	$CT_1 := d-1$	$CT_2 := d-1$	$CT_3 := 0$

Source: compiled by the authors

Changing the counter state is determined by the following rule:

$$CT_j(t+1) = CT_j(t) + (-1)^{k_2, j}, \quad (5)$$

where $CT_j(t)$ – status of j-th counter in step t.

The number selected from j-th subrange is determined based on the subrange number and the state of the corresponding counter:

$$p = j \cdot d + CT_j(t). \quad (6)$$

When using counters together with the PRS generator in the way described above, a collision may occur when in the current subrange of the vector C all numbers have already been selected in the previous steps, and the generator points to this subrange again. This situation can distort the process of encrypting/decrypting the vector by overwriting the previously selected number, which will make it impossible to restore the input image. To eliminate this, an additional mechanism for checking the current state of the counter is proposed. If $CT_j(t) = d - 1$ when $k_{2,j} = 0$ or $CT_j(t) = 0$ when $k_{2,j} = 1$, then forming the corresponding number from the subrange j does not happen. The process of forming a pseudo-random sequence of numbers is completed when the above conditions are met for all counters. Considering the specifics of implementing substitution and permutation operations using a certain secrecy, the secret key K must have four components: k_1 – initial state code of the shift register (64 bits); k_2 – a set of 8-bit parameter codes *a, b, c* for the substitution rule; k_3 – l-bit binary code that defines the initial states of counters and the rule of their operation. Thus the bit depth of the secret key K is equal to $(88 + l)$ bit.

Operations that implement substitution and permutation rules to perform the image encryption process have been described and defined above. This process is now presented as a single aggregate function to further simplify the description of image distribution, recovery, and decryption processes. Let the encryption for the *SP* method be described as:

$$E_K^{SP}(\mathrm{V}) = S_{K_2}^{SP}(\mathrm{V}) \circ P_{K_1, K_3}(\mathrm{V}'), \quad (7)$$

where S_{K_2} – substitution function; P_{K_1, K_3} – permutation function.

Similarly, the encryption function is presented for the *PS* method:

$$E_K^{PS}(\mathrm{V}) = P_{K_1, K_3}(\mathrm{V}) \circ S_{K_2}^{PS}(\mathrm{V}'). \quad (8)$$

Thus, the authors proposed and described a new image encryption method that combines substitution and controlled permutation operations using a pseudorandom sequence generator based on a linear feedback shift register (LFSR). After encrypting the input image, vector $C = (c_0, c_1, \ldots, c_{N-1})$ is divided into n subvectors for the purpose of organising distributed storage without the ability to restore the full image if only a part of the subvectors is present. For this purpose, a secret distribution scheme of the type (n, n), which guarantees that only if all n parts are possible to restore the original content correctly, which increases the level of information security.

For subvector distribution, it is proposed to use a cyclic uniform distribution, which ensures uniform data filling of each subvector and allows maintaining a linear correspondence between the elements of the original and encrypted vector, which simplifies the decryption process and minimises overhead calculations. This approach, in comparison with chaotic or random distribution schemes, is quite stable for structural analysis, but simultaneously remains controlled and deterministic, which is critical for further image restoration.

Each subvector of distributed image – $P_y = \{p_{y,0}, p_{y,1}, p_{y,2}, \ldots\}$, where y {0, 1,..., $n-1$} , is formed from the elements of the vector C for which the condition is met:

$$p_{y,q} = c_i, where\ y = i \bmod n, q = y + \left\lfloor \frac{i}{n} \right\rfloor \cdot n. \quad (9)$$

Thus, a sequence of subvectors is formed $\{P_0, P_1, \ldots, P_{n-1}\}$, the dimensions of which do not exceed $\frac{N}{n} \pm 1$. To restore the input image, a complete set of $\{P_0, P_1, \ldots, P_{n-1}\}$ parts is required.

Since the cyclic uniform approach was used for the distribution, the indexes of the elements of the encrypted vector were assigned in a deterministic order, which preserves a one-to-one correspondence between the elements of each part P_y and their positions in C. This allows generating a reverse transformation that restores C by simply combining fragments with fixed indexes according to the original distribution order.

