Evaluation of Image Cryptography by Using Secret Session Key and SF Algorithm

Noor Kareem Jumaa
University of Technology, Computer Eng. Dept.
Baghdad, Iraq
noor.k.jumaa@uotechnology.edu.iq

Abbas Muhammed Allawy
Federal Public Services Council
Baghdad, Iraq
abbasmouhamd@gmail.com

Abstract
In the unreliable domain of data communication, safeguarding information from unauthorized access is imperative. Given the widespread application of images across various fields, ensuring the confidentiality of image data holds paramount importance. This study centers on the session keys concept, addressing the challenge of key exchange between communicating parties through the development of a random-number generator based on the Linear Feedback Shift Register. Both encryption and decryption hinge on the Secure Force algorithm, supported by a generator. The proposed system outlined in this paper focuses on three key aspects. First, it addresses the generation of secure and randomly generated symmetric encryption keys. Second, it involves the ciphering of the secret image using the SF algorithm. Last, it deals with the extraction of the image by deciphering its encrypted version. The system's performance is evaluated using image quality metrics, including histograms, peak signal-to-noise ratio, mean square error, normalized correlation, and normalized absolute error (NAE). These metrics provide insights into both encrypted and decrypted images, analyzing the extent to which the system preserves image quality. This assessment underscores the system's capability to safeguard and maintain the confidentiality of images during data transmission.

SF, LFSR, secure key exchange, image quality, histogram

I. INTRODUCTION

To ensure the confidentiality, integrity, authenticity, and accessibility of digital data, robust security measures should be implemented. Data security during network transmission can be guaranteed through encryption, which transforms data into an unreadable format for unauthorized individuals, decipherable only by authorized parties. In advanced technology, maintaining data protection through communication networks and the Internet becomes a significant challenge. Encryption emerges as a public method for enhancing image security, finding applications in various industries such as multimedia systems, military communication, internet communication, telemedicine, and medical imaging [1, 2].

Cryptography, also known as encryption or enciphering, employs mathematical procedures to transform understandable data (texts, images, audios, or videos), referred to as plaintext, into unintelligible data, known as ciphertext, rendering it incomprehensible without decryption. Decryption is the process of returning the encrypted message to its original format, making it readable [2, 3].

The use of computers, mobile devices, cell phones, or other communication equipment presents challenges to image security. Two types of digital image security encryption exist: **low-level security encryption** and **high-level security encryption**. Although the encrypted image under low-level security encryption may have lower visual quality than the original, it remains understandable to viewers. By contrast, high-level security encryption transforms the entire content, converting the image into random noise, rendering it unreadable to observers [2].

Symmetric key algorithms use the same key for both encryption and decryption. In secure key cryptography systems, also known as symmetric key cryptography systems, Alice and Bob use an identical key to encrypt and decrypt data during their communications. However, the logistical challenge of securely transferring the key between parties while restricting access by attackers poses a significant issue for symmetric encryption [1, 4].

The session key is unique and used only for a specific session. After use, the key is deleted, and a new key is randomly generated for the next session. This random key ensures the individuality of each encryption/decryption operation. A major challenge in symmetric cryptography systems is sharing an absolutely random key for both encryption and decryption operations [5].

This study investigates the encryption and decryption of grayscale images using the secure (symmetric) key SF cryptographic algorithm. A random key generator based on LFSR is employed to address the challenge of secret key distribution.

The remainder of this paper is structured as follows: A brief overview of related works is presented in Section 2. The modeling of the proposed system is discussed in Section 3, and the cryptographic scheme for the grayscale images is proposed in Section 4. The results and discussions are presented in Section 5, and the paper is concluded in Section 6.

II. Related Works

Kholood J. Moulood (2017) introduces a novel design for a pseudo-random generator, detailed in [6], aimed at creating binary sequences applicable as encryption keys in Stream Cipher Cryptosystems (SCC). The Address Shift LFSR (ASLFSR) cryptosystem, formed through a combination of nonlinear functions and LFSRs, employs LFSRs as the building blocks of the SCC. The ASLFSR generator's output undergoes analysis using Basic Efficient Criteria (BEC) to assess its performance as an efficient random number generator. The adherence of the ASLFSR cryptosystem to specified requirements underscores its ability to generate secure and unpredictable encryption keys.

Noor K. Jumaa (2018) proposes a method utilizing a random number generator to generate a secret key. The subsequently created random key is then employed for both encrypting and decrypting messages. The method, employing the Advanced Encryption Standard (AES) and the random key generator, successfully encrypts and decrypts grayscale and colored RGB images. Evaluation based on image quality metrics, including mean square error (MSE), peak signal-to-noise ratio (PSNR), normalized correlation (NK), and normalized absolute error (NAE), demonstrates the preservation of image quality, with plain and decrypted images being fully matched (MSE = 0 and NK = 1) [2].

Maisa'a A. Ali and Alyaa H. Zwiad (2019) utilize the SF algorithm for image encryption, as presented in [7]. Haar wavelet transform (HWT) is employed to convert plain images into frequency coefficients based on the Haar filter. Distortion measures such as PSNR, RMAE, MSE, and correlation measures are computed, revealing the efficiency, potency, and high security of the SF algorithm in cryptography.

Samer H. Majeed et al. (2020), through the application of the Taguchi method, as discussed in [8], demonstrate that an SF cryptographic system is a viable approach to encrypting images. Optimization experiments employing the L_9 orthogonal array highlight key parameters, including the symmetric ciphering/deciphering key, cryptography algorithms (SF), and image file extension type (JPG images), as crucial settings for obtaining optimal grayscale encrypted image quality. The study concludes that the SF algorithm, coupled with any manual key, represents the most effective cryptographic technique. Through the use of the Taguchi Method, insights into the rationale behind using JPG image types for encryption and steganography purposes are provided.

Balsam A. et al. (2022) employ Linear Congruential Generators (LCG) and Linear Feedback Shift Registers (LFSR) in their publication detailed in [9]. This approach combines these technologies to generate pseudo-random numbers, enhancing confidentiality and unpredictability. The results affirm the success of the tests and the resistance to differential and brute-force attacks. This hybrid technique proves effective for applications requiring reliable key generation.

