applied sciences

Article

Enhancing Visual Data Security: A Novel FSM-Based Image Encryption and Decryption Methodology

Gulmira Shakhmetova [1], Alibek Barlybayev [2,3,*], Zhanat Saukhanova [1], Altynbek Sharipbay [2], Sayat Raykul [1] and Altay Khassenov [1]

[1] Department of Information Security, Faculty of Information Technologies, L.N. Gumilyov Eurasian National University, Astana 010008, Kazakhstan; sh_mira2004@mail.ru (G.S.); saukhanova@mail.ru (Z.S.); sayatraykulov@gmail.com (S.R.); khassenov_aye@enu.kz (A.K.)
[2] Department of Artificial Intelligence Technologies, Faculty of Information Technologies, L.N. Gumilyov Eurasian National University, Astana 010008, Kazakhstan; sharalt@mail.ru
[3] Higher School of Information Technology and Engineering, Astana International University, Astana 010008, Kazakhstan
* Correspondence: frank-ab@mail.ru

Abstract: The paper presents a comprehensive exploration of a novel image encryption and decryption methodology, leveraging finite state machines (FSM) for the secure transformation of visual data. The study meticulously evaluates the effectiveness of the proposed encryption algorithm using a diverse image dataset. The encryption algorithm demonstrates high proficiency in obfuscating the original content of images, producing cipher images that resemble noise, thereby substantiating the encryption's effectiveness. The robustness of the proposed methodology is further evidenced by its performance in the National Institute of Standards and Technology Statistical Test Suite (NIST STS). Such achievements highlight the algorithm's capability to maintain the stochastic integrity of encrypted data, a critical aspect of data security and confidentiality. Histogram analysis revealed that the encryption process achieves a uniform distribution of pixel values across the encrypted images, masking any identifiable patterns and enhancing the security level. Correlation analysis corroborated the success of the encryption technique, showing a substantial reduction in the correlation among adjacent pixel values, thereby disrupting spatial relationships essential for deterring unauthorized data analysis. This improvement indicates the algorithm's efficiency in altering pixel patterns to secure image data. Additionally, a comparative analysis of correlation coefficients using various encryption methods on the Lenna image offered insights into the relative effectiveness of different techniques, emphasizing the importance of method selection based on specific security requirements and data characteristics.

Keywords: finite state machines; invertible finite automata; image encryption and decryption; NIST statistical test suite; histogram analysis; correlation analysis

check for updates

Citation: Shakhmetova, G.; Barlybayev, A.; Saukhanova, Z.; Sharipbay, A.; Raykul, S.; Khassenov, A. Enhancing Visual Data Security: A Novel FSM-Based Image Encryption and Decryption Methodology. *Appl. Sci.* **2024**, *14*, 4341. https://doi.org/10.3390/app14114341

Academic Editor: Andrea Prati

Received: 18 April 2024
Revised: 15 May 2024
Accepted: 17 May 2024
Published: 21 May 2024

1. Introduction

Cryptography strongly depends on the processes of encryption and decryption [1], both of which are primarily used to protect data. Encryption techniques are employed to convert plain text into encrypted text, while decryption reverses this process [2] by transforming encrypted text back into plain text. An expert specializing in the manipulation of cryptographic algorithms for the purpose of encryption and decryption is denoted as a "cryptographer." When an individual possesses exclusive covert knowledge, the identification of information becomes feasible. The pivotal element in this context is the concealed information that undergoes secure transmission [3] to the designated recipient. Within the realm of cryptography, the encryption process constitutes the secure transformation of data from one state to another. Consequently, ciphertext, representing the outcome of encryption, remains impervious to unauthorized comprehension. Encryption serves as a protective measure for data, whether stored within a computer system or transmitted across the Internet. The crux of any encryption lies in the encryption key, which is classified into public and private keys [4]. Both keys are integral to the processes of encryption and decryption, with public keys accessible to all, while private keys necessitate strict confidentiality. The strength of encryption is directly proportional to the key size [5], rendering the decryption of data progressively more formidable with larger key sizes. The decryption process involves the conversion of encrypted text into a readable and comprehensible form by employing the requisite codes or keys manually. The absence of the secret key poses a formidable challenge to decoding data, ultimately yielding the original text post-decryption. In the domain of cryptography, automata theory finds varied applications. Deterministic finite automaton (DFA) [6], a branch of the Theory of Computation grounded in Automata, generates a unique encoded string for every given string character input. The terms "deterministic" and "uniqueness of computation" are synonymous in this context. Non-deterministic finite automata, on the other hand, permit zero, one, or multiple transitions from one state to another for a comparable input symbol. If S represents a non-empty collection of k, j, l states, the outcome is an element of S for deterministic automata and a subset of S for non-deterministic automata. Consequently, a finite state machine (FSM) functions as an algorithm characterized by a fixed count of states and transitions. This methodology is presently extensively employed in cryptography for data encoding, ensuring the preservation of data privacy. In cryptography, there are three main groups of methods. These methods represent different approaches to cryptography, utilizing various principles and techniques for securing information and communication. The Vernam Cipher focuses on perfect secrecy through one-time pad usage [7], finite automaton public key cryptosystems involve computational models for key generation [8], and cellular automata may find applications [9] in certain cryptographic processes.

The Vernam Cipher, also known as the one-time pad, is a symmetric key algorithm where the key is as long as the message and is completely random. The key is used only once, making it theoretically unbreakable if used correctly. It operates by combining each character of the plaintext with a corresponding character in the key using modular addition. Within the Vernam cipher algorithm, a key is employed to encrypt character strings, with the key's length aligning precisely with that of the string. Encryption Algorithm Steps: 1. Attribute a numerical value to each position within the string. 2. Perform the summation of the assigned numerical values. 3. In the event that the resultant integer exceeds 26, subtract said integer from 26; otherwise, proceed with the encryption process. Secret key cryptography, or symmetric-key cryptography, involves using the same key for both encryption and decryption. The key must be kept secret between the communicating parties. Algorithms like DES, AES, and Blowfish are examples of secret key cryptography [10], where the security of the system relies on keeping the key secret.

A finite automaton is a computational model consisting of states, transitions, and input symbols. In the context of public key cryptography, finite automata are employed to create unique encoded strings based on input characters. Finite automata serve as the underpinnings for cryptosystems rooted in language theory. The majority of cryptosystems based on linguistic and word-related challenges exhibit vulnerabilities or fail to meet the requisite criteria for digital signature robustness. Automata-centric cryptosystems are categorized into three classes: transducers, cellular automata, and acceptors. Primary finite automata-based cryptosystems encompass FAPKC [11], Gysin [12], and Dömösi [13] cryptosystems. Public key cryptography, or asymmetric cryptography, uses a pair of keys: a public key for encryption and a private key for decryption. The public key can be shared openly, while the private key is kept secret. Common algorithms include RSA and elliptic curve cryptography. The use of finite automata in public key cryptosystems may involve the generation or manipulation of cryptographic keys. Adherence to two fundamental principles, namely secrecy and authenticity, is imperative for any cryptosystem. Symmetric cryptosystems pose a challenge in maintaining confidentiality, given that a single secret key is employed for both encryption and decryption. Consequently, the exchange of this key between communicating parties or reliance on a third entity, such as a key allocation center, becomes necessary for key distribution. However, such dependence on a third entity introduces potential risks to the confidentiality of the secret key. Contrastingly, in public key cryptography, each user is mandated to generate a key pair. One key, termed the private key, remains concealed, while the other, known as the public key, is disclosed openly. The decision of whether to utilize the giver's private key or the receiver's public key for encrypting the original message is contingent upon the specific application at hand.