For each part $P_y \subset C$, restoring an element of vector C occurs using the equation:

$$c_i = p_{y,q}\ , where\ y = i \bmod n;\ q = y + \left\lfloor \frac{i}{n} \right\rfloor \cdot n. \quad (10)$$

After assembling the parts from encrypted vector C to restore the image, sequential substitution and permutation operations are performed, the order of which depends on the original encryption method – *SP* or *PS*. Since the substitution operation S_{K_2} is bijective, its inverse $S_{K_2}^{-1}$ restores information before performing a substitution. Accordingly, the inverse substitution operation $S_{K_2}^{-1}$ is performed as follows:

$$S_{K_2}^{-1, SP}(V') = (V_i' - Z(t)) \ mod\ 256; \quad (11)$$

$$S_{K_2}^{-1, PS}(C) = (C_i - Z(t)) \ mod\ 256. \quad (12)$$

Below is a mathematical representation of the decryption function $D_K(C)$, as a result of which the initial vector V is obtained:

$$D_K^{SP}(\mathrm{C}) = P_{K_1, K_3}(\mathrm{C}) \circ S_{K_2}^{-1, SP}(V'); \quad (13)$$

$$\mathrm{D}_K^{PS}(\mathrm{C}) = S_{K_2}^{-1}, PS\,(\mathrm{C}) \circ P_{K_1, K_3}(V'). \quad (14)$$

The last step is to restore the image from the resulting vector by folding it back into pixels and presenting it as an image file. This step involves accurately reproducing the spatial structure of the image based on the stored index order generated during encryption. As a result of folding the provided parts and decrypting the image, the initial secret image is formed.

Results and Discussion

During the experiments, the cryptographic characteristics of the proposed algorithm were evaluated on test sets for greyscale and coloured images with a size of 512 × 512 pixels (.tiff format). The following secret key parameters are selected for testing: K_1 = 0xA1B2C3D4E5F60708, K_2 = {5, 7, 11}, K_3 = {0, 1, 0, 1} for 4 subranges and K_3 = {0, 1, 0, 1, 1, 0, 0, 1} for 8 subranges. The feedback positions in the shift register are defined as [63, 62, 60, 59]. For this purpose, statistical testing of the encryption method was performed using the National Institute of Standards and Technology (NIST SP 800-22) statistical test kit (Rukhin *et al.*, 2010). Table 2 shows the results of statistical tests that were performed on a 2 Mbit encrypted data sequence. To ensure sufficient data sampling, 12 greyscale images were encrypted and sequentially written to the test file as bits, and up to 4 test images were used for coloured images.

Table 2. Results of statistical testing using NIST SP 800-22

Coloured 4		Coloured 8		Greyscale 4		Greyscale 8		Statistical test
P	Fraction	P	Fraction	P	Fraction	P	Fraction	
0.534	9/10	0.739	10/10	0.534	10/10	0.122	10/10	Frequency
0.534	10/10	0.35	10/10	0.739	9/10	0.739	10/10	BlockFrequency
0.442	9/10	0.545	10/10	0.545	10/10	0.094	10/10	CumulativeSums
0.534	10/10	0.739	10/10	0.534	10/10	0.122	10/10	Runs
0.213	10/10	0.911	9/10	0.122	10/10	0.534	10/10	LongestRun
0.213	10/10	0.213	10/10	0.534	10/10	0.739	10/10	Rank
0.35	10/10	0.911	10/10	0.35	10/10	0.739	10/10	FFT
0.35	10/10	0.35	10/10	0.008	10/10	0.911	10/10	Overlapping template
0.066	10/10	0.213	9/10	0.350	10/10	0.35	9/10	Universal
0.122	10/10	0.911	10/10	0.122	10/10	0.911	10/10	ApproxEntropy
0.739	10/10	0.534	10/10	0.534	10/10	0.739	10/10	LinearComplexity
–	10/10	–	10/10	–	10/10	–	10/10	Serial median
–	9.91/10	–	9.89/10	–	9.86/10	–	9.91/10	NonOverlapping template median
–	5/5	–	3.88/4	–	7/7	–	6/6	RandomExcursions median
–	5/5	–	3.89/4	–	7/7	–	6/6	RandomExcursions variant median

