“An Image Encryption Scheme using Modified SNOW 3G Algorithm”

Developed By

Name	University Registration Number	University Roll Number
Sourav Kundu	003669-(2019-2020)	430119010015
Ritobhas Ray	003773-(2019-2020)	430119010011
Prakash Kumar Haldar	20127010020007	430120110211
Soham Chakma	20127010020008	430120110214

Under the Supervision of

Rahul Das Gupta

Assistant Professor

Department of Computer Science, NiT

(May 2023)

An Image Encryption Scheme using Modified SNOW 3G Algorithm

A Dissertation Submitted in partial fulfillment for the Degree of Bachelor of Technology (B. TECH), 8th Semester in Computer Science & Engineering

Submitted By

NAME	UNIVERSITY REGISTRATION NUMBER	UNIVERSITY ROLL NUMBER
Sourav Kundu	003669-(2019-2020)	430119010015
Ritobhas Ray	003773-(2019-2020)	430119010011
Prakash Kumar Haldar	20127010020007	430120110211
Soham Chakma	20127010020008	430120110214

Under the Supervision of

Rahul Das Gupta
Assistant Professor
Department of CSE, NiT

Narula Institute of Technology

MAULANA ABUL KALAM AZAD
UNIVERSITY OF TECHNOLOGY,
WEST BENGAL

Utech

In Pursuit of Knowledge And Excellence

Maulana Abul Kalam Azad University of Technology (May 2023)

CERTIFICATE OF ORIGINALITY

The project entitled "**An Image Encryption Scheme using Modified SNOW 3G Algorithm**" has been carried out by ourselves in partial fulfillment of the degree of Bachelor of Technology in Computer Science & Engineering of Narula Institute of Technology, Agarpara, Kolkata under Maulana Abul Kalam Azad University of Technology during the academic year 2019-2023

While developing this project no unfair means or illegal copies of software etc. have been used and neither any part of this project nor any documentation have been submitted elsewhere or copied as far in our knowledge.

Signature:
Name: SOURAV KUNDU
University Roll No.:430119010015
University Registration No.:003669 of 2019-2020

Signature
Name: RITOBHAS RAY
University Roll No.:430119010011
University Registration No.: 003733 of 2019-2020

Signature
Name: PRAKASH KUMAR HALDAR
University Roll No.: 430120110211
University Registration No.: 201270100120007

Signature
Name: SOHAM CHAKMA
University Roll No.:430120110214
University Registration No: 201270100120008

CERTIFICATE OF APPROVAL

This is to certify that the project entitled "**An Image Encryption Scheme using Modified SNOW 3G Algorithm**" has been carried out by Sourav Kundu, Ritobhas Ray, Prakash Kumar Haldar, Soham Chakma, under my supervision in partial fulfillment for the degree of Bachelor of Technology (B. TECH) in Computer Science & Engineering of Narula Institute of Technology, Agarpara affiliated to Maulana Abul Kalam Azad University of Technology during the academic year 2019-2023

It is understood that by this approval the undersigned do not necessarily endorse any of the statements made or opinion expressed therein but approves it only for the purpose for which it is submitted.

Submitted By:

Name: SOUROV KUNDU	Name: PRAKASH KUMAR HALDAR
University Roll No.:430119010015	University Roll No.:430120110211
University Registration No:003669-(2019-2020)	University Registration No:201270100120007
Name: RITOBHAS RAY	Name: SOHAM CHAKMA
University Roll No.:430119010011	University Roll No.:430120110214
University Registration No:003669-(2019-2020)	University Registration No:201270100120007

Rahul Das Gupta

Assistant Professor

Department of CSE, NiT

(External Examiner)

(Dr. Subhram Das)

HOD, CSE Dept

ACKNOWLEDGEMENT

We would like to express my sincere gratitude to all those who have supported me in the completion of this project. First and foremost, we would like to thank my project guide **Rahul Das Gupta**, for their invaluable guidance and support throughout the project. Their insights and expertise have been instrumental in shaping this project and making it a success.

We’re also thankful to **Subrata Nandi**, assistant professor of CSE dept, Narula Institute of Technology, for his insights on the domain of cryptography.

We would also like to thank my college faculty members of Narula Institute of Technology, who provided me with the necessary resources and guidance to undertake this project. Their constant encouragement and support have been a source of motivation for me.

Last but not least, we would like to thank all the participants who took part in the study and provided valuable data for the project. Their participation has been essential in making this project a success. Thank you all for your invaluable support and guidance. We are truly grateful for all the help and encouragement that we have received throughout this project.

CONTENTS

INTRODUCTION

In today's world, digital images play a critical role in multimedia communications that are happening at a breakneck pace because of Internet of Things (IoT) applications. However, if the data is sensitive, it is critical to ensure its protection. Even though the Internet of Things architecture provides numerous advantages to humanity, data transfers pose various security threats, particularly when sensitive images are transferred. These sensors gather data from their surroundings and broadcast it across unsecured public networks. In this context, the adversary can play a crucial role in taking advantage of the system by manipulating the data via various security attacks. As IoT application sensors have some storage and computation constraints, providing security to those applications becomes difficult. Traditional ciphers, as a result, cannot be utilized in IoT devices. SNOW 3G Algorithm , on the other hand, may be used to provide security to IoT devices in this resource-constrained context because this stream cipher is capable of constructing complicated patterns and pseudo-random sequences efficiently at high speed. It is also simple to put into hardware. Role of cloud-based IoT infrastructure is vital in the present digital world. In this configuration, small devices use sensors to collect data and send it over the Internet to cloud storage servers.

These data are utilized for further analysis and processing at the application layer [1]. Actuators and sensors are deployed at the perception layer [2]. In contrast, gateway devices with some computational capabilities are installed at the network layer in a typical IoT network scenario. These sensors collect data from regions humans cannot observe directly. In the application layer, users are allowed to interact with the cloud. Many traditional ciphers are available to maintain information security at the network or application layers. However, these cannot be used at the perception layers due to the resource limits of the sensory equipment. As a result, a fast and lightweight encryption technique in the perception layer.

Word Oriented Stream cipher [3,4,5] plays an important role in modern day communication. In 4G and 5G communications, SNOW 3G [6] cipher is used to keep the confidentiality and integrity of the data. Moreover, it gives 128−bit security, and it has high throughput. It can be used efficiently in hardware, software or embedded system devices. In short, the major contributions of this work are listed below:

- A modified SNOW 3G KSG by changing the Linear Feedback Shift Register (LFSR) with 64 input-output, 8 delay blocks σ−LFSR is proposed. A feedback configuration matrix [7,8] has been used instead of the feedback matrix of SNOW 3G to reduce the encryption time of traditional cipher and enhance the randomness of the keystream.
- Security aspect of the KSG is proved theoretically. Furthermore, the randomness of the proposed KSG is evaluated by the NIST randomness test suit.
- Proposed image encryption scheme is validated by standard tests like information entropy test, histogram analysis, correlation coefficients, NPCR, UACI etc.

MOTIVATION

The increasing prevalence of digital images in various domains, such as healthcare, finance, and multimedia, has raised concerns about the security and privacy of visual data. As images contain sensitive information, ensuring their confidentiality and integrity has become a critical challenge. Traditional encryption schemes are often inadequate for securing images due to their unique characteristics, such as large file sizes, complex structures, and susceptibility to various attacks.

In this context, the modified SNOW 3G algorithm presents an exciting opportunity for image encryption. Originally designed for securing telecommunications, the SNOW 3G algorithm possesses several desirable properties that make it suitable for image encryption. It offers high security, excellent resistance against known attacks, and efficient implementation on modern hardware platforms.