Fatima F. Saleh and Nada H. M. Ali (2022) introduce a new method utilizing LFSR and the concept of chaotic images to generate the initial key. Genetic Algorithm (GA) is subsequently employed to create the final keys. The randomness of the generated key is verified using the NIST test group, with the P-value consistently $\geqslant 0.01$. This key is then utilized to encrypt images, as discussed in [10].

Mohammed A. and Saad Al-Momen (2023) engage in a discussion and comparison of two steganography techniques, outlined in [2]. The first technique operates in the spatial domain, utilizing the least significant bits (LSBs) for data embedding, achieving a typical PSNR of 43.5292 and a payload capacity of up to 16% of the cover image. The second technique operates in the frequency domain, concealing the secret message in the LSBs of the discrete cosine transform (DCT) coefficients in the medium-frequency area, offering a payload capacity of 8% and an average PSNR of 38.4092. This technique provides stronger defenses against attacks along with greater exposure.

III. Secure Force Cryptography

The Secure Force (SF) algorithm is a low-complexity cryptographic technology designed for WSN operations. Only five rounds of encryption are used to increase energy efficiency and reduce power usage. With each encryption round, four bits of data are subject to six direct mathematical operations, thus enhancing security. To make the data resistant to various forms of attack, this tactical strategy seeks to provide sufficient uncertainty and disseminate the data. To produce unique keys for various encryption rounds, the key expansion method uses complex mathematical processes (multiplication, permutation, transposition, substitution, and rotation). The decryption now carries the bulk of the calculation, extending the life of the sensor nodes. The encryption algorithm receives the generated keys in a secure manner to begin the encryption. It is strong, secure, and built for WSNs [7, 11, 12].

The overall SF algorithm comprises the following blocks [11]:

A) ***Key Expansion Block:*** The main method to generate unique keys for various encryption and decryption rounds used to secure images is known as key expansion. To enhance the efficiency of algorithm and eliminate the susceptibility of weak keys, various actions are implemented to confuse and diffuse information. The input cipher key is used to create the five distinct round keys. In this study, the initial input for key expansion is generated by the LFSR.

B) ***Encryption Block:*** The key results from the use of a key expansion block. Operations like AND, OR, XOR, XNOR, left shift (LS), substitution (S-boxes), and swapping are necessary for encryption to spread uncertainty and hazards. The plaintext of 64 bit has two halves of 32 bits each, which are further divided into two halves of 16 bits each, and then 16 bits are exchanged in each round.

C) ***Decryption Block:*** The decryption is the inverse of the encryption algorithm.

The details of the SF and its block structure can be seen in [11] and [12].

IV. Proposed System Modeling

The three main components of the proposed system are secret key generation, secret key distribution, and image encryption using SF. Subsequent subsections provide detailed coverage of each component.

A. *Session Key Generation*

To address the key generation aspect, this study employs the LFSR technique to generate a symmetric random key. The proposed technique randomly generates sixteen hexadecimal digits, serving as the cryptographic key utilized in both the encrypting and decrypting algorithms of SF.

When referring to "random numbers," the term actually denotes "pseudo-random numbers." This distinction arises because true random sequences are not employed; instead, pseudo-random sequences are generated through PRNGs. These PRNGs, based on internal equations, produce values that appear random and often align with various statistical definitions of randomness. All PRNGs have cycles, with the series of numbers repeating in the same order after completing one full cycle [2, 4, 13].

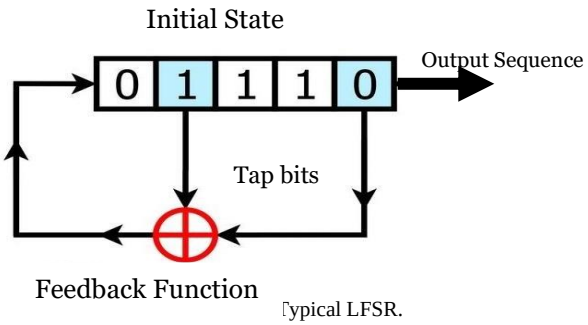
Applications that utilize cryptography, such as data encryption keys and secure communication channels, frequently rely on PRNGs based on LFSR. This preference is justified by the superior performance of LFSR-based PRNGs in terms of hardware, area, and speed compared to alternative counters [14].

In the LFSR, a feedback shift register is composed of two main components [2, 10]:

- Shift register
- Feedback function

The LFSR, a type of feedback shift register (FSR), executes the feedback function through the XOR operation on a subset of the register bits, forming a "tap sequence" [2, 13, 14]. Figure 1 illustrates a standard LFSR structure, and Table 1 provides an interface link between the tap sequence bits and the maximum length of the generated sequence.

Table 1 presents an interface link among the tap sequence bits and the maximum length of the sequence generated.



ypical LFSR.

An LFSR with n flip-flops produces (2^n - 1) distinct states, excluding the "all-zeros" state to prevent counter lockup. Pseudo-random numbers generated by LFSRs form "maximal-length sequences" that do not repeat until reaching the state of (2^n - 1). The following properties are found in the maximal sequence length generated [2]:

1. The number of 1s roughly equals the number of 0s.
2. The arithmetical distribution of 1s and 0s is consistently well defined.

Table I. LFSR maximal length taps for 2 to 24 bits.

Bits (n)	Taps	Period (2^n-1)
2	[0,1]	3
3	[0,2]	7
4	[0,3]	15
5	[1,4]	31
6	[0,5]	63
7	[0,6]	127
8	[1,2,3,7]	255
9	[3,8]	511
10	[2,9]	1,023
11	[1,10]	2,047
12	[0,3,5,11]	4,095
13	[0,2,3,12]	8,191
14	[0,2,4,13]	16,383
15	[0,14]	32,767
16	[1,2,4,15]	65,535
17	[2,16]	131,071
18	[6,17]	262,143
19	[0,1,4,18]	524,287
20	[2,19]	1,048,575
21	[1,20]	2,097,151
22	[0,21]	4,194,303
23	[4,22]	8,388,607
24	[0,2,3,23]	16,777,215

For detailed information on LFSR and polynomial feedback, refer to various references, with specific details available in [15].