Source: compiled by the authors

The obtained results of statistical testing demonstrate the stable characteristics of the proposed image encryption method. For most tests, P values exceed the set significance threshold $\alpha = 0.01$, which indicates that there are no detected signs of statistical non-randomness in the generated bit sequences. This indicates that the method provides a sufficient level of entropy and uniformity of distribution, which are key criteria for cryptographically stable transformations. For coloured images with an increased number of subranges, there is a slight increase in the uniformity of bit distribution and a more uniform distribution of P over the intervals. This indicates improved diffusion as the number of subranges increases, which reduces the likelihood of local structures appearing in the encrypted data. In greyscale images with fewer subranges, the algorithm shows only slightly lower performance for individual tests (for example, in the case of frequency and block frequency tests), but these values still remain within the acceptable range ($P \geqslant \alpha$). The Fraction value indicates the number of successfully completed tests. According to the results, almost all tests were passed successfully, a deviation of 1 unsuccessful test is acceptable. Figure 3 shows a histogram of the distribution of coefficients depending on the image type and the specified number of subranges.

Figure 4 shows the pixel brightness distribution (in the range of values from 0 to 255) for the input greyscale image. The nature of the curve indicates the presence of pronounced peaks and troughs, which reflects the predominance of certain brightness levels. Such unevenness is typical for natural or visually understandable images, since their structure and content form statistical patterns that can be used by an attacker to simplify cryptanalysis. Figure 5 shows the distribution for the encrypted image, which is almost uniform, which indicates a high level of randomness of the received data, which significantly complicates their cryptanalysis and is a sign of a stable encryption algorithm.

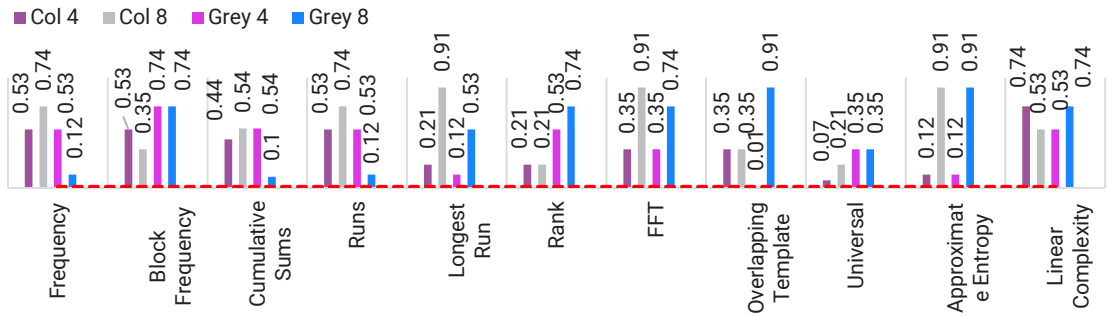


Figure 3. Histogram of NIST statistical testing results
Note: red line indicates the permissible threshold value $\alpha = 0.01$
Source: compiled by the authors

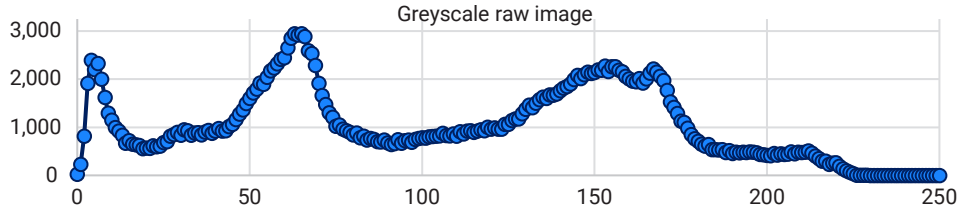


Figure 4. Pixel brightness distribution in the input greyscale image
Source: compiled by the authors

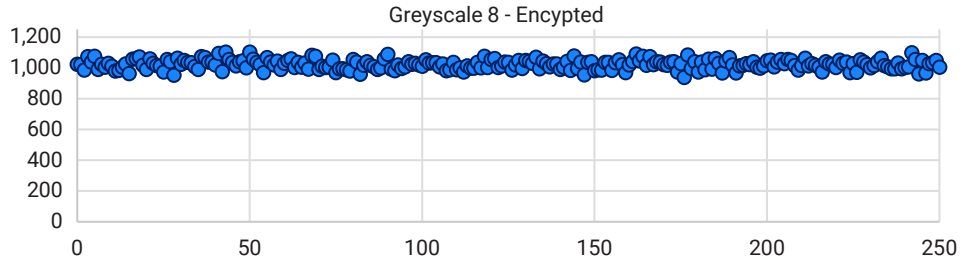
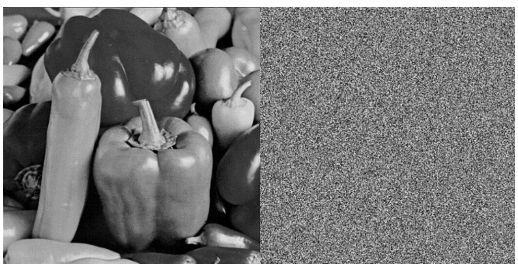