The motivation behind utilizing the modified SNOW 3G algorithm lies in its adaptability and robustness. By modifying the algorithm to suit the specific requirements of image encryption, we can leverage its strengths to protect visual data effectively. The modified version can address the challenges posed by image characteristics, including pixel dependencies, color channels, and spatial correlations, while maintaining the algorithm's high level of security.

Moreover, the modified SNOW 3G algorithm allows for efficient encryption and decryption processes, enabling real-time or near-real-time applications that require quick processing of large image datasets. This capability is crucial in domains like video surveillance, where the timely and secure transmission of images is of utmost importance.

Furthermore, the use of the modified SNOW 3G algorithm in image encryption contributes to the broader goal of standardization and interoperability. By leveraging an established algorithm, we can ensure compatibility and interoperability across different systems and applications. This aspect is particularly important for scenarios where images are shared across different platforms, such as medical imaging and collaborative multimedia projects.

Overall, the motivation behind employing the modified SNOW 3G algorithm for image encryption lies in its adaptability, security, efficiency, compatibility, and standardization benefits. By leveraging this algorithm's strengths and addressing the unique challenges of image encryption, we can develop a robust and practical framework for safeguarding visual data. This project aims to explore the potential of the modified SNOW 3G algorithm, evaluate its effectiveness in image encryption, and contribute to the advancement of secure image transmission in various domains.

SURVEY

There are plenty of works on image encryption using various techniques. In this section, some of the recent studies have been presented. A good review of various types of the image encryption scheme is presented in [9]. The majority of the techniques are a mixture of more than two different techniques. A novel technique to generate a pseudo random key for encryption was proposed by A. Kumar et al. [10]. This pseudo-random key is used to build three secure and efficient methods for satellite image encryption: Logistic map (LM) [11], cosine transformed Logistic map (CTLM) [12,13], and cosine transformed Logistic-Sine Map (CTLSM) [14]. During the encryption procedure, the technique employs the 384-bit sharing key.

Xiang et al. proposed an improvement model to suppress the difference dynamic degradation of chaotic maps infinite precision devices. After that, they used part of an input image and the improved chaotic map to generate part of the key for image encryption. This scheme effectively resists various security attacks and achieves good values of entropy, NPCR and UACI. Jithin et al. [19] combined Arnold's chaotic map with DNA codes for encryption. In addition, they used a Mandelbrot set2 to add more confusion to the cipher images. Their scheme achieved better security features when compared to other existing schemes.

Image encryption scheme using DNA computing was proposed by many researchers in recent years [28,29,30,31]. These techniques make use of DNA computing along with the chaotic map. It results in the key space being large enough to resist brute-force attacks. Besides, these techniques provide better correlation coefficient values, MSE and PSNR.

Babaei et al. presented an image encryption scheme based on CA and DNA sequences; however, it requires a significant amount of CPU and memory, making it unsuitable for IoT applications.

There are many works on image encryption using deep learning [27,32,33]. These works utilize cycle-GAN and Deep Neural Network (DNN) for training the encryption module. The performance of these schemes is good at the cost of resources and computations.

Similarly, meta-heuristic-based image encryption techniques [34,35] also suffer from slow encryption speed because of their operational complexities.

Although there exist many image encryption techniques in the literature that produce high-quality cipher images, most of these are slow and unsuitable for use in resource-constrained environments. These are not a single technique but a mixture of two or more techniques to introduce randomness in the cipher image. Moreover, most of these techniques cannot be implemented easily in hardware or embedded systems. The proposed scheme takes care of the resource constraints, implementation simplicity and speed limitations. It generates high-quality cipher images as validated in the result and analysis discussed later.

BROAD OBSERVATIONS

A cache timing attack is a type of side-channel attack that exploits variations in the time it takes to access data from a cache. Traditional Linear Feedback Shift Register (LFSR) implementations are susceptible to cache timing attacks because the execution time of the algorithm can be influenced by the cache behavior, leading to timing leaks that can reveal secret information.

In a cache timing attack on LFSR, an attacker can monitor the execution time of the algorithm by measuring the time it takes to access cache memory during LFSR operations. By carefully analyzing these timing variations, an attacker can infer information about the internal state of the LFSR, which can ultimately lead to the recovery of secret keys or other sensitive data.

A fault attack is another type of attack that aims to manipulate the behavior of a cryptographic algorithm by injecting faults into the system. Traditional LFSR implementations can be vulnerable to fault attacks because an attacker can introduce intentional faults into the LFSR operations, causing deviations from expected behavior and potentially revealing internal state information.

By manipulating the inputs or the execution of the LFSR, an attacker can induce faulty behavior that may disclose secret keys or weaken the security of the system. Fault attacks on LFSR can be particularly powerful if the attacker has the ability to control the fault injection and observe the resulting faulty outputs.

The SNOW 3G algorithm is a stream cipher used in the 3GPP mobile communications standards, designed to provide secure and efficient encryption. Modified versions of the SNOW 3G algorithm have been developed to address the security vulnerabilities associated with cache timing attacks and fault attacks on traditional LFSR implementations.

To mitigate cache timing attacks, modified SNOW 3G algorithms are designed to minimize the influence of cache behavior on the execution time. This can be achieved through various techniques, such as careful memory access patterns, instruction scheduling, or algorithmic changes. By reducing timing variations introduced by cache accesses, the modified SNOW 3G algorithm makes it harder for an attacker to exploit timing leaks.

Regarding fault attacks, modified SNOW 3G algorithms implement additional fault detection and correction mechanisms. These mechanisms aim to detect and mitigate the effects of injected faults during the LFSR operations. By ensuring the correct behavior of the algorithm even in the presence of faults, the modified SNOW 3G algorithm enhances the resistance against fault attacks and maintains the security of the system.

In summary, modified versions of the SNOW 3G algorithm address the vulnerabilities of traditional LFSR implementations to cache timing attacks and fault attacks. These modifications focus on minimizing timing leaks caused by cache behavior and implementing fault detection and correction mechanisms to maintain the algorithm's security even in the presence of intentional faults.

OBJECTIVES

The objectives of image encryption are as follows:

1. Confidentiality: The primary objective of image encryption is to ensure the confidentiality of the image content. By encrypting the image, unauthorized users are unable to view or understand its original content without the appropriate decryption key. This protects sensitive and private information from unauthorized access.

2. Privacy: Image encryption aims to preserve the privacy of individuals or organizations by preventing unauthorized individuals from viewing or intercepting their images. This is particularly important when transmitting or storing images in vulnerable environments, such as over the internet or on shared servers.

3. Secure transmission: Image encryption enables secure transmission of images over untrusted networks. By encrypting the image data, it prevents unauthorized interception or eavesdropping during transmission, ensuring that only the intended recipient can access and decipher the image.

4. Protection against tampering: Image encryption can help detect and prevent unauthorized modifications or tampering of images. By employing cryptographic techniques, any alteration made to the encrypted image becomes evident upon decryption, ensuring the integrity and authenticity of the image.

5. Compliance with regulations and standards: Many industries and organizations are subject to regulations and standards that require the protection of sensitive image data. Image encryption helps ensure compliance with these requirements, such as data protection laws or industry-specific regulations.

6. Defense against attacks: Image encryption strengthens the security of images against various attacks, such as unauthorized access, data breaches, or image forgery. It provides an additional layer of protection to prevent image-related security vulnerabilities and mitigate potential risks.

The Modified SNOW 3G algorithm is a stream cipher that was initially designed for use in the 3rd Generation Partnership Project (3GPP) mobile communications standards. It offers confidentiality and integrity for data transmission. When applied to image encryption, the Modified SNOW 3G algorithm can be used to provide secure encryption and protect the confidentiality of image data.