Cellular automata are discrete, computational models composed of a grid of cells, each in a particular state. Cellular automata exhibit dynamic behavior characterized by unique attributes. The state of each cell evolves over discrete time steps based on a set of rules. This algorithm is constituted by a sequence of cells, which undergo sequential updates with stochastic timing. Cellular automata consist of two essential components: (1) an assembly of cells and (2) a prescribed set of rules. In Principal Component Analysis (PCA), control signals are applied to the structure of cellular automata. The fundamental objective in the design of Lightweight Cellular Automata (LCASE) is the substantial enhancement of requisites for both parties involved [14]. Nevertheless, the model grapples with several shortcomings to address conventional security concerns, and various challenges considered in formulating the proposed algorithm include swift performance coupled with low code density, resilience against attacks such as traditional cryptanalysis and timing assaults, as well as a judicious balance between code efficiency and straightforward implementation. Cellular automata represent a distinctive computational paradigm that provides a facile, extensible, and efficient framework for simulating extensive systems and conducting intricate computations grounded in environmental evidence. Cellular automata have been applied in various fields for modeling and computation. Cryptography involves secure communication through the use of codes and ciphers. While cellular automata are not a standard tool in traditional cryptography, they have been explored for certain cryptographic applications. For instance, cellular automata may be used in the design of stream ciphers or in generating pseudorandom sequences, contributing to cryptographic protocols. The objective of this paper is to develop an innovative encryption system utilizing finite-state machines and their invertible properties.

2. Literature Review

The 2008 book authored by Renji Tao [15] delves into finite automata and their applications in cryptography, offering a comprehensive exposition of the pivotal role played by finite automata within cryptographic systems. The book discusses the different types of finite automata, their applications in cryptography, and the importance of security and efficiency in cryptographic systems. By exploring these concepts and techniques, the paper offers valuable insights into the use of finite automata in the development and implementation of secure and efficient cryptographic systems.

In the scientific paper [16] by Mitchell, Shmatikov, and Stern, the authors present a detailed analysis of the Secure Sockets Layer (SSL) 3.0 protocol. SSL is a widely used cryptographic protocol that ensures secure communication between a client and a server over the Internet. The authors aim to analyze the potential vulnerabilities and attacks on SSL 3.0 by employing finite-state machine techniques. Their findings underscore the need for continued research and development in the field of cryptography to ensure the security and privacy of Internet communications.

In the field of cryptography, ensuring the secure exchange of cryptographic keys is a vital aspect of preventing unauthorized access. In this scientific paper [17], Isa, Ahmad, Sani, Hashim, and Mahmod propose a new cryptographic key exchange protocol that uses message authentication codes (MAC) and finite state machines. The protocol offers strong security guarantees and improved efficiency compared to other existing key exchange protocols. The authors use MAC to ensure the integrity and authenticity of the exchanged messages, while finite state machines are employed to manage the state transitions during the key exchange process.

The scientific paper by Kearns and Valiant [18] explores the relationship between cryptography and learning in the context of boolean formulae and finite automata. This paper investigates the limitations of learning algorithms in the presence of cryptographic techniques and discusses how these limitations can be applied to improve the security of learning systems. The authors begin by providing an overview of the field of cryptography and its applications, with a focus on the role of cryptographic primitives in securing communication and protecting sensitive information. They then introduce the concept of learning algorithms, which are used to identify patterns and structure within datasets and discuss the potential vulnerabilities that these algorithms may exhibit when faced with cryptographic techniques.

In recent years, the application of finite automata in cryptography has gained significant attention due to its ability to provide enhanced security and efficiency in various cryptographic algorithms. In this scientific paper [19] by Sharipbay, Saukhanova, and Shakhmetova, the authors explore the potential of finite automata in the design and implementation of cryptographic systems. By exploring various cryptographic applications of finite automata, the authors demonstrate that this mathematical model can be a valuable tool in the development of secure and efficient cryptographic algorithms.

The scientific paper [20] by Amorim, Machiavelo, and Reis investigates the use of linear finite automata in cryptographic systems. In this paper, the authors present a detailed study of the properties and characteristics of linear finite automata and their potential applications in cryptography. By examining the properties and characteristics of LFAs and analyzing their use in various cryptographic protocols, the authors demonstrate the potential of LFAs as a powerful tool in the field of cryptography.

Automata theory offers a powerful mathematical framework for the design and analysis of encryption and decryption algorithms. The cryptosystem presented in this paper [21] demonstrates the practical application of Automata theory in the field of cryptography. This work contributes to the ongoing research in cryptography, providing a new approach to secure data transmission and communication.

The scientific paper [22] by Vayadande, Agarwal, Kabra, Gangwal, and Kinage demonstrates the potential of Automata Theory in enhancing the security and efficiency of cryptographic algorithms. By combining the power of Automata Theory and cryptography, researchers can develop more secure and robust communication systems that can withstand the ever-evolving threats from malicious actors.

In the scientific paper [23], Khaleel, Turaev, Al-Shaikhli, and Tamrin provide a comprehensive analysis of cryptosystems that are based on finite automata. They discuss various types of cryptosystems, their classification, and their applications in the field of cryptography. The paper also highlights the challenges faced by cryptosystems based on finite automata and suggests some potential future directions for research and development. These include improving the security and efficiency of these cryptosystems, as well as exploring new applications and integrating them with other cryptographic techniques. In conclusion, the scientific paper provides a comprehensive overview of cryptosystems based on finite automata, their classification, and their various applications in the field of cryptography.

In this paper [24] paper by Peña and Torres present a novel authenticated encryption scheme based on the finite automata cryptosystem. The authors demonstrate that their method offers competitive performance and security properties, making it a promising candidate for real-world applications in the field of cryptography.

In recent years, the field of nonlinear dynamics has seen significant advancements, particularly in the study of deterministic chaotic systems. This paper [25], by Alawida, Samsudin, Teh, and Alshoura, explores the concept of deterministic chaos in the context of finite-state automata, a fundamental concept in computer science and automata theory. The authors present a novel approach to modeling and analyzing deterministic chaos in finite-state automata, which has potential applications in various fields such as secure communication, cryptography, and control systems. The paper concludes by discussing the potential applications of deterministic chaotic finite-state automata in various fields, such as secure communication, cryptography, and control systems. The authors emphasize the need for further research to explore the potential of these systems and their practical implementation.

In recent years, the use of unmanned aerial vehicles (UAVs) has become increasingly prevalent in various industries, including military, agriculture, and environmental monitoring. The exchange of data between UAVs and ground stations is crucial for the efficient functioning of these systems. However, ensuring the security of these data is of utmost importance, as unauthorized access can lead to significant consequences. To address this concern, a new image encryption algorithm based on a DNA state machine has been proposed for UAV data encryption. In this scientific paper [26] by Alawida, Teh, and Alshoura, the authors present a novel image encryption algorithm that utilizes the principles of DNA state machines to provide robust security for UAV data transmission. The primary objective of this algorithm is to enhance the security of UAV data while maintaining the efficiency of the overall communication process. In conclusion, the new image encryption algorithm based on DNA state machines presents a promising solution for securing UAV data transmission. The algorithm offers high levels of security, efficiency, and visual quality, making it an attractive option for use in various UAV-based applications. The authors of the paper have demonstrated the effectiveness of their algorithm through a comprehensive evaluation and comparison with other encryption techniques.

The paper [27] by Geng, Wu, Wang, Zhang, and Wang presents an image encryption algorithm based on block scrambling and finite state machines. This paper aims to provide a comprehensive analysis of the algorithm, its advantages, and potential applications. The combination of block scrambling and FSM ensures a high level of security while maintaining a reasonable computational cost. The algorithm's performance was evaluated using various experiments and comparisons with other existing encryption algorithms, demonstrating its superiority in terms of security and efficiency. As a result, this algorithm holds great potential for practical applications in fields such as digital forensics, medical imaging, and secure communication.

In recent years, the field of cryptography has seen significant advancements, particularly in the area of image encryption. This scientific paper [28] by Dougherty, Klobusicky, Şahinkaya, and Ustun introduces an S-Box construction based on exponentiation in finite fields. The proposed method offers a balance between security and visual quality while maintaining a competitive encryption speed. The use of exponentiation in finite fields for S-Box construction provides increased security and reduced complexity, making it a promising approach for future research in the field of cryptography and digital image encryption. In papers [26–28] where images were encrypted and decrypted, the following methods were used: NIST STS Analysis, Histogram Analysis, and Correlation Analysis.