Figure 5. Pixel brightness distribution in the encrypted greyscale image
Source: compiled by the authors

Figure 6 shows the results of performing an encryption scheme, where the input image has a natural structure and easily recognisable content, and the encrypted image visually resembles random noise without any noticeable dependencies. This type of encrypted image is typical for systems with efficient diffusion, where each change in input pixels significantly affects the output result. This indicates that the algorithm is reliably resistant to attacks based on visual analysis.



A B

Figure 6. Encryption results for the greyscale Peppers test image
Note: A – input image; B – encrypted image
Source: compiled by the authors

For coloured images, the brightness distribution was analysed separately for each of the three colour channels (*R, G, B*), which allows identifying specific features of their structure. As can be seen from Figure 7, the input image has pronounced peak values in each channel, reflecting the uneven colour distribution and the presence

of dominant shades. The encryption result (Fig. 8) demonstrates that all channels have a uniform distribution, which indicates an effective destruction of the initial correlations between pixels in each of the channels and makes it impossible to restore input characteristics based on statistical analysis.

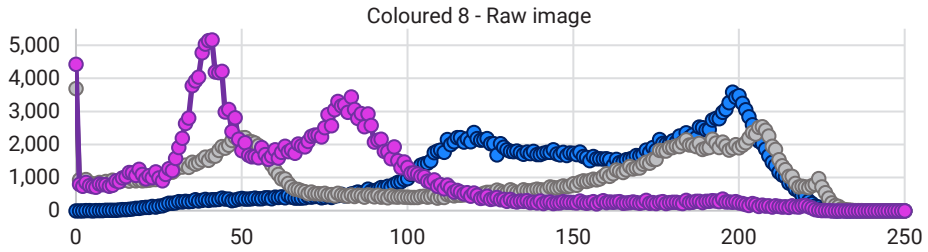


Figure 7. Pixel brightness distribution in the input coloured image
Source: compiled by the authors

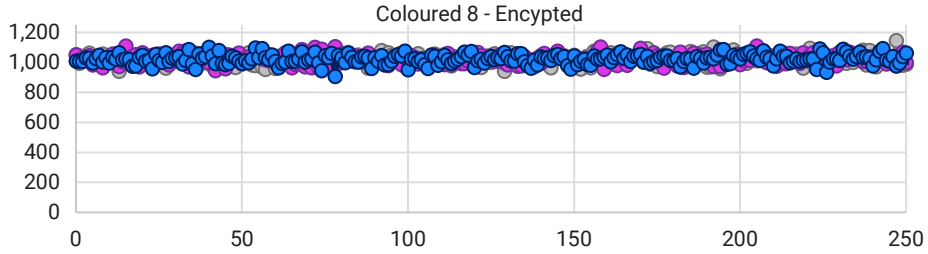


Figure 8. Pixel brightness distribution in the encrypted coloured image
Source: compiled by the authors

Similar to the greyscale example, encryption turns the coloured input image into noise and makes it impossible to recognise the secret (Fig. 9). This visual transformation confirms the efficiency of diffusion in the algorithm, which is a critical factor for resistance to visual analysis. It also demonstrates the absence of a residual structure that could have been used to reconstruct the original image.

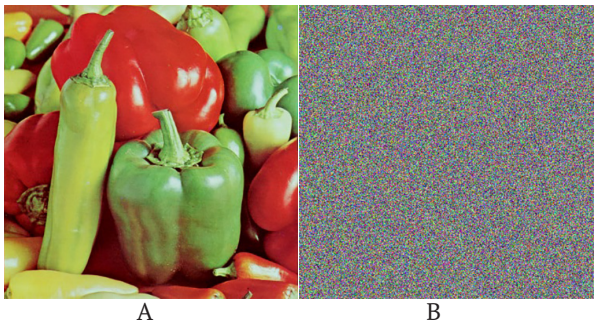


Figure 9. Encryption results for the coloured Peppers test image
Note: A – original image; B – encrypted image
Source: compiled by the authors