WORKING PROCEDURES/ PROPOSED APPROACHES/ IMPLEMENTATION

Working Procedures

A Pseudo Random Number Generator (PRNG)) is a process which produces a sequence of numbers whose properties approximate the properties of sequences of random numbers. Linear Feedback Shift Register (LFSR) is one of the popular PRNG. It is used in Cryptography and Coding theory for its good statistical properties and easy implementation in hardware. As it has a drawback in software implementation, Word-based LFSR is used in modern Word-based processors. σ−LFSR is one kind of Word-based LFSR [36].

Definition 1. We dene a σ−LFSR [37] as a word based LFSR which follows the following recurrence relation,

$$\mathbf{D_{n+b}} = \mathbf{C}_{b-1}D_{n+b-1} + \mathbf{C}_{b-2}D_{n+b-2} + \cdots + \mathbf{C}_0 D_n, \qquad (1)$$

where, each $s_i \in \mathbb{F}_2^m$ and $C_i \in \mathbb{F}_2^{m*m}$. Here, each delay block has m-inputs and m-outputs and the σ-LFSR generates a sequence of vectors in $\mathbb{F}_2^m$. The matrices $C_0, C_1, \cdot\cdot\cdot, C_{b-1}$ are referred to as the gain matrices of the σ-LFSR and the following matrix is defined as its configuration matrix.

where, ze, Id $\in \mathbb{F}_2^{m*m}$ are the all-zero and identity matrices respectively. We shall refer to the structure of this matrix as the M-companion structure. To study how to generate M companion matrix, article [38] explains the algorithm with O (n^4) time complexity. In this article, we have used this paper to generate the configuration matrix of σ−LFSR. The main purpose of σ−LFSR is to introduce a vector-based pseudorandom number generator in our work. In this article, every operation is performed over Galois Field (GF (2)), ⊕ is used for XOR between two integers, << sign is used for left shift operator, || is used for bitwise OR operator, ⊞ is used for addition of two integers modulo 2^{64} , F_2^n signifies n dimensional vector over GF(2), F_2^{n*n} represents a matrix of dimension n over GF(2) and F_2^s Finite fields with 2^8 elements over F_2.

PROPOSED APPROACHES

The Modified SNOW 3G algorithm comprises a σ− Linear Feedback Shift Register (LFSR) and nonlinear Finite State Machine (FSM) used in SNOW 3G. The σ−LFSR has b = 8 delay blocks (D_i |i ∈ [8]) of each is of size m = 64 bit. Figure 1 describes the sketch of the SNOW 3G algorithm.

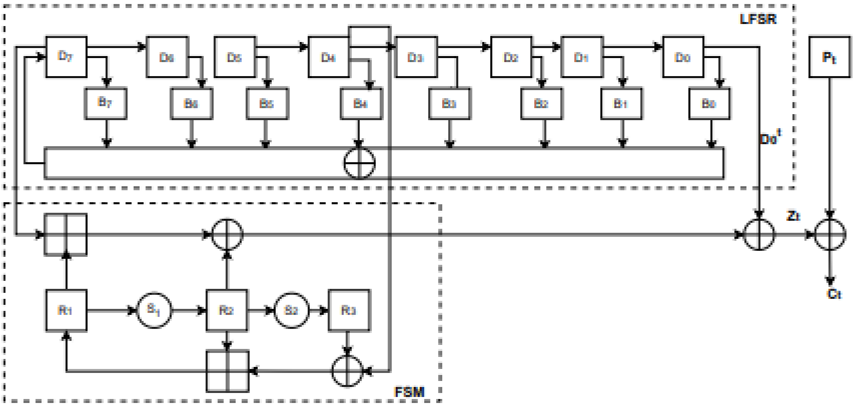


Fig. 1. Modified SNOW 3G

The LFSR part of Figure 1 has a feedback polynomial which has 8 gain matrices ($B_i \in F_2{}^{64\times 64}$). The feedback polynomial $x^{8} + B_7x^7 + B_6x^6 + \cdots + B_1x + B_0$ over matrix ring $M_{64}(F_2)$ is generated from the primitive polynomial $x^{512} + x^{419} + x^{321} + x^{125} + 1$ over F_2. In these contexts, we use the configuration matrix generation algorithm proposed by [38,39] to find the feedback polynomial of the σ−LFSR. Next, we describe the driving equation of σ −LF SR and Finite State Machine (FSM).

$$D_7^{t+1} = \begin{cases} \sum_{i=0}^{7} B_i D_i^t \oplus F^t & \text{if } t <= 32, \\ \sum_{i=0}^{7} B_i D_i^t, & \text{otherwise}. \end{cases} \qquad (3)$$

In the above equation D_i^t is the value i−th delay block at t^{th} timestamp and F^t is output of an FSM at t^{th} time-stamp which is described below. The FSM part of SNOW 3G consists of three registers R_1^t, R_2^t, $R_3^t \in F_2{}^{64}$ and two substitution boxes [6] S_1 and S_2. We use 8 parallel S-BOXes (composed of AES Subbyte and Mixcolumn operation) to compute each S-BOX function. Suppose b_i^t is the one-byte input data at t^{th} timestamp and b_i^{t+1} denotes the output data at t + 1 timestamp. The general structure of the S-BOX equation 4 of S_1, S_2 as follows:

$$\begin{bmatrix} B_0^{t+1} \\ B_1^{t+1} \\ B_2^{t+1} \\ B_3^{t+1} \end{bmatrix} = \begin{bmatrix} U & U+1 & 1 & 1 \\ 1 & U & U+1 & 1 \\ 1 & 1 & U & U+1 \\ U+1 & U & 1 & 1 \end{bmatrix} \begin{bmatrix} S\left(B_0^t\right) \\ S\left(B_1^t\right) \\ S\left(B_2^t\right) \\ S\left(B_3^t\right) \end{bmatrix}. \qquad (4)$$

In equation 4 the matrix is circulant MDS[40] matrix where $U \in F_2{}^{8}$ and generally U = 2 is chosen.

In equation 4, S(a) = 0 when a = 0, and $S(a) = a^{254}$ otherwise, where, $a \in F_2{}^{8}$.

The S-BOX equation can be written in the following:

$$Si(in) = S(in_1)||S(in_2)||S(in_3)||S(in_4)||S(in_5)||S(in_6)||S(in_7)||S(in_8), \qquad (5)$$

where i ∈ {1, 2}, $in \in F_2{}^{64}$and each $in_j \in F_2{}^{8} | j \in [8]$.