3. Materials and Methods

3.1. General Finite State Machines

Finite state machines undergo state transitions and generate output sequences based on input symbols [29]. The behavior of a finite state machine is characterized by a set of components, including the set of states, set of input symbols, set of output symbols, start state, and final state [30]. There exist two primary types of finite state machines: Moore Machines [31] and Mealy Machines [32]. These machines are collectively referred to as finite-state machines due to the finite nature of their states. Renowned for their application in cryptography, these machines generate output sequences corresponding to each input sequence. Cryptography applications often leverage cellular automata, transducers (both Moore and Mealy Machines), and state machines with accepting states, excluding output symbols. In essence, a finite state machine is defined as a system that yields an output sequence for each input sequence across distinct states. Moore and Mealy machines are commonly depicted using graphical notations suitable for digital circuit representation [33].

Finite state machines, encompassing both Moore and Mealy variants, are denoted as M and defined by six tuples, i.e., $M = (X, Y, S, \delta, \lambda, x_0)$. Here, the tuples X, Y, S, and x_0 signify the finite sets of states, input symbols, output symbols, and the initial state. The tuples δ and λ represent one-way mapping functions. Specifically, δ is defined as $\delta : X \times Y \to X$, illustrating the transition function for both Moore and Mealy machines. Simultaneously, λ is articulated as $\lambda : X \to S$ for Moore machines and $\lambda : X \times Y \to S$ for Mealy machines, delineating the output mapping functions associated with each respective machine type.

Since the finite automaton acts as an encoder/decoder, it must have invertibility properties. In our case, a weakly invertible state machine with delay is used. A finite state machine is called invertible if, knowing the output sequence and the initial state, the input sequence can be uniquely determined.

3.2. Encryption Algorithm Using Finite State Machines

In the initial phase of the encryption process, the file undergoes processing by a permutation machine, followed by encryption utilizing a weakly invertible state machine. Specifically, a permutation state machine, serving as an automaton, orchestrates the rearrangement of characters within the input string based on predefined permutation rules. As the permutation state machine is integral to the decryption procedure, its invertible is imperative. To achieve this, we employed an invertible finite automaton possessing the right invertibility. The requisite condition for right invertibility mandates that the automaton mapping X_a be surjective for any initial state 'a'. In other words, it is both necessary and sufficient for each row of the output table to encompass occurrences of every output signal at least once [34].

The program incorporates a module dedicated to the generation of finite-state machines. It is emphasized that the quantity of finite automata within this framework, characterized by n states, x input symbols, and y output symbols, follows a combinatorial expression of $(n \times y)^{n \times x}$ [35]. It is acknowledged that not all automata within this classification exhibit invertibility. Consequently, an algorithm designed to verify invertibility is employed to ascertain the membership of a given finite automaton within the category of invertible finite automata.

Subsequently, the produced automata undergo an assessment for invertibility utilizing the Kohavi algorithm [36,37]. The invertibility testing module facilitates the construction of a repository comprising invertible finite state machines. Subsequently, these machines are deployed for the encryption of the file.

Per Theorem 1 presented in Article [35] and Theorem 1.4.1 elucidated in Book [15], a finite automaton attains invertibility if and only if its associated graph is devoid of cycles. The algorithm employed for verifying invertibility encompasses the following sequential steps [36]:

1. Consider a finite automaton denoted as $M =< X, Y, S, \delta, \lambda >$, represented in the format of a state table. In this context, both X and Y are elements of the set {0, 1};
2. A testing table (TTab) is constructed, consisting of 2 parts;
   - Upper part of TTab: The state table of the finite state machine is rewritten; the rows in the upper part of the table correspond to the states of the machine, and the columns to the output symbols. The entry at the intersection of row s_i and column y_m are states that can be reached from state s_i using a single transition associated with the output symbol y_m. The entire upper half of the table is actually a list of the next states of the machine and is therefore called the output table;
   - Bottom of the TTab: Each compatible state pair appearing at the top becomes the row header of the bottom of the testing table. The next states of these pairs are: they consist of all implied joint pairs. Any implied pair of states that have not already been used as a row header becomes the row header, its next state being found in the same way. The process terminates when all compatible state pairs have been used as row headers;

3. A testing-oriented weighted graph G, according to TTab, is constructed using the steps as follows:
    - Each compatible pair of states $s_i s_j$ corresponds to a vertex in G;
    - An arc labeled y_m goes from vertex $s_i s_j$ to vertex $s_p s_q$, where $p \neq q$, if and only if the compatible pair $(s_p s_q)$ is the next state of the compatible pair $(s_i s_j)$;
4. Determine the ratio of the original finite state machine to the group of invertible finite state machines with memory and find its delay.
    - The graph G from Step 3 is checked for the presence of a cycle. If there is a cycle, then the finite automaton is not an invertible automaton with finite memory; if there is no cycle, the finite automaton is an invertible automaton with finite memory;
    - Find the delay $\mu = l + 1$, where l is the longest path of the graph G from step 3.

For example:
The finite state machine is given in the form of Table 1.

Table 1. State machine for checking for invertibility.

PS	NS, z	
	x = 0	x = 1
0	0,0	1,0
1	2,0	3,0
2	3,1	2,1
3	1,1	0,1

We build testing Table 2 for the proposed finite state machine.

Table 2. Test table for checking the finite state machine for invertibility.

S	z = 0	z = 1
0	(0,1)	-
1	(2,3)	-
2	-	(2,3)
3	-	(0,1)
(0,1)	(0,2), (0,3) (1,2), (1,3)	-
(2,3)	-	(0,2), (0,3) (1,2), (1,3)

According to testing Table 2, a testing graph is constructed, which is shown in Figure 1.

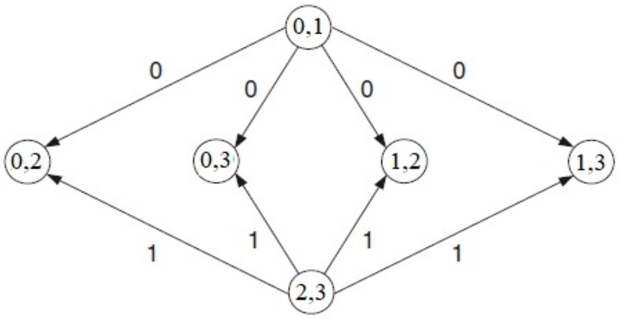


Figure 1. Testing graph.

Based on testing Table 2, a graph adjacency matrix is constructed. The graph adjacency matrix is shown in Figure 2. To check the testing graph for the presence of loops or cycles, there is the following algorithm:

1. Based on the testing table, an adjacency matrix of the graph G with dimension p × p is constructed (p is the number of vertices of the graph);
2. The row that has 0 in all positions is deleted, and the corresponding column is deleted. If there are no such lines, go to step 4;
3. Step 2 is repeated;
4. If the matrix has not completely disappeared, then the graph has a cycle. If the matrix has disappeared, then the graph does not have a cycle.

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{1} \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{1} \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{1} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \xrightarrow{2} \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \xrightarrow{2} [0]$$

Figure 2. Graph adjacency matrix.

According to Step 1, an adjacency matrix is constructed, and then in Step 2, zero rows and their corresponding columns are removed. As a result, our matrix disappears. Therefore, the graph has no cycles. It is worth noting that the number of iterations required to remove all rows of zeros represents the finite delay τ.

After checking, we can say that the considered machine is invertible with a delay of 2.

In addition, it should be noted that the upper estimate of the delay μ, which was determined by R.Tao, where $\mu \leq \frac{|S|(|S|-1)}{2}$. Indeed, the automaton considered in the example has four states, then we obtain $\frac{|4|(|4|-1)}{2} = 6 \geq \mu$. This statement was investigated and demonstrated in [15]; R. Tao tested the possibility of constructing invertible automata with a delay μ. When choosing a delay μ for more states of the automaton, the automaton was not invertible.