Since the specifics of image encryption differ significantly from the process of encrypting ordinary bit data (text), methods for evaluating cryptographic stability also require some adaptation. Performing statistical testing inherent in bit sequence encryption does not always fully reflect the key characteristics of encrypted images, in particular, their resistance to attacks. Therefore, in addition to NIST tests, the correlation between neighbouring pixels is additionally analysed, which is described by the NPCR (Number of Pixels Change Rate) and UACI (Unified Average Changing Intensity) parameters. The NPCR indicator reflects the percentage of pixels that changed their value with a slight change in the input data, and is used to estimate the sensitivity of the algorithm to the initial conditions: values close to 100% indicate high stability, when even a change in one pixel in the original image leads to a completely unpredictable result. The UACI parameter determines the average intensity of pixel brightness changes between two images and reflects the degree of destruction of the initial correlations; according to

Y. Wu *et al*. (2011) and Y. Alghamdi & M. Arslan (2024), its optimal value for stable algorithms is in the range of 32-34%. In addition, the entropy of the initial and encrypted images was calculated as a measure of the randomness of the pixel distribution, which allows estimating the approximation of the result to a completely random one. The correlation requirements between pixels determine: the closer this value is to zero, the smaller the relationship. The parameters of cryptographic stability and pixel correlation are shown in Table 3.

Table 3. Parameters of pixel correlation and NPCR, UACI

		Coloured 4	Coloured 8	Greyscale 4	Greyscale 8
Entropy	original	7.6698	7.6698	7.5715	7.5715
	encrypted	7.9997	7.9997	7.9993	7.9993
Horizontal correlation	original	0.9704	0.9704	0.9792	0.9792
	encrypted	0.0004	0.0008	-0.0016	-0.0003
Vertical correlation	original	0.9715	0.9715	0.9826	0.9826
	encrypted	-0.0014	-0.0009	0.0006	0.0006
Diagonal correlation	original	0.9576	0.9576	0.9680	0.9680
	encrypted	0.0009	0.0006	-0.0002	0.0040
NPCR		99.61%	99.61%	99.61%	99.60%
UACI		32.18%	32.28%	31.00%	31.00%

Source: compiled by the authors

In the table below, the results are presented for four options for using the encryption scheme: coloured and greyscale images with 4 and 8 subranges, respectively. For coloured images, the entropy of encrypted data is close to the maximum theoretical value (7.9997), which indicates a high level of randomness, and the UACI of ~32.2% corresponds to the optimal range for protection against attacks that analyse local changes and dependencies between pixels of the input image and the encrypted one. The NPCR value consistently exceeds 99.6%, which demonstrates the high sensitivity of the algorithm to changing even one pixel. For greyscale, the entropy of encrypted data also shows results close to 8 (7.9993). UACI (~31%) also meets the sustainability requirements, while NPCR retains a value of approximately 99.6%. Comparative analysis shows that increasing the number of subranges from 4 to 8 has a positive effect on increasing the degree of randomness of the encrypted image. It is also noted that the method shows better results when working with colour images due to an increase in the natural difference in brightness distribution and colour depth compared to greyscale. Table 4 shows estimates of the NPCR and UACI values of the proposed encryption algorithm and other known encryption algorithms. For evaluation, a 512×512 coloured image was taken divided into 8 subranges, the encryption results of which showed the best results for the proposed method: NPCR = 99.61%, UACI = 32.28%.

Table 4. Comparison of NPCR and UACI (512 × 512 coloured images)

No.	Author	NPCR (%)	UACI (%)
1	Z. Liang *et al.* (2021)	99.6	33.3
2	X. Wang *et al*. (2019)	99.6	31.5
3	Y. Abanda & A. Tiedeu (2016)	99.6	32.05
4	B. Stoyanov & K. Kordov (2015)	99.5-99.7	31-32
5	Y. Alghamdi *et al*. (2022)	99.5-99.7	31-32
6	Authors' scheme	99.61	32.28

Source: compiled by the authors

Z. Liang *et al*. (2021) proposed a scheme based on a five-dimensional chaotic system using DNA coding and genetic operations. This study demonstrated NPCR and UACI at a fairly high level, which is explained by multi-level diffusion at the bit operation level and the use of chaos. Compared to the algorithm under study, this approach has a higher implementation complexity and a larger number of parameters to configure, which makes it difficult to use on platforms with limited resources. Such a scheme shows better diffusion results, the disadvantage is increased implementation complexity and higher computational costs.