The registers R_1, R_2, R_3 in the KSG are updated as follows:

$$R_3^{t+1} = S2(R_2^t). \qquad (6)$$

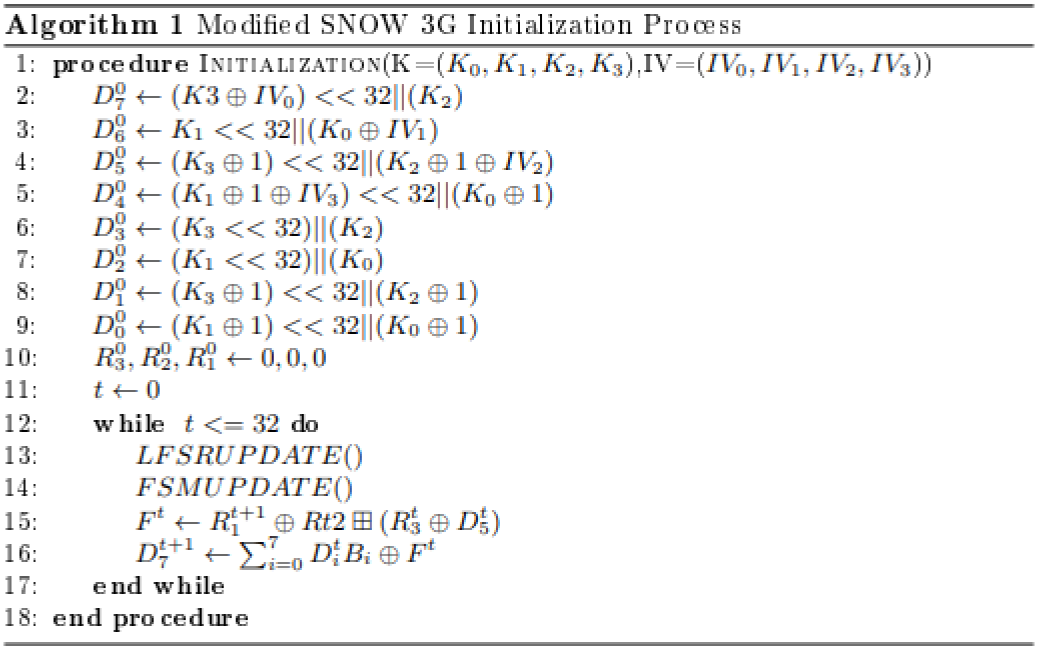
$$R_2^{t+1} = S1(R_1^t). \qquad (7)$$

$$R_1^{t+1} = R_2^t \boxplus (R_3^t \oplus D_5^t). \qquad (8)$$

where $\boxplus$ is integer addition modulo 2^{64} and $\oplus$ is vector wise XOR operation.

IMPLEMENTATION

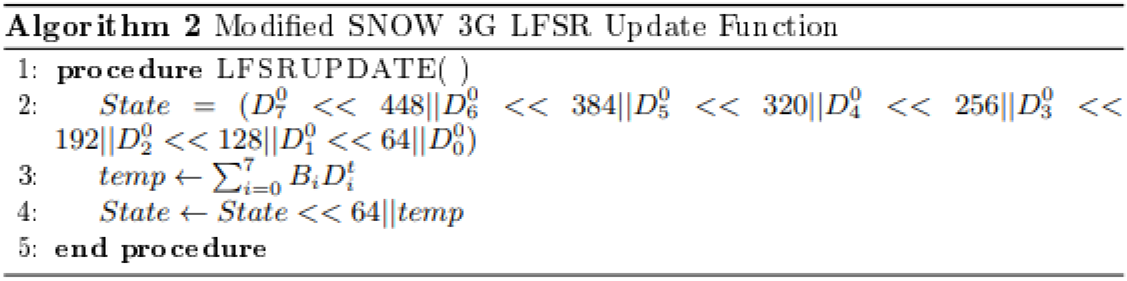
Proposed Algorithm of Modified SNOW 3G for image encryption and decryption: In this section, we explain the various functions of the modified SNOW 3G algorithm and use those algorithms for Image encryption and decryption algorithm. The main algorithm can be broken into three parts such as INITIALIZATION (), LFSRUPDATE () and FSMUPDATE () function. Each of them is described as follows:



Algorithm 1 Modified SNOW 3G Initialization Process

```
procedure INITIALIZATION(K=($K_0, K_1, K_2, K_3$),IV=($IV_0, IV_1, IV_2, IV_3$))
    $D_7^0 \leftarrow (K3 \oplus IV_0) << 32||(K_2)$
    $D_6^0 \leftarrow K_1 << 32||(K_0 \oplus IV_1)$
    $D_5^0 \leftarrow (K_3 \oplus 1) << 32||(K_2 \oplus 1 \oplus IV_2)$
    $D_4^0 \leftarrow (K_1 \oplus 1 \oplus IV_3) << 32||(K_0 \oplus 1)$
    $D_3^0 \leftarrow (K_3 << 32)||(K_2)$
    $D_2^0 \leftarrow (K_1 << 32)||(K_0)$
    $D_1^0 \leftarrow (K_3 \oplus 1) << 32||(K_2 \oplus 1)$
    $D_0^0 \leftarrow (K_1 \oplus 1) << 32||(K_0 \oplus 1)$
    $R_3^0, R_2^0, R_1^0 \leftarrow 0, 0, 0$
    $t \leftarrow 0$
    while $t <= 32$ do
        $LFSRUPDATE()$
        $FSMUPDATE()$
        $F^t \leftarrow R_1^{t+1} \oplus Rt2 \boxplus (R_3^t \oplus D_5^t)$
        $D_7^{t+1} \leftarrow \sum_{i=0}^{7} D_i^t B_i \oplus F^t$
    end while
end procedure
```

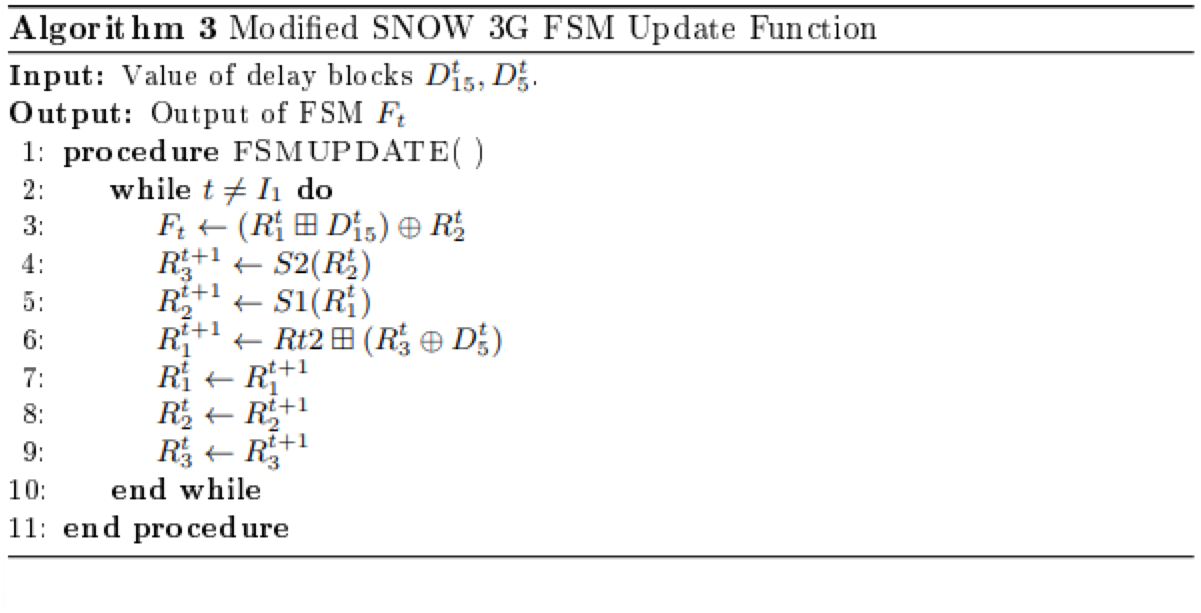
Algorithm 1 encompasses the initialization of the KSG. It takes key $K \in F_2^{128}$ which is broken into 4 sub-keys K_0, K_1, K_2, K_3, each $K_i \in F_2^{32}$. It also takes an initialization vector $IV \in F_2^{128}$, which must be changed after each communication. The IV is also divided into 4 sub-parts (IV_0, IV_1, IV_2, IV_3) of size 32 bit each. Here we initialize the delay blocks of the σ−LFSR by K/IV combination. 1 represents the value $2^{32} - 1$. Keystream runs for 32 clock cycles and is not accessible to the adversary. This procedure adds more entropy to the key than the initial.