In this study, a 16-digit hexadecimal session key, generated using a 5-bit LFSR with 31 random states, serves as the secret key for encryption and decryption procedures. The 4-bit LFSR with 15 states is unsuitable for the SF method due to its insufficient 16 hexadecimal digits. Algorithm 1 outlines the pseudocode for a 5-bit LFSR.

Five bits are selected at random from the day, month, and year bits in accordance with a predetermined agreement between the originator and receiver through any traditional communication tool. By exchanging the positions of these bits, an initial state can be established for the LFSR. Let the initial state be:

[Year (9), Day (3), Month (3), Day (5), Year (1)]. These bits will constitute elements within the key generator algorithm, forming an agreement between the sender (Alice) and the transmitter (Bob). The binary date will serve as the initial state for the LFSR, as explained below.

Day	Month	Year
11001	0111	11111100111
Initial state = 10111		

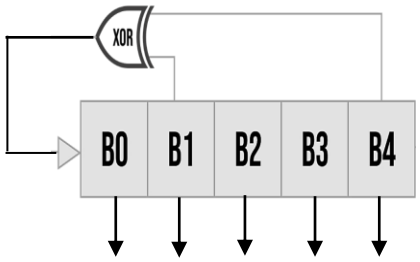
For initial state 10111, 1 is the ninth bit in the year binary form, 0 is the third bit in the day binary form, 1 is the third bit in the month binary form, 1 is the fifth bit in the day binary form, and 1 is the first bit in the year binary form, as outlined in the previous array [Year (9), Day (3), Month (3), Day (5), Year (1)]. This array defines the specifics of the initial state of the LFSR, marking the agreement between the sender (Alice) and the receiver (Bob).

Algorithm (1): LFSR Session Key Generation

Input: initial state of 5 bits, taps bits [2, 9], n length of required LFSR-session key
Output: LFSR-session key

Start
Step 1: take the hexadecimal form of the LFSR bits to be output as the LFSR-session key.
Step 2: shifting to the RIGTH all bits in LFSR stream by one step.
Step 3: doing an XOR operation between tap bits to be the next first bit in the LFSR stream.
Step 4: INSERT the next first bit in position 1 from the LFSR stream.
Step 5: repeat steps 1 to 5 until the length of LFSR-session key equals n.
End

Figure 2 illustrates the 5-bit LFSR used as a secret key generator. The random encryption/decryption hexadecimal key is [171B0D060211181C].



	Bit_0	Bit_2	Bit_3	Bit_3	Bit_4	FB	Output
1	1	0	1	1	1	**1**	**17**
2	1	1	0	1	1	**0**	**1B**
3	0	1	1	0	1	**0**	**0D**
4	0	0	1	1	0	**0**	**06**
5	0	0	0	1	1	**1**	**02**
6	1	0	0	0	1	**1**	**11**
7	1	1	0	0	0	**1**	**18**
8	1	1	1	0	0	**1**	**1C**

Fig 2. Random Key Generator.

B. Secure Key Distribution

In symmetric key cryptography systems, the communicating parties (Alice and Bob) use the same secret key for both encryption and decryption. However, ensuring the secure transfer of this key between the two parties, while preventing access by potential attackers, poses a significant challenge [16].

By configuring the initial state of the random key generator based on the current date for each ciphering process, this paper addresses the challenge of secret key distribution. Both Alice and Bob are well-versed in the 5-bit LFSR algorithm for key generation, as well as the associated processes of encryption and decryption. Alice can transmit a concise message to Bob, such as "Year (9), Day (3), Month (3), Day (5), Year (1)," to inform him of the initial state, facilitating the exchange of the secret key. Subsequently, Bob utilizes these bits as the initial state for the LFSR algorithm, thereby obtaining the secret key.

The following presumptions underpin the scenario presented in this study:

1. Both Alice and Bob are acquainted with the encryption and decryption algorithms.
2. Both Alice and Bob are knowledgeable about the 5-bit LFSR secret key generation algorithm.
3. Alice and Bob share solely the initial state, derived from the encrypted image's date of delivery or receipt.

Figure 3 shows how Alice and Bob communicate using the suggested method to share the secret key.

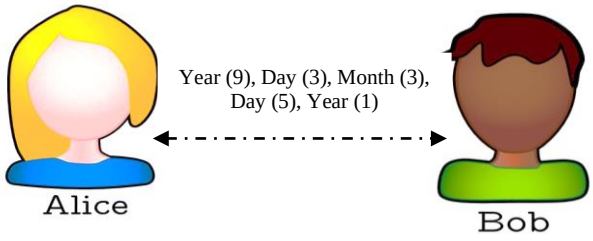


Fig. 3. Sharing of Secret Key.

V. Proposed Cryptographic System Image Ciphering

The encryption for grayscale images, detailed in Algorithm 2, is depicted in Figure 4. Figure 5 illustrates the decryption, as outlined in Algorithm 3.

The SF cryptographic system takes both the plain image and the 64-bit session key generated by the LFSR as input. The plain image undergoes division into blocks of 8 bytes each. Subsequently, the SF cryptographic system combines each block with an 8-byte (64-bit) key generated by the LFSR for encryption, resulting in the generation of a cipher image. As each 8-byte cipher block is assembled, the resulting cipher image presents itself to the viewer as a seemingly nonsensical image.

This entire cryptographic system was implemented using Matlab 2018a on an HP Pavilion PC equipped with an Intel Core i7 CPU and running a 64-bit Windows 11 OS. The decryption mirrors the encryption procedures in reverse.