At the decryption stage, the encrypted file is submitted for transformation to a finite state machine, which is the inverse of the machine used in the encryption process. The decryption process using invertible FSMs involves several key steps:

1. The decryption begins with the FSM in a known initial state. The encrypted data (output sequence of the encryption process) is then fed into the FSM;
2. As the FSM processes the input data, it transitions between states according to its transition function, which is designed to be reversible. The transition functions and the output functions are crafted such that each input and output symbol leads to a unique next state;
3. Since the FSM is invertible, the sequence of states it transitions through can be used to uniquely determine the original input sequence. This sequence represents the decrypted data.

The invertible finite state machine was built using the finite state machine algorithm described by Olson [38]. After the intermediate file is received, it is fed to the inverse permutation finite state machine, which is built according to the rule described in the article [34].

3.3. Statistical Test Suite

The National Institute of Standards and Technology (NIST) Statistical Test Suite (STS) comprises an assemblage of open-source cryptographic software and algorithms [39]. Encompassing both encryption and digital signature schemes, the STS serves diverse purposes, including secure communication and data protection. Engineered for broad applicability, it is well-suited for deployment in both commercial and research contexts. The STS provides not only cryptographic tools but also incorporates test materials and benchmark results, enabling a comprehensive evaluation of the performance and security attributes of cryptographic systems. Consequently, the STS stands as an invaluable resource for developers, researchers, and organizations engaged in the formulation and implementation of cryptographic systems.

The NIST STS includes a variety of tests designed to assess the randomness and statistical properties of cryptographic algorithms and random number generators. Some of the key tests commonly employed in the NIST STS include the following:

1. Frequency (Monobit) Test—assesses whether the number of ones and zeros in a sequence are approximately equal;
2. Block Frequency Test—examines the frequency of fixed-size blocks in the sequence;
3. Runs Test—analyzes the number of runs of ones and zeros in the sequence;
4. Longest Run of Ones in a Block Test—investigates the length of the longest run of ones in a sequence;
5. Binary Matrix Rank Test—assesses the rank of disjoint submatrices of the entire sequence;
6. Discrete Fourier Transform (Spectral) Test—utilizes the discrete Fourier transform to analyze the frequency content of the sequence;
7. Non-overlapping Template Matching Test—checks for the presence of specific fixed-size patterns in the sequence;
8. Overlapping Template Matching Test—similar to the non-overlapping template matching test but allows for overlapping patterns;
9. Maurer's Universal Statistical Test—measures the compressibility of a sequence;
10. Random Excursions Test—analyzes the number of cycles having specified numbers of visits in a random walk;
11. Random Excursions Variant Test—examines a variant of the random excursions test.

These tests collectively provide a comprehensive evaluation of the randomness and statistical properties of sequences, helping to ensure the robustness of cryptographic systems. The NIST STS is widely used in cryptographic research and standardization efforts.

3.4. Histogram Analysis

Histogram analysis is a fundamental technique in image processing used to understand the distribution of pixel intensities within an image [40]. A histogram is a graphical representation that shows the frequency of pixel intensity values in an image, typically ranging from 0 (black) to 255 (white) in grayscale images. In the context of color images, separate histograms are generated for each color channel (e.g., red, green, and blue). The histogram provides insights into the overall distribution of pixel intensities in an image. Peaks and valleys in the histogram reveal the predominant intensity levels, helping to understand the contrast and brightness of the image. Histogram analysis is often used to enhance the contrast of an image. By stretching or equalizing the histogram, the full range of intensity values can be utilized, resulting in a visually more appealing image. Histogram analysis aids in adjusting the brightness of an image. Shifting the histogram to the left or right can make an image darker or brighter, respectively. Histogram analysis is a versatile tool that forms the basis for many image-processing algorithms. Understanding and manipulating the pixel intensity distribution through histogram analysis can significantly improve the visual quality and interpretability of images.

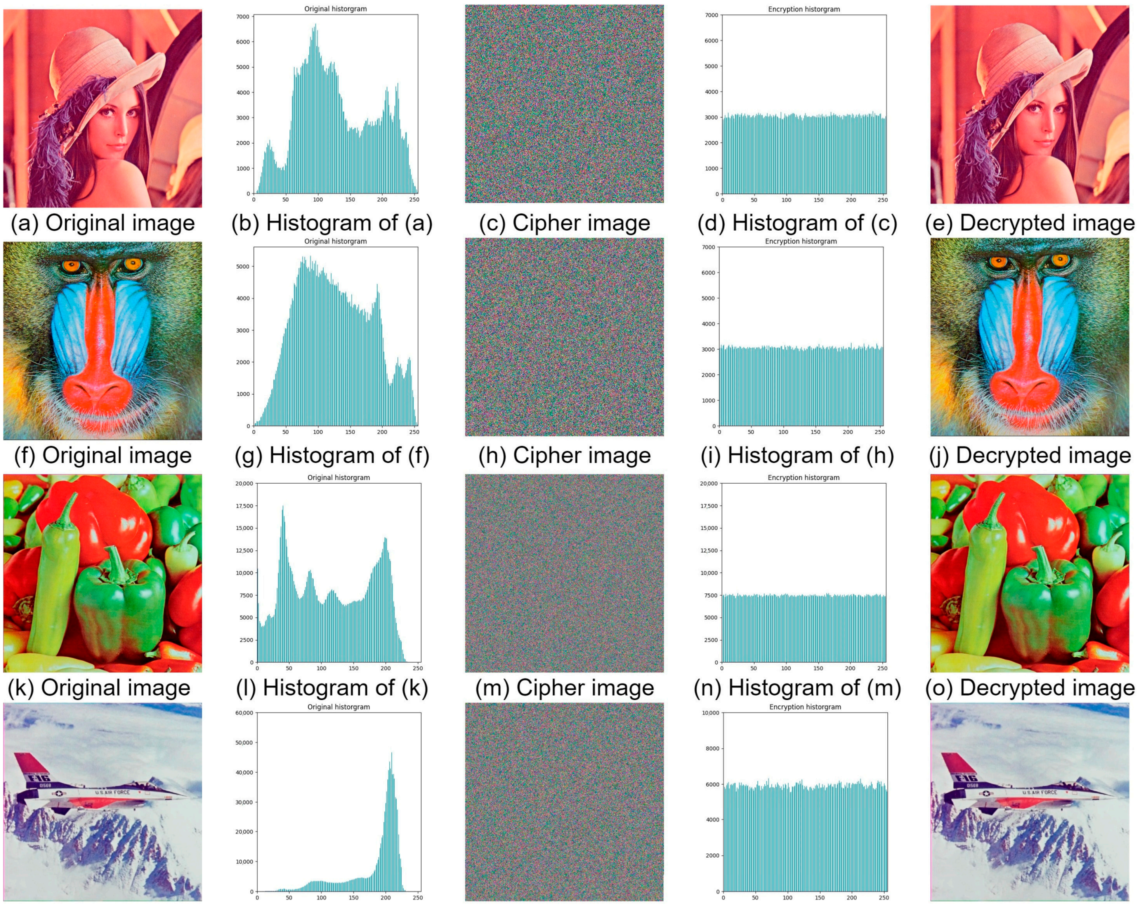
3.5. Experimental Data

To evaluate the efficacy of file encryption, image files featuring Lenna, Mandrill, Peppers, and Airplane were employed, each possessing dimensions of 512 pixels by 512 pixels.

4. Results and Discussion

4.1. Image Encryption and Decryption

The experimental procedure for image encryption was conducted using four distinct images: Lenna, Mandrill, Peppers, and Airplane. Each of these images possesses a resolution of 512 × 512 pixels and was sourced from the University of Southern California's Signal and Image Processing Institute's database, accessible at https://sipi.usc.edu/database/ (accessed on 20 May 2024). This selection of images encompasses a diverse range of visual content, facilitating a comprehensive assessment of the encryption methodology's efficacy across various image characteristics. Figure 3 illustrates the proficiency of the proposed algorithm in producing cipher images that resemble noise, thereby demonstrating the algorithm's encryption effectiveness.