Algorithm 2 Modified SNOW 3G LFSR Update Function

```
1: procedure LFSRUPDATE( )
2:     $State = (D_7^0 << 448||D_6^0 << 384||D_5^0 << 320||D_4^0 << 256||D_3^0 << 192||D_2^0 << 128||D_1^0 << 64||D_0^0)$
3:     $temp \leftarrow \sum_{i=0}^{7} B_i D_i^t$
4:     $State \leftarrow State << 64||temp$
5: end procedure
```

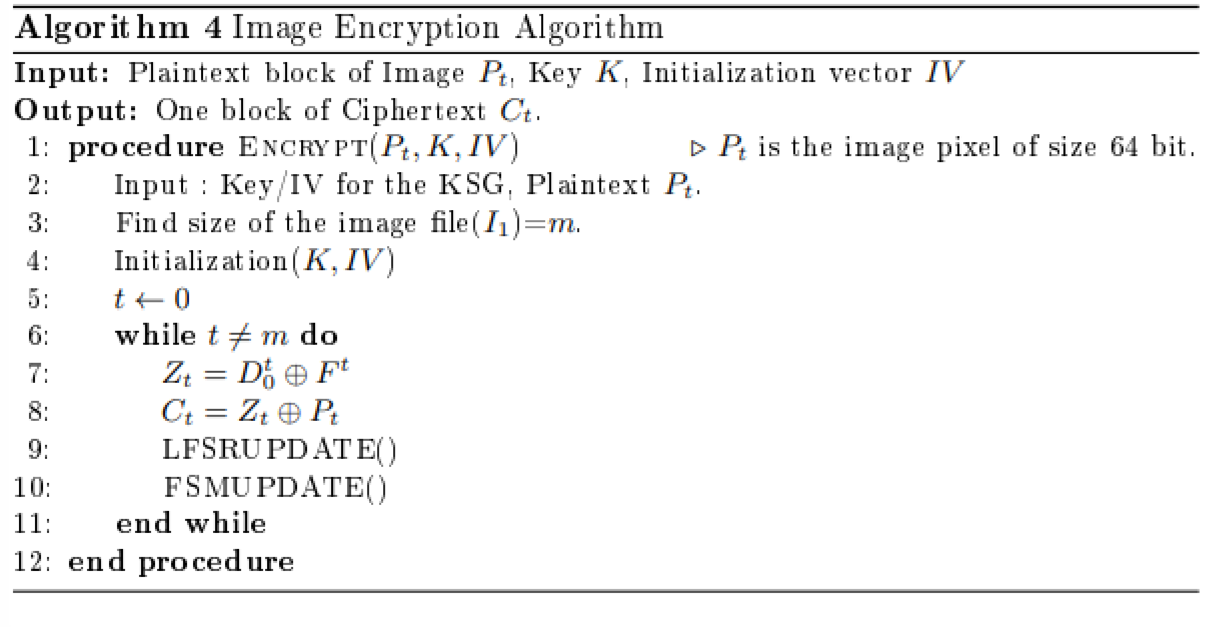
The Algorithm 2 presents the state updation of σ−LFSR with the help of delay blocks $D_i \in F_2^{64}$ and gain matrices $B_i \in F_2^{64*64}$. || and << operators are used as bit-wise OR and AND operators. Line number 2 in the algorithm states that the state of the LFSR (size 512) is formed using the values of delay blocks D_i. Line numbers 3 and 4 explain the 64-bit updation of the state of the LFSR and the 64-bit left shift of the state value of σ−LFSR. The state of the σ−LFSR is updated in each clock pulse if we call the LFSRUPDATE () function of the KSG.



Algorithm 3 Modified SNOW 3G FSM Update Function

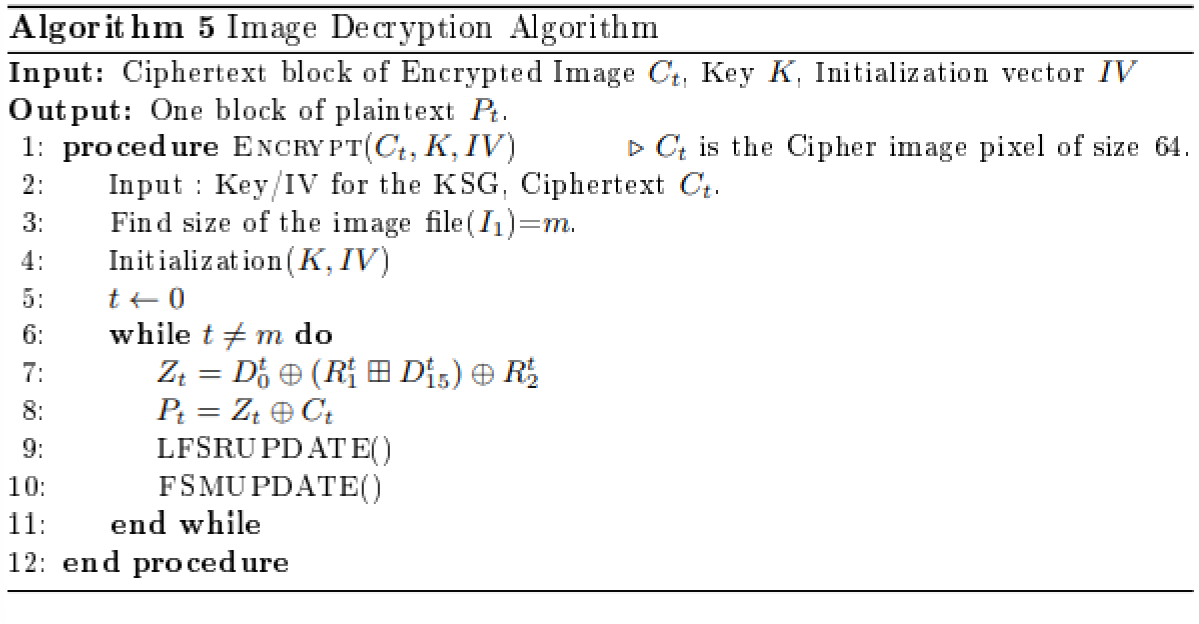
```
Input: Value of delay blocks $D_{15}^t, D_5^t$.
Output: Output of FSM $F_t$
procedure FSMUPDATE( )
    while $t \neq I_1$ do
        $F_t \leftarrow (R_1^t \boxplus D_{15}^t) \oplus R_2^t$
        $R_3^{t+1} \leftarrow S2(R_2^t)$
        $R_2^{t+1} \leftarrow S1(R_1^t)$
        $R_1^{t+1} \leftarrow Rt2 \boxplus (R_3^t \oplus D_5^t)$
        $R_1^t \leftarrow R_1^{t+1}$
        $R_2^t \leftarrow R_2^{t+1}$
        $R_3^t \leftarrow R_3^{t+1}$
    end while
end procedure
```

Algorithm 3 represents the non-linear part of the key stream generator. It uses three nonlinear functions $\boxplus$, nonlinear SBOX S1 and S2(function $F_2^{64} \rightarrow F_2^{64}$) which is used in between three registers R_1, R_2, R_3. How each register Ri $\in F_2^{64}$ for i $\in$ [3] is updated is given in the equations. The Algorithm outputs $F_t \in F_2^{64}$in each clock pulse.