X. Wang *et al*. (2019) proposed an encryption algorithm that uses an S-box formed from a chaotic sequence. This results in increased substitution nonlinearity and stable NPCR ≈ 99.6, UACI ≈ 31 – 32. The main difference from the presented algorithm is the use of nonlinear substitution. The algorithm provides sufficient statistical stability, but creating and verifying an S-box increases complexity and complicates hardware implementation.

Simple but reliable combinations of chaotic maps and "mixing" to generate permutations and diffusion are used in the study by Y. Abanda & A. Tiedeu (2016). The results show UACI ≈ 32.05% and NPCR ≈ 99.6. Such an algorithm provides stable results with minimal implementation complexity, but provides less control over determining the sequence of transformations and complicates formal proof of security.

B. Stoyanov & K. Kordov (2015) presented an encryption scheme using polynomial chaotic mappings, which showed UACI results in the range of 31-32% with a sufficient NPCR value. The difference is a mathematically more complex model of the PRS generator, which gives a wider key space. In this scheme, balanced statistical characteristics are noted at the same level as the author's, but it is worth considering the increase in hardware complexity of implementation.

Y. Alghamdi *et al*. (2022) considered algorithms for devices with limited resources, where local block permutations and simple chaotic maps are used. Test statistics showed NPCR ≈ 99.5 – 99.7 and UACI ≈ 31 – 32. This indicates that the algorithm is better adapted to hardware limitations compared to previous methods, but the correlation indicators are less than the required minimum and, accordingly, the author's scheme.

The obtained NPCR and UACI values showed that the proposed algorithm belongs to the group of advanced efficient solutions. NPCR = 99.61% indicates a high sensitivity of the encryption to minimal changes in the input image. UACI = 32.28% shows a good intensity of change, indicating effective diffusion, although this figure is slightly lower than in some examples with deeper multilevel diffusion (UACI ≳ 33%, which is the reference value). The results obtained indicate a balance between cryptographic stability and computational ease of implementation. This confirms the feasibility of using the algorithm in systems with strict resource constraints and the need for guaranteed and deterministic image recovery.

Conclusions

This paper proposed a deterministic scheme for encryption and uniform distribution of vectorised images based on a linear feedback shift register and controlled reversible counters. The main objectives – to provide linear time computational complexity, deterministic and reproducible partitioning without storing additional metadata, and to achieve a cryptographically acceptable level of randomness of encrypted data – were successfully achieved and experimentally validated. The proposed algorithm demonstrated linear complexity with respect to the number of pixels (O(N)), since each iteration requires one single invocation of the PRS generator and a constant set of counter update and write operations. The cyclic uniform distribution mechanism ensured uniform filling of subvectors and allowed restoring the original vector only if all n parts are available, thereby satisfying the requirements of an (n, n) secret distribution scheme. During reconstruction, zero reconstruction errors and no pixel degradation were observed when combining fragments and performing decryption.

The results of testing according to NIST SP 800-22 showed no pronounced signs of statistical non-randomness in the generated bit sequences (p values are greater than the threshold $\alpha = 0.01$ for most tests and the proportion of successfully passed tests met the required criteria). The entropy of the encrypted images approached the theoretical maximum of 7.999, which indicates a uniform distribution of pixel intensities. Correlation coefficients between adjacent pixels horizontally, vertically, and diagonally dropped from high values of 0.95-0.98 in the original images to values close to zero or small negative values in encrypted images (about 10^{-3}-10^{-4}), which confirmed the destruction of spatial dependencies. Attack resistance indicators based on the analysis of differences between open and encrypted images – NPCR > 99.6% and UACI 31-32% – met the generally accepted criteria for effective avalanche behaviour and sufficient diffusion between original-ciphertext pairs. Due to the simplicity of operations (bit shifts, modular arithmetic, such as addition with a modulus of 256) and minimal memory requirements, the proposed scheme is well suited for implementation on embedded platforms, mobile devices, and other systems with limited computational resources. The absence of the need to store metadata for fragment recovery reduces network and memory overhead in distributed storage.