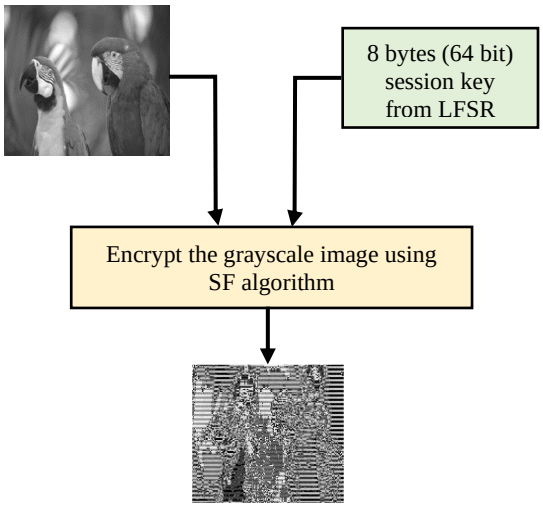


Fig 4. Structure of the Ciphering System for Grayscale image.

Algorithm (2): Image encryption by session key and SF algorithm
Input: 256 × 256 grey-scale plain image, 16 digit hexadecimal session key **Output:** cipher image
Start **Step 1:** divide the image matrix into blocks with 64 bits each. **Step 2:** convert each pixel to binary. **Step 3:** convert session key to binary. **Step 4:** do the SF encryption operations between the binary image block and the binary session key. **Step 5:** convert each 64-bit block to a decimal. **Step 6:** merge blocks to be a 256×256 image matrix. **End**

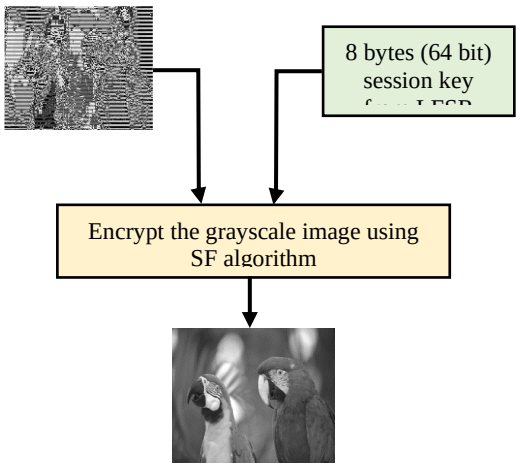


Fig 5. Structure of the Deciphering System for Grayscale image.

Algorithm (3): Image decryption by session key and SF algorithm
Input: 256 × 256 cipher image, 16 digit hexadecimal session key **Output:** decipher image
Start **Step 1:** divide the cipher image matrix into blocks with 64 bits each. **Step 2:** convert each pixel to binary. **Step 3:** convert session key to binary. **Step 4:** do the SF decryption operations between the binary image block and the binary session key. **Step 5:** convert each 64-bit block to a decimal. **Step 6:** merge blocks to be a 256×256 image matrix. **End**

VI. Results and Discussions

The outcomes obtained in this study are assessed based on the quality of images, with a focus on visual results. The following parameters are used to evaluate quality of images:

I. **Histogram** serves as a gauge of unpredictability or uncertainty inherent in the pixel values of an image. It functions as a statistical measure, quantifying the information encapsulated within an image. The computation of an image's entropy entails the utilization of the pixel value histogram. Images characterized by higher entropy are typically more intricate, whereas those with lower entropy may exhibit a greater degree of simplicity or uniformity. This paper undertakes the measurement of histograms for plain, encrypted, and decrypted images [17].

Histogram can be calculated using the frequency density obtained with the following formula:

$$D = \frac{W}{F} \quad \ldots\ldots\ldots\ldots\ldots (1)$$

Where D is the frequency density of a class interval in an image, F is the frequency, and W is the class width.

II. **Peak Signal to Noise Ratio (PSNR)** functions as a measure for the compressed reconstruction of an image. In this study, PSNR was computed between the original plain image and the ciphered image, as well as between the plain image and the deciphered image.

$$PSNR = \frac{10Log255^2}{MSE} \quad \ldots\ldots\ldots\ldots\ldots (2)$$

III. **Mean Square Error (MSE)** is the average of the squared intensity differences between two images. In this study, MSE was calculated between the original plain image and the ciphered image, as well as between the plain image and the deciphered image. Following the PSNR, MSE stands out as one of the most frequently employed quality parameters.

$$MSE = \frac{1}{MN}\sum_1^M \sum_1^N \left(f(i,j) - \bar{f}(i,j)\right)^2 \quad \ldots\ldots (3)$$

Where N and M are the sizes of the image's matrix, *f(i,j)* is the original image, and $\bar{f}(i,j)$ is the encrypted/decrypted image.

IV. **Normalized Correlation (NK)** assesses the similarity between two images, specifically the plain image and the encrypted image. Elevated NK levels suggest diminished image quality. This study includes calculations of NK between the original image and the ciphered image as well as between the original image and the deciphered image.

$$NK = \frac{\sum_1^M \sum_1^N \left(f(i,j).\bar{f}(i,j)\right)}{\sum_1^M \sum_1^N \left(f(i,j)\right)^2} \quad \ldots\ldots\ldots\ldots\ldots (4)$$

V. **Normalized Absolute Error (NAE)** is a metric for gauging the degree of dissimilarity between the modified image and the original image, with a value of zero indicating an exact match. This study computes the NAE between the original image and the ciphered image as well as between the original image and the deciphered image.

$$NAE = \frac{\sum_1^M \sum_1^N |f(i,j)-\bar{f}(i,j)|}{\sum_1^M \sum_1^N |f(i,j)|} \quad \ldots\ldots\ldots\ldots\ldots (5)$$

Further details on PSNR, MSE, NK, and NAE are available in [18, 19]. Table 2 provides a comprehensive presentation of histogram measurements, serving as one of the image quality parameters for plain images, encrypted images, and decrypted images.

Table II . Histogram readings.

Image Name	Entropy		
	Original Image	Encrypted Image	Decrypted Image
Cameraman.tif	7.0097	7.8787	7.0097
Rice.tif	7.0115	7.9170	7.0115
Lina.jpg	7.4592	7.9323	7.4592
Football.jpg	6.6902	7.8851	6.6902
Onion.png	7.3325	7.9396	7.3325
Birds.png	7.2813	7.9111	7.2813

Within the context of image cryptography, a histogram is a visual representation of the frequency distribution of pixel values present in an image. It elucidates the number of pixels in the image with a specific intensity or color value. In this research, 8-bit grayscale images were employed. Consequently, the horizontal axis of the histogram illustrates the range of pixel values (from 0 to 255), whereas the vertical axis portrays the proportion of pixels corresponding to each value.