Algorithm 4 Image Encryption Algorithm

```
Input: Plaintext block of Image $P_t$, Key $K$, Initialization vector $IV$
Output: One block of Ciphertext $C_t$.
procedure ENCRYPT($P_t, K, IV$)                ▷ $P_t$ is the image pixel of size 64 bit.
    Input : Key/IV for the KSG, Plaintext $P_t$.
    Find size of the image file($I_1$)=$m$.
    Initialization($K, IV$)
    $t \leftarrow 0$
    while $t \neq m$ do
        $Z_t = D_0^t \oplus F^t$
        $C_t = Z_t \oplus P_t$
        LFSRUPDATE()
        FSMUPDATE()
    end while
end procedure
```



Algorithm 5 Image Decryption Algorithm

Input: Ciphertext block of Encrypted Image C_t, Key K, Initialization vector IV

Output: One block of plaintext P_t.

```
procedure ENCRYPT($C_t, K, IV$)            ▷ $C_t$ is the Cipher image pixel of size 64.
    Input : Key/IV for the KSG, Ciphertext $C_t$.
    Find size of the image file($I_1$)=$m$.
    Initialization($K, IV$)
    $t \leftarrow 0$
    while $t \neq m$ do
        $Z_t = D_0^t \oplus (R_1^t \boxplus D_{15}^t) \oplus R_2^t$
        $P_t = Z_t \oplus C_t$
        LFSRUPDATE()
        FSMUPDATE()
    end while
end procedure
```

Algorithms 4 and Algorithm 5 are encryption and decryption algorithms that use modified SNOW 3G. To run the Algorithm, we need to break the image file into 64-bit plaintext. Besides, the key K and the Initialization vector IV are also used to run the keystream. Here, Z_t is the key generated from the KSG, and it is exored with the plaintext and the ciphertext as in Algorithms 4 and 5.

RESULT EVALUATION /DISCUSSION

On an Intel (R) Core (TM) i5-3230M 2.60 GHz CPU, 4 GB RAM, and WINDOWS 8.1 pro operating system, the proposed scheme was tested. The comparison with known algorithms such as AES [41], DES [42], 3-DES [43], and the scheme proposed by Babaei et al. [29] is carried out using Python language. The original images used were Lena, Baboon and Cameraman. The original images and the corresponding cipher images are shown in Figure 2 and Figure 3, respectively.

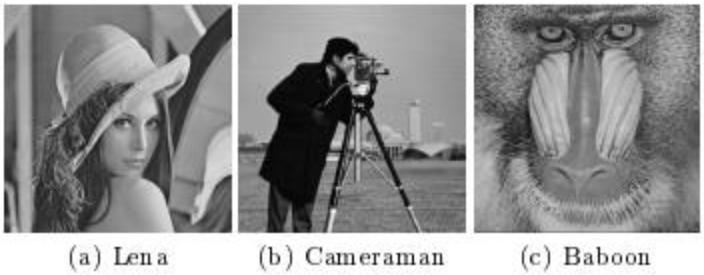

(a) Lena (b) Cameraman (c) Baboon

Fig 2. Original images

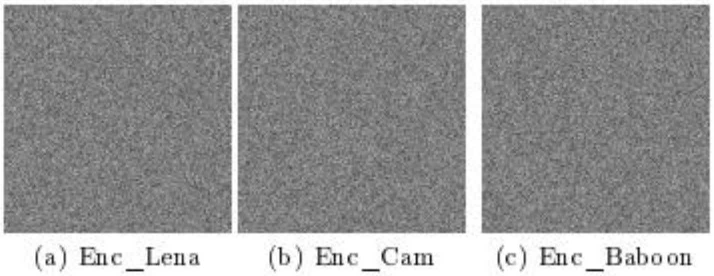
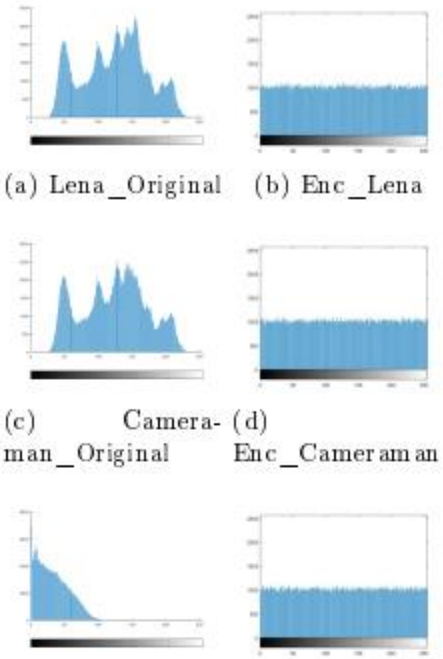

(a) Enc_Lena (b) Enc_Cam (c) Enc_Baboon

Fig 3. Encrypted Images.

RESULT EVALUATION:

In this section, a detailed discussion is presented regarding various performance analysis of the proposed model.



(a) Lena_Original (b) Enc_Lena

(c) Cameraman_Original (d) Enc_Cameraman

(e) Baboon_Original (f) Enc_Baboon

Fig 4. Original and encrypted images' histogram plots

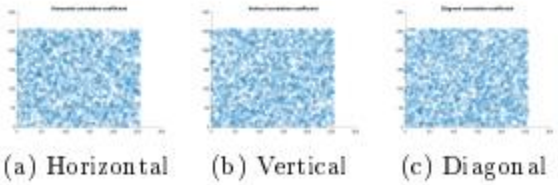
Analysis of Histogram: A digital image's histogram is an important feature. It is a graphical depiction of an image's tonal distribution. It also represents the number of pixel values for each tone-value distribution. The histogram's atter and uniform distribution indicate that the created cipher-image has more randomness. The histogram plots of various plain images are shown in Figures 4(a) to 4(e). Figures 4(b) to 4(f) show histogram graphs of corresponding encrypted photos in the same way. The distribution of the cipher pictures is relatively uniform, as seen from the histogram. It adequately justifies the suggested scheme's security. Correlation Coefficient Analysis The correlation coefficient α is used to check how two neighboring pixels in an image are related. If both pixels are part of an image, the value of α tends to 1 and 0 if both pixels belong to a random image. The correlation coefficient α is calculated using Equation (9).

$$\alpha = \frac{\sum_a \sum_b (X_{ab} - \bar{X})(Y_{ab} - \bar{Y})}{\sqrt{(\sum_a \sum_b (X_{ab} - \bar{X})^2)(\sum_a \sum_b (Y_{ab} - \bar{Y})^2)}}. \qquad (9)$$

X¯ and Y¯ denote the mean of all pixel values in the image. X_{ab} stands for the pixel value in the a th row and b th column. The horizontal, vertical, and diagonal correlation coefficients have all been examined. Table 1 displays the exact values in each instance for both plain and encrypted images. Figure 5 displays the correlation coefficient analysis plots for the images of Lena, Cameraman, and Baboon, respectively. The presence of a significant number of random bits in cipher-image plots demonstrates the strength of the proposed method.

Image	Type	Horizontal	Vertical	Diagonal
Lena	Original	0.973321`	0.986331	0.966332
Lena	Encrypted	0.012859	0.040757	-0.013456
Cameraman	Original	0.936282	0.919205	0.865661
Cameraman	Encrypted	0.011704	0.023504	-0.041105
Baboon	Original	0.878303	0.758934	0.732143
Baboon	Encrypted	-0.004029	0.015977	0.009952

Table 1. Values of Correlation Coefficients for different Images



(a) Horizontal (b) Vertical (c) Diagonal

Fig 5. Correlation coefficient plot of Encrypted Cameraman in horizontal, vertical and diagonal spectrum.