Since some practices use a cryptographic threshold (k, n) a secret distribution scheme, so the prospects for further research are to develop a scheme that will allow effective application of the recovery mechanism to images encrypted by the proposed method. Future research should focus on optimising the implementation of the algorithm, as this will reduce encryption time, which is especially important for devices with limited resources or systems that transmit data in real time. It is also advisable to strengthen the diffusion properties in order to bring the UACI indicators closer to the reference level (~ 33%), while not complicating the overall structure of the algorithm, which will ensure a balance between safety and efficiency. In addition, advanced testing of the algorithm on various sets of images of different formats will allow for a deeper study of its resistance to statistical attacks, identify possible relationships between data types and encryption efficiency, and identify potential limitations or weaknesses of the proposed approach.

Acknowledgements

None.

Funding

The study received no funding.

Conflict of Interest

None.

References

[1] Abanda, Y., & Tiedeu, A. (2016). Image encryption by chaos mixing. *IET Image Processing*, 10(10), 742-750. doi: 10.1049/iet-ipr.2015.0244.

[2] Alghamdi, Y., & Arslan, M. (2024). Image encryption algorithms: A survey of design and evaluation metrics. *Journal of Cybersecurity and Privacy*, 4(1), 126-152. doi: 10.3390/jcp4010007.

[3] Alghamdi, Y., Munir, A., & Ahmad, J. (2022). A lightweight image encryption algorithm based on chaotic map and random substitution. *Entropy*, 24(10), article number 1344. doi: 10.3390/e24101344.

[4] Babenko, V., Myroniuk, T., & Krivous, H. (2021). Algorithms for application of permutation operations controlled by information for implementation of cryptographic transformation of information. *Bulletin of Cherkasy State Technological University*, 26(3), 44-58. doi: 10.24025/2306-4412.3.2021.247252.

[5] Dridi, F., El Assad, S., Wajih, E., & Machhout, M. (2023). Design, hardware implementation on FPGA and performance analysis of three chaos-based stream ciphers. *Fractal and Fractional*, 7(2), article number 197. doi: 10.3390/fractalfract7020197.

[6] Eichelberg, M., Kleber, K., & Kämmerer, M. (2020). Cybersecurity in PACS and medical imaging: An overview. *Journal of Digital Imaging*, 33(6), 1527-1542. doi: 10.1007/s10278-020-00393-3.

[7] El Kinani, K., Amounas, F., Bendaoud, S., & Bayane, Y. (2025). Hybrid approach for IoT-based medical image encryption and compression using modified AES and chaos theory. In Y. Farhaoui, T. Herawan, A.L. Imoize & A.E. Allaoui (Eds.), *Intersection of artificial intelligence, data science, and cutting-edge technologies: From concepts to applications in smart environment. ICAISE 2024. Lecture notes in networks and systems* (Vol. 1353, pp. 390-395). Cham: Springer. doi: 10.1007/978-3-031-88304-0_54.

[8] Ettiyan, R., & Geetha, V. (2023). A hybrid logistic DNA-based encryption system for securing the Internet of Things patient monitoring systems. *Healthcare Analytics*, 3, article number 100149. doi: 10.1016/j.health.2023.100149.

[9] Ihsan, A., & Nurettin, D. (2023). Improved affine encryption algorithm for color images using LFSR and XOR encryption. *Multimedia Tools and Applications*, 82(5), 7621-7637. doi: 10.1007/s11042-022-13727-w.

[10] Liang, Z., Qiuxia, Q., Changjun, Z., Ning, W., Yi, X., & Wenshu, Z. (2021). Medical image encryption algorithm based on a new five-dimensional three-leaf chaotic system and genetic operation. *PLOS One*, 16(11), article number e0260014. doi: 10.1371/journal.pone.0260014.

[11] Liu, Z., Li, C., Zhang, C., & Yang, X. (2025). Dual-domain image encryption scheme based on fractional wavelet transform and hyperchaotic system. *Physica Scripta,* 100(3), article number 035234. doi: 10.1088/1402-4896/adb529.

[12] Luzhetskyi, V.A., & Horbenko, I.S. (2013). Method for forming permutations of an arbitrary number of elements. *Information Security*, 15(3), 262-267.

[13] Oikonomou, P., Kranas, G.K., Sapounaki, M., Spathoulas, G., Aretaki, A., Kakarountas, A., & Adam, M. (2025). Square-based division scheme for image encryption using generalized Fibonacci matrices. *Mathematics*, 13(11), article number 1781. doi: 10.3390/math13111781.