Table 2 reveals that the histogram values of the decrypted images closely align with those of the original plain images, signifying the successful recovery of the original pixel intensity during decryption. The congruence of histogram values in both decrypted and plain images is a pivotal indicator that the SF image cryptographic system effectively maintains the image's integrity throughout encryption and decryption. The images featured histograms with seven bins covering the entire spectrum of possible pixel frequencies, ranging from 0 to 255. Across this full range, the histogram delineates the distribution of pixels within each of the seven intervals. Each bin represents a distinct range of pixel values. This type of histogram, presenting the distribution of pixel intensities on a broader scale, proves valuable when reducing the level of detail in pixel intensity information. Figures 6 to 11 illustrate the histogram of the plain, encrypted, and decrypted images, along with the vision illustrations of these images. Table 3 lists the parameters measuring the image quality for grayscale images of (plain-encrypted) and (plain-decrypted) pairs.

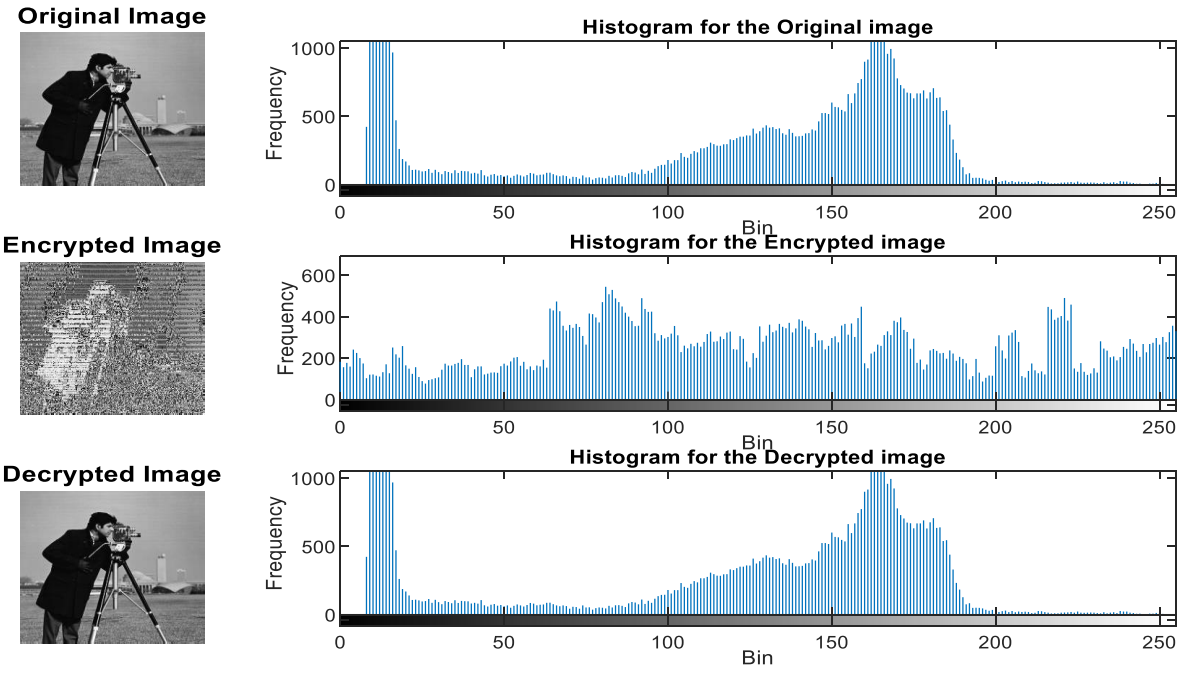
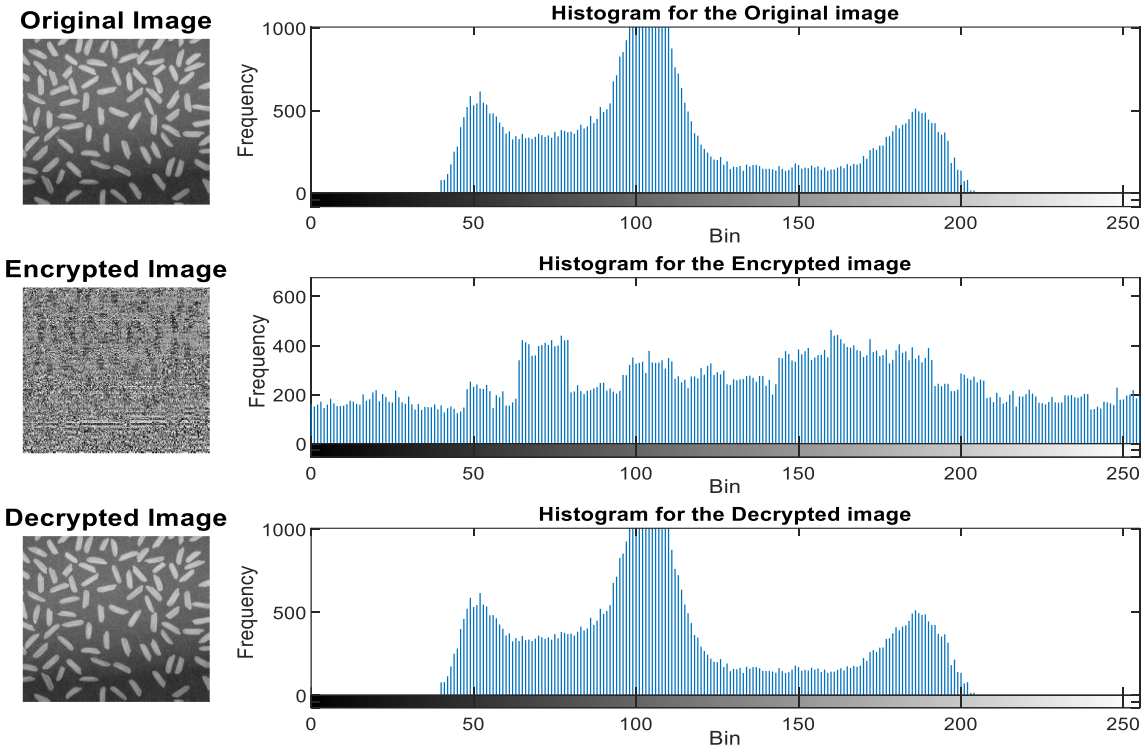


Figure 6. Histogram and vision results of



cameraman tif image

Fig 7. Histogram and vision results of rice tif image.