(a) Original image (b) Histogram of (a) (c) Cipher image (d) Histogram of (c) (e) Decrypted image

(f) Original image (g) Histogram of (f) (h) Cipher image (i) Histogram of (h) (j) Decrypted image

(k) Original image (l) Histogram of (k) (m) Cipher image (n) Histogram of (m) (o) Decrypted image

(p) Original image (q) Histogram of (p) (r) Cipher image (s) Histogram of (r) (t) Decrypted image

Figure 3. The dataset comprises three distinct categories of images: the original images, the encrypted (cipher) images, and the decrypted images.

The design of the encryption process is predicated on achieving both confusion and diffusion within a single iteration of each round. This is accomplished by ensuring that a minimal alteration (a single bit change) to the secret key results in the generation of distinct round keys, which in turn precipitates a comprehensive diffusion of the effect across the entirety of the encrypted image. Consequently, an inspection of the cipher images yields no discernible or valid information. As depicted in Figure 3, the cipher images are devoid of any meaningful content, underscoring the effectiveness of the encryption. Furthermore, the images are successfully restored to their original state through the decryption process, validating the invertible nature of the proposed encryption technique.

4.2. NIST STS Analysis

The stochastic nature of the proposed methodology is substantiated through rigorous validation using the NIST STS, applied to the Lena image dataset. The outcomes of these assessments demonstrate a high degree of success and are systematically elucidated in Table 3.

Table 3. Results of the NIST statistical test suite for the proposed Method.

Test Name		p-Value	Conclusion
2.01. Frequency Test:		0.011603	True
2.02. Block Frequency Test:		0.898022	True
2.03. Run Test:		0.194845	True
2.04. Run Test (Longest Run of Ones):		0.366963	True
2.05. Binary Matrix Rank Test:		0.062628	True
2.06. Discrete Fourier Transform (Spectral) Test:		0.575633	True
2.07. Non-overlapping Template Matching Test:		0.985412	True
2.08. Overlapping Template Matching Test:		0.28564	True
2.09. Universal Statistical Test:		0.874041	True
2.10. Linear Complexity Test:		0.809612	True
2.11. Serial Test:		0.159966	True
		0.422105	True
2.12. Approximate Entropy Test:		0.224807	True
2.13. Cumulative Sums (Forward):		0.021299	True
2.13. Cumulative Sums (Backward):		0.021299	True
2.14. Random Excursion Test:			
‘−4’	3.200982	0.669032	True
‘−3’	6.112628	0.295414	True
‘−2’	3.762932	0.584027	True
‘−1’	10.89873	0.053425	True
‘+1’	7.531646	0.184007	True
‘+2’	5.497578	0.358212	True
‘+3’	5.469185	0.361337	True
‘+4’	4.987242	0.417439	True
2.15. Random Excursion Variant Test:			
‘−9.0’	123	0.39589	True
‘−8.0’	119	0.411277	True
‘−7.0’	101	0.627375	True
‘−6.0’	66	0.755169	True
‘−5.0’	50	0.44187	True
‘−4.0’	54	0.452213	True
‘−3.0’	57	0.433789	True
‘−2.0’	64	0.49084	True
‘−1.0’	85	0.633124	True
‘+1.0’	55	0.056219	True
‘+2.0’	54	0.25085	True
‘+3.0’	48	0.270057	True
‘+4.0’	48	0.351261	True
‘+5.0’	53	0.490519	True
‘+6.0’	58	0.614454	True
‘+7.0’	59	0.658999	True
‘+8.0’	40	0.42307	True
‘+9.0’	30	0.344424	True

The comprehensive evaluation of the proposed methodology’s stochastic characteristics, validated through the NIST statistical test suite applied to the Lena image dataset, reveals a consistently high level of success across a spectrum of statistical tests. The results, as summarized in Table 3, demonstrate that the methodology satisfies the stringent criteria set forth by each test within the suite. Specifically, each test, ranging from frequency and block frequency tests to discrete Fourier transform and cumulative sums analyses, yields p-values that surpass the commonly accepted significance level of 0.05. This signifies a robust adherence to randomness in the generated data, as indicated by the “True” conclusions associated with each test. Notably, the Random Excursion and Random Excursion Variant Tests, which assess the excursion behavior of the generated sequences, also exhibit conformity with expected randomness patterns. The p-values associated with various excursions, both positive and negative, consistently exceed the significance threshold, affirming the statistical integrity of the proposed methodology.

4.3. Histogram Analysis

The histogram serves as a graphical representation that delineates the pixel value distribution within an image, which fundamentally exists as a two-dimensional matrix populated with pixel values ranging from 0 to 255. This graphical tool maps the frequency of occurrence of each pixel value, offering a total of 256 possible pixel intensities. In the context of cipher images, an ideal histogram exhibits a uniform distribution, indicating that each pixel value appears with equivalent probability, thereby obfuscating any discernible patterns within the encrypted image. This study encapsulates the encryption of four distinct plain images, followed by the generation and analysis of their histograms both pre- and post-encryption, as depicted in Figure 3. The analysis reveals that the histograms of the encrypted (cipher) images manifest as uniformly distributed (flat histograms), signifying the absence of identifiable biases or statistical anomalies. Such uniformity in pixel distribution across the histogram underscores the efficacy of the encryption process, as it intuitively demonstrates the achievement of uniform pixel distribution within the encrypted images, thus ensuring the concealment of any potential patterns.

4.4. Correlation Analysis

The correlation coefficient is a critical metric for analyzing the relationship between adjacent pixel values within an image. In unencrypted (plain) images, adjacent pixel values typically exhibit a strong correlation, a characteristic that is markedly diminished in encrypted (cipher) images. The calculation of correlation coefficients encompasses three principal orientations: vertical, horizontal, and diagonal. According to the methodology deployed in the proposed encryption algorithm, it is possible to significantly disrupt the correlation between adjacent pixels within merely two rounds of encryption, thereby enhancing security. For the scope of this investigation, correlation coefficients for all test images were computed across the aforementioned orientations, with the results detailed in Figure 4. This quantitative analysis demonstrates that the proposed encryption technique effectively disperses both the pixel values of the plain images and the cryptographic key values throughout the encrypted images. Such dispersion results in a negligible correlation among adjacent pixel values. Conversely, the pixels in plain images display a pronounced correlation with one another, a relationship that is clearly visualized in both Figure 4 and Table 4. This contrast underscores the effectiveness of the encryption process in obfuscating the inherent patterns and relationships within the original image data.

Analysis of Table 4 reveals that subsequent to the encryption process, the encrypted (cipher) images exhibit a significant reduction in the correlation among adjacent pixels. Moreover, Figure 4 illustrates a marked contrast in the distribution patterns of adjacent pixels between the original and encrypted images. Specifically, the distribution of adjacent pixels in the original images is observed to be highly concentrated, signifying a strong correlation among these pixel values. This concentration indicates that pixel values in unencrypted images tend to be similar to their immediate neighbors, reflecting inherent patterns and textures within the visual content. Conversely, the distribution of adjacent pixels within the cipher images is characterized by a random dispersion, which suggests a substantial diminution in correlation. This randomness in the pixel distribution of cipher images signifies an effective disruption of the original image's spatial relationships and patterns, as achieved by the encryption algorithm. The resultant low correlation among adjacent pixels in the encrypted images is indicative of the encryption process's efficacy in obfuscating the original content, thereby ensuring the confidentiality and integrity of the encrypted visual data.