Information Entropy Analysis: Information entropy is an important measure of unpredictability in cipher images [48]. Equation (10) can be used to calculate the information entropy, ε.

$$\varepsilon = \sum_{i=0}^{ab} R(G_i)\log_2 \frac{1}{R(G_i)}. \qquad (10)$$

The image's histogram count is represented by R(Gi). The dimension of the image matrix is a × b. Table 2 shows the detailed test findings. Table 2 shows that the achieved entropy values are quite closer to the theoretical entropy values [49,50]. It further signifies that the proposed scheme can resist the majority of the existing security attacks.

Image Name	Type of Image	Entropy
Lena	Original	7.4550
	Encrypted	7.9992
Cameraman	Original	7.4279
	Encrypted	7.9993
Baboon	Original	7.5394
	Encrypted	7.9992

Encrypted Image Quality Analysis: In this work, two conventional error metrics, Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR), were used to verify the quality of the encrypted image. To create high uncertainty in a cipher image, a high PSNR score (> 30) is required [51,52,53,54]. The MSE, Mp and PSNR, Pp measurements are calculated using Equations (11) and (12), respectively. The plain image and cipher image, respectively, are Po and Pe. The image's row and column are represented by D1 and D2. The picture matrix's maximum supported pixel value is U. Good encryption quality is indicated by high PSNR and low MSE values. A comparison was made with the encryption algorithms of [55] and [56]. Table 3 depicts the outcome.

Table 3. Difference of MSE and PSNR of Cipher Image with Existing Schemes

Image	Lena		Baboon	
Scheme/Metrics	MSE	PSNR	MSE	PSNR
Proposed Scheme	90.92	29.86	72.89	29.98
Ref [55]	90.73	28.58	72.73	29.54
Ref [56]	90.85	28.58	72.78	29.54

$$M_p = \frac{\sum_{D_1,D_2}[P_o(d_1,d_2) - P_e(d_1,d_2)]^2}{D_1 \times D_2}. \qquad (11)$$

$$P_p = 10\ log_{10}\left(\frac{U^2}{M_p}\right). \qquad (12)$$

Differential Analysis Number of Pixel Change Rate (NPCR) and United Averaged Changing Intensity (UACI) are used in the differential analysis (UACI). These two numbers indicate an encryption algorithm's resilience to differential cryptanalysis assaults. The first time NPCR and UACI were used was in [57]. They have been commonly utilized for image encryption methods

since then. When only one pixel in a simple image is modified, NPCR denotes the rate of change of the number of pixels. UACI, on the other hand, is a metric for determining the average intensity of differences between the plain and encrypted images [58].

Assume X_1 is the encrypted image before the one-pixel change in the plain picture, and X_2 is the cipher image after the one-pixel change. $X_1(a, b)$ and $X_2(a, b)$ are the pixel values of X_1 and X_2 in the image matrix P, described by Equation (13). Equations (14) and (15) are used to calculate NPCR (Np) and UACI (Up) values. D_1 and D_2 are the image's width and height, respectively, while U is the cipher image's maximum supported pixel value. It will be 255 in the case of a gray-scale image.

$$P(a,b) = \begin{cases} 0 & \text{if } X_1(a,b) = X_2(a,b), \\ 1 & \text{if } X_1(a,b) \neq X_2(a,b). \end{cases} \qquad (13)$$

$$N_p(X_1,X_2) = \sum \frac{P(a,b)}{D_1 \times D_2} \times 100\%. \qquad (14)$$

$$U_p(X_1,X_2) = \sum_{a,b} \frac{|X_1(a,b) - X_2(a,b)|}{U \times D_1 \times D_2} \times 100\%. \qquad (15)$$

NPCR and UACI should be close to 99.61 per cent and 33.44 percent, respectively [58]. Table 4 shows the

NPCR and UACI results in greater detail. The obtained NPCR and UACI are more than the threshold values specified in Table 4. As a result, the suggested method can be claimed to be robust to differential cryptanalysis attacks.

Table 4. Result of Differential Analysis

Test/Cipher Image	Lena	Cameraman	Baboon
NPCR	99.78634	99.71278	99.91445
UACI	33.85143	33.78347	33.98965

NIST Randomness Tests Various NIST [59]-recommended randomness tests were carried out to ensure that the suggested system is random. The test results for Lena's cipher image are provided in Table 5. NIST tests were also carried out on the Cameraman and Baboon images. Each test is accompanied by a reference distribution and a decision rule for being non-random[1]. Block Frequency Test, for example, follows the χ^2 distribution. The sequence is termed non-random; if the p-value is less than 0.01, it is random. Table 5 shows that all p-values are more significant than the decision threshold value. As a result, the suggested scheme generates a random cipher image

Table 5. Result of NIST randomness tests

Test Name	P-Value	Result
Longest Run of 1's	0.198533	✓
Frequency	0.88342	✓
Block Frequency	0.194323	✓
The Binary Matrix Rank	0.182659	✓
Cumulative Sum	0.976602	✓
Runs	0.195945	✓
Non-Overlapping Template Matching	0.990263	✓
Discrete Fourier (Spectral)	0.049341	✓
Maurer's Universal Statistical	0.992466	✓
Linear Complexity	0.120487	✓
Serial	0.130323	✓
Approximate Entropy	0.988676	✓

Runtime Comparison This section compares the runtime of encryption of ?? with some of the existing and popularly used ciphers. The ciphers used for comparison are AES, DES, and 3-DES. Furthermore, the scheme has been compared with the scheme of Babaei et al. [29], represented as Ref. Tech. in the figure. The result of runtime comparison is shown in Figure Ref fig: runtime.

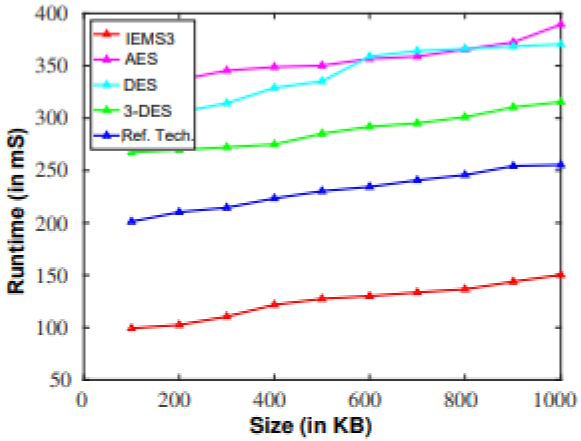


Fig 6. Runtime Analysis

DISCUSSION

The resultant graph shows that the proposed scheme, modified SNOW 3G algorithm, shows a better result among all the techniques considered for comparison. Subsequently, the proposed scheme is an ideal fit for any real-time application.

CONCLUSION

This paper provides a word-oriented stream cipher for image encryption based on the $\sigma-$LFSR with a modified block size. This system generates cipher-images as random images, negating the effects of assaults on the Fog nodes' insecure communication route between the perception layer and the network layer. The proposed approach can be used in critical real-world settings in which raw data is crucial. For instance, in healthcare, military, and biological image communication, to name a few. The scheme ensures the delivery of raw data from the sensors to the recipient in all circumstances. Any adverse attack would be life-threatening; therefore, communication must be quick and encrypted. The proposed system may be helpful in such urgent situations. All of the statistical randomness tests in the NIST test suite were passed by the cipher-images created by the proposed system. The scheme's robustness is also confirmed by standard analyses such as the differential, correlation coefficient, entropy, and image quality. It ensures that an opponent attempting cryptanalysis cannot understand the raw data. The proposed approach also has a low time complexity, which will help to speed up communication. It can be parallelized in the future, basically the matrix-vector multiplication, resulting in a substantially shorter runtime. A cache-efficient algorithm with a 128 bit keystream will be more efficient and help extend the sensor's battery life by reducing the number of matrix-vector multiplication.