Fig 8. Histogram and vision results of Lina jpg image.

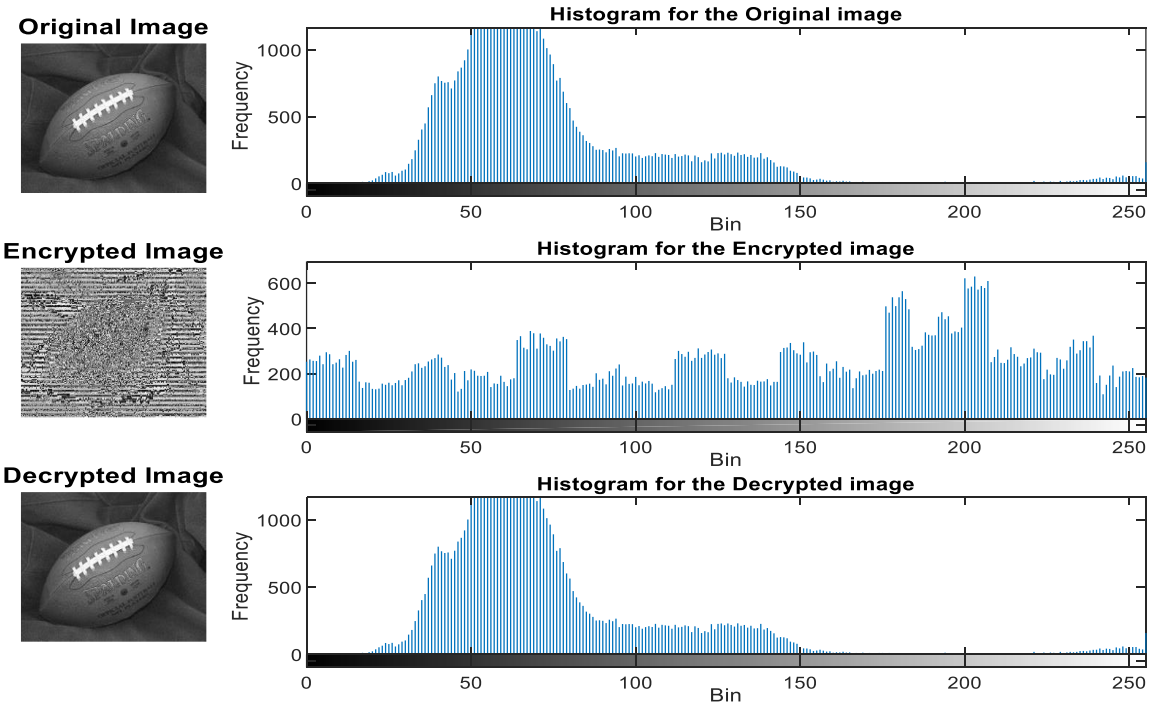


Fig 9. Histogram and vision results of football jpg image.

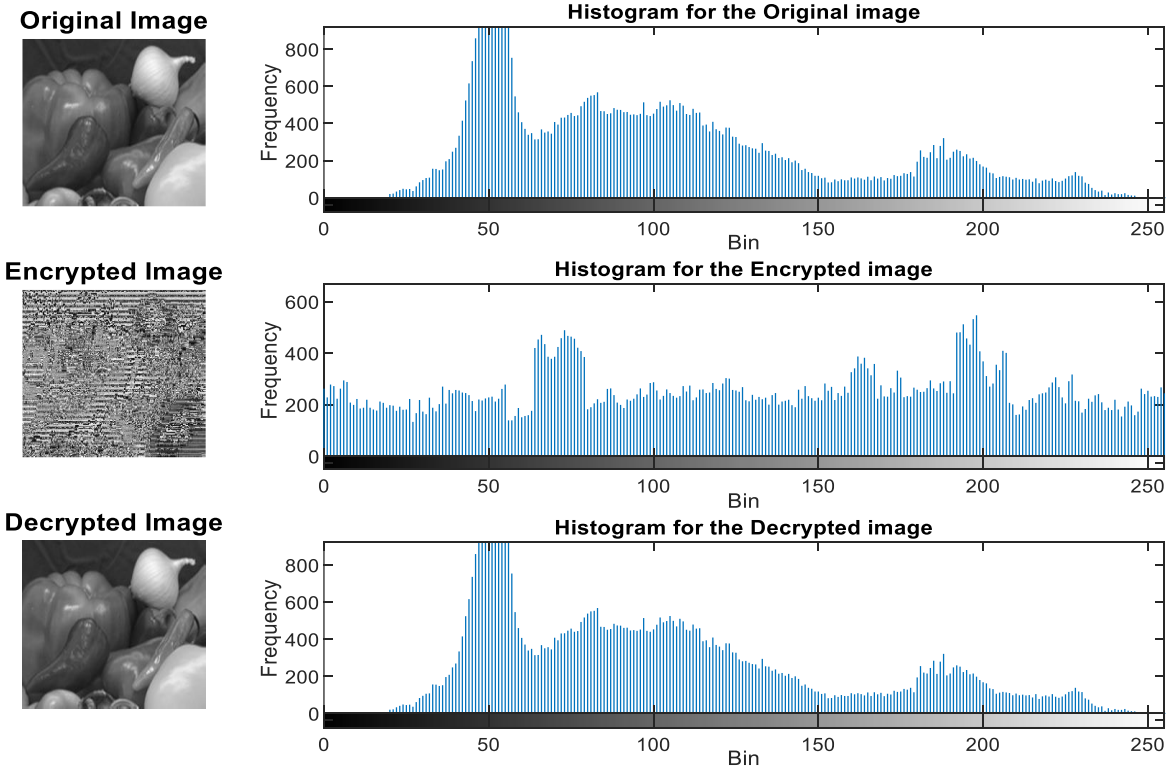


Fig.10. Histogram and vision results of onion png image.