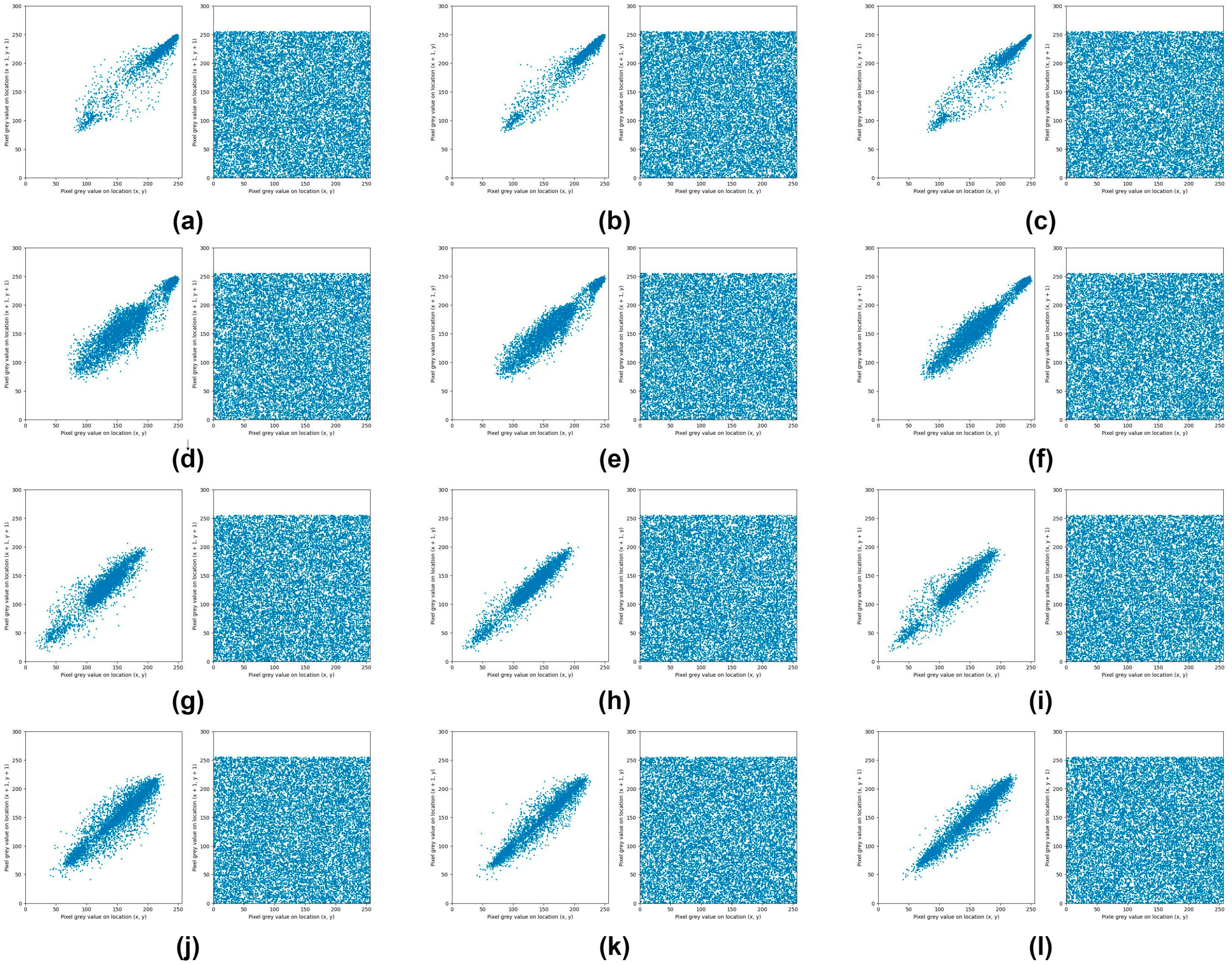


Figure 4. Distribution of adjacent pixels in the original images and corresponding ciphered images: (**a**) Horizontal plain image and cipher image of Lenna, (**b**) vertical plain image and cipher image of Lenna, (**c**) diagonal plain image and cipher image of Lenna, (**d**) horizontal plain image and cipher image of Mandrill, (**e**) vertical plain image and cipher image of Mandrill, (**f**) diagonal plain image and cipher image of Mandrill, (**g**) horizontal plain image and cipher image of Peppers, (**h**) vertical plain image and cipher image of Peppers, (**i**) diagonal plain image and cipher image of Peppers, (**j**) horizontal plain image and cipher image of Airplane, (**k**) vertical plain image and cipher image of Airplane, (**l**) diagonal plain image and cipher image of Airplane.

Table 4. Correlation coefficient of encrypted images.

Image	Horizontal	Vertical	Diagonal
Original Lenna	0.9740	0.9694	0.9507
Encrypt Lenna	−0.0066	−0.0106	−0.0014
Original Mandrill	0.9587	0.9723	0.9385
Encrypt Mandrill	0.0034	−0.0016	0.0027
Original Peppers	0.9535	0.9217	0.9106
Encrypt Peppers	0.0019	−0.0037	−0.0144
Original Airplane	0.9578	0.9622	0.9180
Encrypt Airplane	−0.0127	−0.0052	−0.0134

Table 4 presents correlation coefficients for encrypted images compared to their original counterparts across three orientations: vertical, horizontal, and diagonal. This quantitative analysis involves four distinct images: Lenna, Mandrill, Peppers, and Airplane. The correlation coefficients for the original images are predominantly high, signifying strong similarity across the measured orientations. Specifically, these coefficients range from 0.9106% to 0.9740%, indicating a high degree of correlation in the spatial distribution of pixel intensities in these orientations. Conversely, the encrypted versions of these images exhibit markedly negative or low positive correlation coefficients, ranging from −0.0144 to 0.0034. These values indicate a significant reduction in similarity or a complete dissimilarity in the spatial distribution of pixel intensities post-encryption across all orientations. This dramatic shift in correlation coefficients from the original to the encrypted images underscores the effectiveness of the encryption process in dispersing the pixel intensity patterns, thereby obfuscating the visual and structural content of the images. The analysis of such correlation coefficients is crucial in assessing the strength and effectiveness of image encryption techniques. High correlation coefficients in original images reflect natural patterns and similarities, while significantly lower or negative coefficients in encrypted images indicate successful disruption of these patterns, an essential feature for securing image data against unauthorized access or interpretation.

4.5. Comparison of Results

Table 5 presents a comparative analysis of the correlation coefficients for the encrypted Lenna image using various encryption methods. The correlation coefficients are measured across three orientations: horizontal, vertical, and diagonal. These coefficients, which range from −1 to 1, quantify the degree of linear relationship between pixel values in specified directions. Values close to 0 indicate a lack of linear correlation, reflecting effective encryption in disrupting the predictable patterns of the original image. The encryption methods are diverse, encompassing approaches based on finite state machines, elliptic curve cryptography (ECC), chaotic and hyperchaotic maps, generalized Arnold maps, spatiotemporal chaos processes, particle swarm optimization algorithms, and spatial bit-level permutations.

Table 5. Correlation coefficient comparison of the encrypted Lenna image.

Encrypt Lenna Image	Horizontal	Vertical	Diagonal
Finite State Machine based proposed method	−0.0066	−0.0106	−0.0014
Elliptic curve cryptography-based method using primitive polynomial [41]	−0.0022	−0.0123	−0.0067
Chaotic and hyperchaotic-based methods using enhanced S-box pixel permutator [42]	0.0484	0.0119	0.0141
Hyper-chaos-based method [43]	0.0059	0.0105	0.0142
Generalized Arnold map-based method [44]	0.0173	−0.0027	−0.0177
Hyper-chaotic system with only one round diffusion process-based method [40]	0.0117	−0.0369	−0.0422
Spatiotemporal chaos process-based method [45]	0.0143	0.0280	0.4466
Particle swarm optimization algorithm and cellular automata-based method [46]	0.0053	−0.0089	0.0126
Spatial bit-level permutation and high-dimension-based method [47]	−0.0574	−0.0035	0.0578

The correlation coefficients for these methods exhibit a wide range of effectiveness in obscuring linear relationships within the encrypted image. The Finite State Machine-based proposed method shows coefficients close to zero (−0.0066, −0.0106, −0.0014), indicating a high degree of encryption effectiveness. In contrast, the spatiotemporal chaos process-based method shows a significantly high correlation coefficient in the diagonal orientation (0.4466), suggesting a lesser degree of effectiveness in disrupting patterns in this particular direction. The ECC-based method and the methods utilizing chaotic maps, hyper-chaos, and generalized Arnold maps generally demonstrate low correlation coefficients, implying effective encryption. However, the spatial bit-level permutation method exhibits a notably high negative correlation coefficient in the horizontal orientation (−0.0574) and a high positive coefficient in the diagonal orientation (0.0578), indicating varying degrees of pattern disruption.