REFERENCES

[1] S. Roy, U. Rawat, H. A. Sareen, and S. K. Nayak, IECA: an efficient IoT friendly image encryption technique using programmable cellular automata, Journal of Ambient Intelligence and Humanized Computing, vol. 11, no. 11, pp. 50835102, 2020.

[2] S. Roy, U. Rawat, and J. Karjee, A lightweight cellular automata based encryption technique for iot applications, IEEE Access, vol. 7, pp. 3978239793, 2019.

[3] P. Ekdahl and T. Johansson, A new version of the stream cipher snow, in International Workshop on Selected Areas in Cryptography, pp. 4761, Springer, 2002.

[4] P. Ekdahl, T. Johansson, A. Maximov, and J. Yang, A new snow stream cipher called snow-v, IACR Transactions on Symmetric Cryptology, pp. 142, 2019.

[5] P. Kitsos, N. Sklavos, and A. N. Skodras, An fpga implementation of the zuc-stream cipher, in 2011 14th Euromicro Conference on Digital System Design, pp. 814817, IEEE, 2011.

[6] G. Orhanou, S. El Hajji, and Y. Bentaleb, Snow 3g stream cipher operation and complexity study, Contemporary Engineering Sciences-Hikari Ltd, vol. 3, no. 3, pp. 97111, 2010.

[7] S. Krishnaswamy and H. K. Pillai, On the number of linear feedback shift registers with a special structure, IEEE transactions on information theory, vol. 58, no. 3, pp. 17831790, 2011.
[8] S. Krishnaswamy and H. K. Pillai, On multi sequences and their extensions, arXiv preprint arXiv:1208.4501, 2012.

[9] M. Kaur and V. Kumar, A comprehensive review on image encryption techniques, Archives of Computational Methods in Engineering, vol. 27, no. 1, pp. 1543, 2020. 16 No Author Given
[10] A. Kumar and M. Dua, Novel pseudo random key & cosine transformed chaotic maps-based satellite image encryption, Multimedia Tools and Applications, pp. 1 21, 2021.

[11] V. V. Tarasova and V. E. Tarasov, Logistic map with memory from economic model, Chaos, Solitons & Fractals, vol. 95, pp. 8491, 2017.

[12] S. Phatak and S. S. Rao, Logistic map: A possible random-number generator, Physical review E, vol. 51, no. 4, p. 3670, 1995.

[13] S. A. Parah, N. A. Loan, A. A. Shah, J. A. Sheikh, and G. Bhat, A new secure and robust watermarking technique based on logistic map and modification of dc coefficient, Nonlinear Dynamics, vol. 93, no. 4, pp. 19331951, 2018.

[14] Z. Hua, Y. Zhou, and H. Huang, Cosine-transform-based chaotic system for image encryption, Information Sciences, vol. 480, pp. 403419, 2019.

[15] S. Sabir and V. Guleria, “Multi-layer color image encryption using random matrix a-ne cipher, rp2dfrht and 2d arnold map,” Multimedia Tools and Applications, pp. 1-25, 2021.

[16] M. Roy, S. Chakraborty, and K. Mali, “The msk: a simple and robust image encryption method”, Multimedia Tools and Applications, pp. 1-31, 2021.

[17] A. Ur Rehman, X. Liao, and H. Wang, “An innovative technique for image encryption using tri-partite graph and chaotic maps”, Multimedia Tools and Applications, pp. 1-27, 2021.

[18] A. Mansouri and X. Wang, “A novel one-dimensional chaotic map generator and its application in a new index representation-based image encryption scheme,” Information Sciences, vol. 563, pp. 91-110, 2021.

[19] K. Jithin and S. Sankar, “Colour image encryption algorithm combining arnold map, dna sequence operation, and a mandelbrot set,” Journal of Information Security and Applications, vol. 50, p. 102428, 2020.

[20] R. Wang, G.-Q. Deng, and X.-F. Duan, “An image encryption scheme based on double chaotic cyclic shift and josephus problem,” Journal of Information Security and Applications, vol. 58, p. 102699, 2021.

[21] A. M. Odlyzko and H. S. Wilf, “Functional iteration and the josephus problem,” Glasgow Mathematical Journal, vol. 33, no. 2, pp. 235-240, 1991.

[22] S. Jahangir and T. Shah, “A novel multiple color image encryption scheme based on algebra m (2, f2 [u]/< u8>) and chaotic map,” Journal of Information Security and Applications, vol. 59, p. 102831, 2021.

[23] L. Liu, D. Jiang, X. Wang, X. Rong, and R. Zhang, “2d logistic-adjusted-chebyshev map for visual color image encryption,” Journal of Information Security and Applications, vol. 60, p. 102854, 2021.

24. J. S. Khan and S. K. Kayhan, “Chaos and compressive sensing based novel image encryption scheme,” Journal of Information Security and Applications, vol. 58, p. 102711, 2021.

[25] A. Sahasrabuddhe and D. S. Laiphrakpam, “Multiple images encryption based on 3d scrambling and hyper-chaotic system,” Information Sciences, vol. 550, pp. 252- 267, 2021.

[26] L. Lidong, D. Jiang, X. Wang, L. Zhang, and X. Rong, “A dynamic triple-image encryption scheme based on chaos, s-box and image compressing,” IEEE Access, vol. 8, pp. 210382-210399, 2020.

[27] Y. Ding, G. Wu, D. Chen, N. Zhang, L. Gong, M. Cao, and Z. Qin, “Deepedn: a deep learning-based image encryption and decryption network for internet of medical things,” IEEE Internet of Things Journal, 2020.

[28] M. A. Malik, Z. Bashir, N. Iqbal, and M. A. Imtiaz, “Color image encryption algorithm based on hyper-chaos and dna computing,” IEEE Access, vol. 8, pp. 88093- 88107, 2020.

[29] M. Babaei, “A novel text and image encryption method based on chaos theory and DNA computing” Natural computing, vol. 12, no. 1, pp. 101-107, 2013.

[30] Y. Zhang, “The image encryption algorithm based on chaos and dna computing,” Multimedia Tools and Applications, vol. 77, no. 16, pp. 21589-21615, 2018.

[31] W. El-Shafai, F. Khallaf, E.-S. M. El-Rabaie, and F. E. A. El-Samie, “Robust medical image encryption based on dna-chaos cryptosystem for secure telemedicine and healthcare applications,” Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 10, pp. 9007-9035, 2021.

[32] Y. Ding, F. Tan, Z. Qin, M. Cao, K.-K. R. Choo, and Z. Qin, "Deepkeygen: A deep learning-based stream cipher generator for medical image encryption and decryption," IEEE Transactions on Neural Networks and Learning Systems, 2021.

[33] W. Sirichotedumrong, Y. Kinoshita, and H. Kiya, "Pixel-based image encryption without key management for privacy-preserving deep neural networks," IEEE Access, vol. 7, pp. 177844-177855, 2019.

[34] R. Enayatifar, A. H. Abdullah, and I. F. Isnin, "Chaos-based image encryption using a hybrid genetic algorithm and a dna sequence," Optics and Lasers in Engineering, vol. 56, pp. 83-93, 2014.

[35] A. H. Abdullah, R. Enayatifar, and M. Lee, "A hybrid genetic algorithm and chaotic function model for image encryption," AEU-International Journal of Electronics and Communications, vol. 66, no. 10, pp. 806-816, 2012.