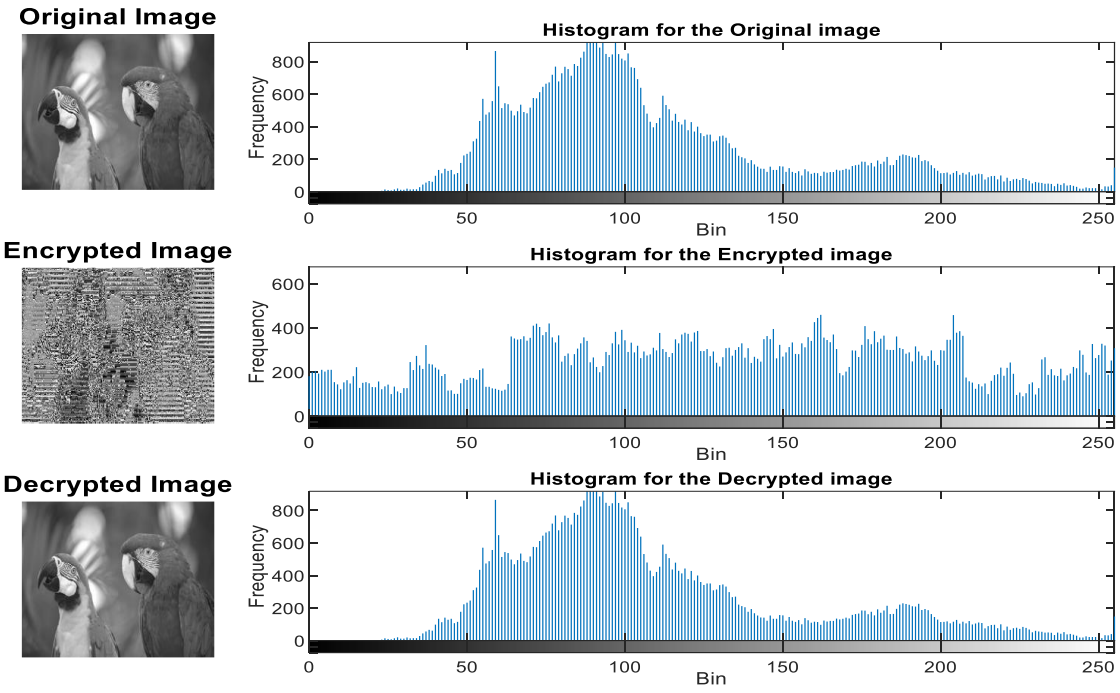


Fig.11. Histogram and vision results of bird png image.

Table III. Histogram readings.

MSE		PSNR		NK		NAE	
Encrypted-Original	Decrypted-Original	Encrypted-Original	Decrypted-Original	Encrypted-Original	Decrypted-Original	Encrypted-Original	Decrypted-Original
105.5244	0	1.0502	∞	1.0170	1	0.3647	0
140.4904	0	0.7888	∞	1.0232	1	0.3404	0
109.7439	0	1.0099	∞	1.0300	1	0.2764	0
178.1490	0	0.6221	∞	1.0345	1	0.5395	0
147.2008	0	0.7529	∞	1.0330	1	0.4175	0
148.9031	0	0.7443	∞	1.0269	1	0.3650	0

The calculated MSE values for the six images varied, ranging from a minimum of 105.5244 to a maximum of 178.1490, encompassing a span of 256 pixels. This considerable range underscores a significant difference between the encrypted image and the original one. The MSE for plain-decrypted images was 0, indicating that neither encryption nor any other form of image manipulation altered it in any way.

For plain encrypted images, the PSNR values exhibited a range from 0.6221 to 1.0502 across the six images. This implies a low signal-to-noise ratio, signifying increased distortion in the encrypted images. This distortion is attributed to the SF encryption and its data manipulation. Conversely, a PSNR value of ∞ for plain-decrypted images implies an indiscernible difference between the original plain image and the decrypted image.

The NK values for plain-encrypted images were close to 1, indicating a positive correlation between the plain and encrypted images. This suggests that the encryption method preserved the most crucial details of the image with minimal alteration. A NK value of 1 for plain-decrypted images signifies a perfect match between the original plain and the decrypted images, affirming the precise restoration of the image without any loss.

The NAE values for plain-decrypted images ranged from 0.3647 to 0.5395, highlighting the dissimilarity between the plain and decrypted images. The SF algorithm successfully restored the original plain image without introducing distortion or error, as evidenced by recorded NAE values of 0 for the plain-decrypted images.

VI. Concluded Remarks

This research introduces a secure key exchange method employing session keys generated by a LFSR, applicable to any cryptographic system. The efficacy of the SF algorithm in image encryption is substantiated through comprehensive image quality assessments. This proposed key exchange technique involves utilizing the LFSR as a generator for the 16 hexadecimal digits (64 bits), forming session keys crucial for encrypting and decrypting images with the SF algorithm.

The effectiveness of the SF algorithm in image encryption/decryption is underscored by its performance evaluation using various image quality measures. The MSE values distinctly differentiate between encrypted and original images, whereas an MSE of 0 for plain and decrypted images highlights the algorithm's ability to maintain image integrity. Variable distortion levels in encrypted images, revealed by PSNR values, result from the SF modification. A PSNR of ∞ for plain and decrypted images underscores the precision of the technique in restoring original images. Moreover, the NK readings of 1 for plain and decrypted images, along with the NAE scores of 0, indicate that the SF algorithm accurately returns the decrypted data of images to their exact origin, regardless of their file type (tif, jpg, or png). Although NK and NAE values suggest extreme similarity between plain and encrypted images, Figures 6 to 11 visually demonstrate that encrypted images closely resemble their plain counterparts. This underscores the SF method's efficacy in preserving the main features of original images during encryption and decryption, albeit with potential considerations as a drawback for the SF cryptographic technique when applied to images.