The proposed method employs a finite state machine architecture, which is not typically utilized in the context of image encryption. This approach allows for dynamic adjustment of encryption behaviors based on the state transitions, providing unique adaptability in the encryption process that can enhance security against attacks that exploit static characteristics of encryption algorithms. The proposed FSM-based method achieves a horizontal correlation coefficient of -0.0066, which is significantly lower compared to methods such as the Chaotic and hyperchaotic-based method (0.0484) and the Spatiotemporal chaos process-based method (0.0143). This suggests that the FSM-based method is particularly effective in disrupting horizontal pixel patterns and enhancing security against horizontal analysis techniques. The FSM method's diagonal correlation coefficient of -0.0014 is notably lower than many other methods, such as the Hyper-chaos-based method (0.0142), the Particle swarm optimization algorithm, and the cellular automata-based method (0.0126). Lower diagonal correlation is indicative of robust encryption that effectively masks the diagonal patterns, which are critical for maintaining image structure. Compared to advanced methods like those using hyper-chaotic systems or spatial bit-level permutation, the FSM-based method not only holds its own but often surpasses these in terms of reducing correlation, demonstrating its robustness and reliability.

5. Conclusions

The comprehensive experimental analysis and subsequent discussions provide a detailed evaluation of a novel image encryption and decryption methodology. This study showcases the efficacy of the proposed FSM encryption algorithm through the utilization of four benchmark images, demonstrating the algorithm's capacity to generate cipher images that significantly obscure the original content, rendering them akin to noise. The robustness of the encryption process is further affirmed by the seamless restoration of images to their original state post-decryption, highlighting the invertible nature of the encryption technique. The effectiveness of the proposed methodology is underscored by its performance in the NIST STS, where it consistently met the stringent criteria for randomness across an array of statistical tests. This achievement attests to the methodology's capability to ensure the stochastic integrity of encrypted data, which is paramount in maintaining data security and confidentiality. Moreover, the histogram analysis revealed that the encryption process yields a uniform distribution of pixel values, effectively concealing any discernible patterns and thereby enhancing the security of the encrypted images. The correlation analysis further validated the encryption technique's success, demonstrating a significant reduction in the correlation among adjacent pixel values across encrypted images. This disruption of spatial relationships is crucial for thwarting attempts at unauthorized data retrieval or interpretation. This uniform distribution is indicative of the algorithm's capability to significantly alter pixel patterns, ensuring a higher level of image data security. The comparison of correlation coefficients among various encryption methods for the Lenna image provided insights into the relative effectiveness of different encryption techniques. The range of effectiveness in disrupting linear relationships within the encrypted image highlighted the importance of selecting appropriate encryption methods based on specific security requirements and the characteristics of the data to be protected. This study contributes valuable empirical evidence to the field of image encryption, presenting a method that effectively combines confusion and diffusion principles to secure image data. The demonstrated high levels of encryption effectiveness, validated through rigorous statistical testing, affirm the proposed FSM methodology's potential for widespread application in safeguarding digital imagery against unauthorized access and exploitation. A novel image encryption and decryption methodology reveals multiple avenues for practical application in various domains requiring robust data security measures. The Finite State Machine encryption algorithm, central to this research, has demonstrated its ability to generate cipher images that significantly obscure original image content, effectively rendering them unrecognizable and akin to noise. This capability is vital for applications where the confidentiality and security of visual data are paramount, such as in medical imaging, military communications, and proprietary industrial designs. One of the primary applications of this research could be in the field of digital rights management, where protecting visual content from unauthorized use is crucial. The robustness of the FSM algorithm, paired with its ability to seamlessly restore images to their original state post-decryption, ensures that original digital artwork, photographs, and other copyrighted materials can be securely distributed and viewed by authorized users without risking infringement.

Author Contributions: Conceptualization, G.S., A.B., Z.S. and A.S.; methodology, G.S. and Z.S.; software, S.R. and A.K.; validation, A.B., S.R. and A.K.; formal analysis, G.S. and A.B.; investigation, G.S., A.B., Z.S. and A.S.; resources, A.B., S.R. and A.K.; writing—original draft preparation, G.S., A.B., Z.S. and A.S.; writing—review and editing, G.S., A.B., Z.S. and A.S.; supervision, G.S. and A.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan under Grant No. AP19677422.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study, in the collection, analyses, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

References

1. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
2. Lai, J.; Deng, R.H.; Guan, C.; Weng, J. Attribute-based encryption with verifiable outsourced decryption. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 1343–1354. [CrossRef]
3. Zhang, X.; Chen, X. Data security sharing and storage based on a consortium blockchain in a vehicular ad-hoc network. *IEEE Access* **2019**, *7*, 58241–58254. [CrossRef]
4. Zhang, X.; Wang, H.; Xu, C. Identity-based key-exposure resilient cloud storage public auditing scheme from lattices. *Inf. Sci.* **2019**, *472*, 223–234. [CrossRef]
5. Nedjah, N.; de Macedo Mourelle, L.; Wang, C. A parallel yet pipelined architecture for efficient implementation of the advanced encryption standard algorithm on reconfigurable hardware. *Int. J. Parallel Program.* **2016**, *44*, 1102–1117. [CrossRef]
6. Liang, K.; Au, M.H.; Liu, J.K.; Susilo, W.; Wong, D.S.; Yang, G.; Xie, Q. A DFA-based functional proxy re-encryption scheme for secure public cloud data sharing. *IEEE Trans. Inf. Forensics Secur.* **2014**, *9*, 1667–1680. [CrossRef]
7. Hussein, M.K.; Alhijaj, A.A. Protection of images by combination of vernam stream cipher, AES, and LSB steganography in a video clip. *Bull. Electr. Eng. Inform.* **2023**, *12*, 1578–1585. [CrossRef]
8. Han, X.; Yao, G. The combined use of FAPKC without compromising the security of the cryptosystem. *Jisuanji Yanjiu Yu Fazhan (Comput. Res. Dev.)* **2005**, *42*, 1692–1697. [CrossRef]
9. Panda, S.P.; Sahu, M.; Rout, U.P.; Nanda, S.K. Encryption and Decryption algorithm using two dimensional cellular automata rules in Cryptography. *Int. J. Commun. Netw. Secur.* **2011**, *1*, 18–23. [CrossRef]
10. Aljawarneh, S.; Yassein, M.B.; Talafha, W.A.A. A resource-efficient encryption algorithm for multimedia big data. *Multimed. Tools Appl.* **2017**, *76*, 22703–22724. [CrossRef]
11. Tao, R.; Chen, S. Two varieties of finite automaton public key cryptosystem and digital signatures. *J. Comput. Sci. Technol.* **1986**, *1*, 9–18. [CrossRef]
12. Gysin, M. *A One-Key Cryptosystem Based on a Finite Nonlinear Automaton*; Springer: Berlin/Heidelberg, Germany, 1995.
13. Dömösi, P. A novel cryptosystem based on finite automata without outputs. In *Automata, Formal Languages and Algebraic Systems*; World Scientific: Singapore, 2010; pp. 23–32. [CrossRef]
14. Roy, S.; Shrivastava, M.; Rawat, U.; Pandey, C.; Nayak, S. IESCA: An efficient image encryption scheme using 2-D cellular automata. *J. Inf. Secur. Appl.* **2021**, *61*, 102919. [CrossRef]
15. Tao, R. *Finite Automata and Application to Cryptography*; Springer: Chicago, IL, USA, 2008.
16. Mitchell, J.C.; Shmatikov, V.; Stern, U. Finite-State Analysis of SSL 3.0. In Proceedings of the 7th USENIX Security Symposium, San Antonio, TX, USA, 26–29 January 1998; Volume 1, pp. 201–216.
17. Isa, M.A.M.; Ahmad, M.M.; Sani, N.F.M.; Hashim, H.; Mahmod, R. Cryptographic key exchange protocol with message authentication codes (MAC) using finite state machine. *Procedia Comput. Sci.* **2014**, *42*, 263–270. [CrossRef]
18. Kearns, M.; Valiant, L. Cryptographic limitations on learning boolean formulae and finite automata. *J. ACM (JACM)* **1994**, *41*, 67–95. [CrossRef]
19. Sharipbay, A.A.; Saukhanova, Z.S.; Shakhmetova, G.B.; Saukhanov, N.S. Application of finite automata in cryptography. In Proceedings of the 5th International Conference on Engineering and MIS, ICEMIS'2019, Astana, Kazakhstan, 6–8 June 2019; pp. 1–3. [CrossRef]