[14] Rukhin, A., *et al*. (2010). *A statistical test suite for random and pseudorandom number generators for cryptographic applications (NIST SP 800-22 Rev. 1)*. doi: 10.6028/NIST.SP.800-22r1a.

[15] Shannon, C.E. (2001). A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1), 3-55. doi: 10.1145/584091.584093.

[16] Stoyanov, B., & Kordov, K. (2015). Image encryption using Chebyshev map and rotation equation. *Entropy*, 17(4), 2117-2139. doi: 10.3390/e17042117.

[17] Sun, Y.-J., Zhang, H., Wang, X.-Y., & Wang, M.-X. (2021). Bit-level color image encryption algorithm based on coarse-grained logistic map and fractional chaos. *Multimedia Tools and Applications*, 80, 12155-12173. doi: 10.1007/s11042-020-10373-y.

[18] Umadevi, Y., Ashwini, M., & Savitha, N. (2022). Fuzzy logic-based parallel data embedding technique for image steganography. *International Journal of Innovative Research in Computer and Communication Engineering*, 10(8). doi: 10.15680/IJIRCCE.2022.1008045.

[19] Wang, N., Wang, X., Liu, A., Wang, W., Ding, Y., Wu, X., & Du, X. (2024). An image partition security-sharing mechanism based on blockchain and chaotic encryption. *PLOS One*, 19(7), article number e0307686. doi: 10.1371/journal.pone.0307686.

[20] Wang, X., Çavuşoğlu, Ü., Kaçar, S., Akgül, A., Pham, V.-T., Jafari, S., Alsaadi, F.E., & Nguyen, X. Q. (2019). S-box based image encryption application using a chaotic system without equilibrium. *Applied Sciences*, 9(4), article number 781. doi: 10.3390/app9040781.

[21] Wu, Y., Noonan, J., & Agaian, S. (2011). NPCR and UACI randomness tests for image encryption. *Journal of Selected Areas in Telecommunications*, 2011, 31-38.

Метод шифрування та розподілу зображень на основі LFSR та лічильників

Володимир Лужецький

Доктор технічних наук, професор
Вінницький національний технічний університет
21021, вул. Хмельницьке шосе, 95, м. Вінниця, Україна
https://orcid.org/0000-0001-7466-7738

Микита Ціхоцький

Асистент
Вінницький національний технічний університет
21021, вул. Хмельницьке шосе, 95, м. Вінниця, Україна
https://orcid.org/0009-0005-8101-3536

Анотація. У сучасних умовах обробки великих обсягів графічних даних постає завдання розробки надійної схеми шифрування зображень зі зменшенням обчислювальних витрат. Метою дослідження було розробити детерміновану схему шифрування та рівномірного розподілу векторизованих зображень із використанням регістра зсуву з лінійним зворотним зв'язком і лічильників. Методи роботи включали перетворення матриці пікселів у послідовність байтів за правилом обходу по рядках, розбиття індексного простору на рівні піддіапазони, генерацію псевдовипадкових індексів на основі станів регістра зсуву та використання реверсивних лічильників. Результати статистичного тестування демонструють стійкі характеристики запропонованого методу шифрування зображень. Також було проведено оцінку зашифрованих тестових зображень до стійкості атаки шляхом визначення коефіцієнтів кореляції між вхідним зображенням та зашифрованим. Зокрема, для кольорових зображень розміром 512×512 при розбитті на вісім піддіапазонів коефіцієнт зміни кількості пікселів склав 99,61 %, а уніфікована середня інтенсивність зміни пікселів – 32,28 %, що відповідає верхньому кластеру оцінок сучасних методів. Ентропія зашифрованих даних наближена до теоретичного максимуму та склала 7,999, а кореляція між сусідніми пікселями істотно зменшена і наближається до нульових значень. Розподіл та відновлення зображення виконується без похибок. Алгоритм відзначається низькими обчислювальними витратами. Практична цінність дослідження полягає в забезпеченні відтворюваності розподілу й високу криптографічну стійкість з використанням математично простих операцій, псевдовипадковості та розширення простору шифрування зображення до повного обсягу, що робить підхід придатним для систем із вимогою точного відновлення й працюють з обмеженими обчислювальними ресурсами

Ключові слова: розподіл секрету; відновлення зображення; перестановка; підстановка; генератор псевдовипадкової послідовності чисел; кореляція пікселів зображення