REFERENCES

[1] Wisam Abed Shukur, Luheb Kareem Qurban, and Ahmed Aljuboori, "Digital Data Encryption Using a Proposed W-Method Based on AES and DES Algorithms", Baghdad Science Journal, Vol. 20, No. 4, 2023. : https://dx.doi.org/10.21123/bsj.2023.5147

[2] Noor Kareem Jumaa, "Digital Image Encryption using AES and Random Number Generator", Iraqi Journal for Electrical and Electronic Engineering (ISSN: 1814-5892), Vol. 14, Issue 1, 2018. https://ijeee.edu.iq/Papers/Vol14-Issue1/144343.pdf

[3] Farah R. Shareef, "A novel crypto technique based ciphertext shifting", Egyptian Informatics Journal, Vol. 21, Issue 2, 2020. https://doi.org/10.1016/j.eij.2019.11.002

[4] Mohammed Abod Hussein and Saad Al-Momen, "Linear Feedback Shift Registers-Based Randomization for Image Steganography", Iraqi Journal of Science (ISSN: 0067-2904), Vol. 64, No. 8, pp: 5031-5046, 2023. https://ijs.uobaghdad.edu.iq/index.php/eijs/article/view/10111

[5] Hiba A. Taresh, "Proposed Lightweight Protocol for IoT Authentication", Iraqi Journal for Computers and Informatics, Vol. 44, Issue 1, 2018. https://www.iasj.net/iasj/download/6e3b7ceb3410da56

[6] Kholood J. Moulood, "New Address Shift Linear Feedback Shift Register Generator", Al-Nahrain Journal of Science **(ISSN: 2663-5461)**, Vol. 20, No. 1, pp.142-148, 2017. https://anjs.edu.iq/index.php/anjs/article/view/67/43

[7] Maisa'a Abid Ali Khodher and Alyaa Hasan Zwiad, "Proposal Cryptography Algorithm Based On Bit Plane Image Slicing Using Wavelet Transform", AL-yarmouk Journal, Vol. 9, Issue 9, 2017. https://www.iasj.net/iasj/download/ba5f370e633dc335

[8] Samer Hammed Majeed , Noor Kareem Jumaa, and Auday A.H. Mohamad, "Taguchi Optimization Method for Testing Best Image Encryption Algorithm", International Journal of Computing and Digital Systems (ISSN: 2210-142X), Vol. 9, No.5, 2020. http://dx.doi.org/10.12785/ijcds/090520.

[9] Balsam Abdulkadhim Hameedi, Dr Anwar Abbas Hattab, and Dr Muna M. Laftah, "A Pseudo-Random Number Generator Based on New Hybrid LFSR and LCG Algorithm", Iraqi Journal of Science (ISSN: 0067-2904), Vol. 63, No. 5, pp: 2230-2242, 2022. https://ijs.uobaghdad.edu.iq/index.php/eijs/article/view/4396

[10] Fatima Faiz Saleh and Nada Hussein M. Ali, "Generating Streams of Random Key Based on Image Chaos and Genetic Algorithm", Iraqi Journal of Science, Vol. 63, No. 8, pp: 3652-3661, 2022. https://doi.org/10.24996/ijs.2022.63.8.39%20

[11] Kakshak Porwal, "Comparison between image encryption algorithms for wireless sensor", International Journal of Advance Research, Ideas and Innovations in Technology, Vol. 5, Issue 5, 2019. https://www.ijariit.com/manuscripts/v5i5/V5I5-1150.pdf

[12] Muhammad Usman, Syed Zain-Ul-Abedin Abidi, Muhammad Hassam Shakil Siddiqui, and M.Sohail Ibrahim, "Implementation of Secure Force (64-bit) on low cost 8-bit

Microcontroller", Conference paper, 2017, Researchgate. https://doi.org/10.1109/ICOSST.2016.7838585.

[13] G M Sridevi, Ashoka D V, and B V Ajay Prakash, "Partial Pseudo-Random Hashing for T tial Pseudo-Random Hashing for Transactional Memor ansactional Memory Read/W y Read/Write Data Processing and Validation", Karbala International Journal of Modern Science, Vol. 8, Issue 2, 2022. https://kijoms.uokerbala.edu.iq/home/vol8/iss2/7/

[14] Sony Smitha, Anirudh P.V., and Geethu R.S., "Design and Analysis of Multi-Bit Linear Feedback Shift Register based PRNG with FPGA Implementation using different Primitive Polynomials", 2nd IEEE International Conference on Intelligent Technologies, CONIT, 2022. Doi: 10.1109/CONIT55038.2022.9848174

[15] A. K. Panda, P. Rajput, and B. Shukla, "FPGA Implementation of 8, 16 and 32 Bit LFSR with Maximum Length Feedback Polynomial Using VHDL", IEEE International Conference on Communication Systems and Network Technologies, 2012. DOI:10.1109/CSNT.2012.168

[16] May H. Abood and Sarah W. Abdulmajeed, "High Security Image Cryptographic Algorithm Using Chaotic Encryption Algorithm with Hash-LSB Steganography", Al-Iraqia Journal for Scientific Engineering Research, Vol. 1, Issue 2, 2022. https://doi.org/10.58564/IJSER.1.2.2022.53

[17] Acharya, U.K., Kumar, S., "Image sub-division and quadruple clipped adaptive histogram equalization (ISQCAHE) for low exposure image enhancement", Multidim Syst Sign Process, Vol. **34**, P.P: 25–45, 2023. https://doi.org/10.1007/s11045-022-00853-9

[18] S. Rajkumar and G. Malathi, "A Comparative Analysis on Image Quality Assessment for Real Time Satellite Images", Indian Journal of Science and Technology, Vol. 9, No. 34, 2016. DOI: 10.17485/ijst/2016/v9i34/96766

[19] L. M. Satapathy and P. Das, "A New Approach of Image Denoising Based on Adaptive Multi-Resolution Technique", NIGERIAN JOURNAL OF TECHNOLOGICAL DEVELOPMENT, VOL. 19, NO. 1, 2022. http://dx.doi.org/10.4314/njtd.v19i1.10