20. Amorim, I.; Machiavelo, A.; Reis, R. *On Linear Finite Automata and Cryptography*; Tech. Rep. DCC-2011-11, Ver. 1.0; Faculdade De Ciências Universidade Do Porto: Porto, Portugal, 2011.
21. Saqib, Z.; Shahid, M.A.; Umair, M. Encryption and Decryption Using Automata Theory. *Int. J. Multidiscip. Sci. Eng.* **2015**, *6*, 14–21.
22. Vayadande, K.; Agarwal, K.; Kabra, A.; Gangwal, K.; Kinage, A. Cryptography using Automata Theory. In *ITM Web of Conferences*; EDP Sciences: Les Ulis, France, 2022; Volume 50, pp. 1–9. [CrossRef]
23. Khaleel, G.; Turaev, S.; Al-Shaikhli, I.; Tamrin, M.M. An overview of cryptosystems based on finite automata. *J. Adv. Rev. Sci. Res.* **2016**, *27*, 1–7.
24. Peña, P.I.S.; Torres, R.E.G. Authenticated Encryption based on finite automata cryptosystems. In Proceedings of the 2016 13th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), Mexico City, Mexico, 26–30 September 2016; pp. 1–6.
25. Alawida, M.; Samsudin, A.; Teh, J.S.; Alshoura, W.H. Deterministic chaotic finite-state automata. *Nonlinear Dyn.* **2019**, *98*, 2403–2421. [CrossRef]
26. Alawida, M.; Teh, J.S.; Alshoura, W.H. A New Image Encryption Algorithm Based on DNA State Machine for UAV Data Encryption. *Drones* **2023**, *7*, 38. [CrossRef]
27. Geng, S.; Wu, T.; Wang, S.; Zhang, X.; Wang, Y. Image Encryption Algorithm Based on Block Scrambling and Finite State Machine. *IEEE Access* **2020**, *8*, 225831–225844. [CrossRef]
28. Dougherty, S.T.; Klobusicky, J.; Şahinkaya, S.; Ustun, D. An S-Box construction from exponentiation in finite fields and its application in RGB color image encryption. *Multimed. Tools Appl.* **2024**, *83*, 41213–41241. [CrossRef]
29. Alawida, M.; Teh, J.S.; Samsudin, A. An image encryption scheme based on hybridizing digital chaos and finite state machine. *Signal Process.* **2019**, *164*, 249–266. [CrossRef]
30. Merhav, N. Perfectly secure encryption of individual sequences. *IEEE Trans. Inf. Theory* **2012**, *59*, 1302–1310. [CrossRef]
31. Khan, S.; Han, L.; Lu, H.; Butt, K.K.; Bachira, G.; Khan, N.U. A new hybrid image encryption algorithm based on 2D-CA, FSM-DNA rule generator, and FSBI. *IEEE Access* **2019**, *7*, 81333–81350. [CrossRef]
32. Benini, L.; De Micheli, G. Automatic synthesis of low-power gated-clock finite-state machines. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **1996**, *15*, 630–643. [CrossRef]
33. Sharipbay, A.; Saukhanova, Z.; Shakhmetova, G.; Barlybayev, A. Development of Reliable and Effective Methods of Cryptographic Protection of Information Based on the Finite Automata Theory. *Eurasia Proc. Sci. Technol. Eng. Math.* **2023**, *26*, 19–25. [CrossRef]
34. Bogachenko, N.F. Application of automata-theoretic models in cryptography. *Math. Struct. Model.* **2007**, *1*, 112–120.
35. Lotfi, Z.; Khalifi, H.; Ouardi, F. Efficient Algebraic Method for Testing the Invertibility of Finite State Machines. *Computation* **2023**, *11*, 125. [CrossRef]
36. Kohavi, Z.; Jha, N.K. *Switching and Finite Automata Theory*; Cambridge University Press: Cambridge, UK, 2009.
37. Shakhmetova, G.; Saukhanova, Z.; Udzir, N.I.; Sharipbay, A.; Saukhanov, N. Application of Pseudo-Memory Finite Automata for Information Encryption. In Proceedings of the 2nd International Workshop on Intelligent Information Technologies and Systems of Information Security, Khmelnytskyi, Ukraine, 24–26 March 2021; pp. 330–339.
38. Olson, R.R. *On the Invertibility of Finite State Machines*; Tech. Rep. TR-EE-703; Faculdade De Ciências Universidade Do Porto: Porto, Portugal, 1970.
39. Pareschi, F.; Rovatti, R.; Setti, G. On statistical tests for randomness included in the NIST SP800-22 test suite and based on the binomial distribution. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 491–505. [CrossRef]
40. Norouzi, B.; Mirzakuchaki, S.; Seyedzadeh, S.M.; Mosavi, M.R. A simple, sensitive and secure image encryption algorithm based on hyper-chaotic system with only one round diffusion process. *Multimed. Tools Appl.* **2014**, *71*, 1469–1497. [CrossRef]
41. Sharma, P.L.; Gupta, S.; Nayyar, A.; Harish, M.; Gupta, K.; Sharma, A.K. ECC based novel color image encryption methodology using primitive polynomial. *Multimed. Tools Appl.* **2024**, 1–40. [CrossRef]
42. Kaushik, P.; Attkan, A.A. Chaotic and Hyperchaotic Map based Image Encryption Protocol for High-End Colour density Images using enhanced S-box pixel permutator. In Proceedings of the 2021 2nd International Conference on Computational Methods in Science & Technology (ICCMST), Mohali, India, 17–18 December 2021; pp. 174–180.
43. Hermassi, H.; Rhouma, R.; Belghith, S. Improvement of an image encryption algorithm based on hyper-chaos. *Telecommun. Syst.* **2013**, *52*, 539–549. [CrossRef]
44. Ye, G.; Wong, K.W. An efficient chaotic image encryption algorithm based on a generalized Arnold map. *Nonlinear Dyn.* **2012**, *69*, 2079–2087. [CrossRef]
45. Song, C.Y.; Qiao, Y.L.; Zhang, X.Z. An image encryption scheme based on new spatiotemporal chaos. *Opt.-Int. J. Light Electron Opt.* **2013**, *124*, 3329–3334. [CrossRef]
46. Zeng, J.; Wang, C. A novel hyperchaotic image encryption system based on particle swarm optimization algorithm and cellular automata. *Secur. Commun. Netw.* **2021**, *2021*, 6675565. [CrossRef]
47. Liu, H.; Wang, X. Color image encryption using spatial bit-level permutation and high-dimension chaotic system. *Opt. Commun.* **2011**, *284*, 3895–3903. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.