Efficient and secure cipher scheme for multimedia contents

Hassan Noura, Mohamad Noura, Ali Chehab, Mohammad Mansour, Raphael Couturier

Efficient and Secure Cipher Scheme for Multimedia Contents

Hassan N. Noura[1], Mohamad Noura[2], Ali Chehab[1], Mohammad M. Mansour[1], and Raphaël Couturier*[2]

[1]American University of Beirut, Department of Electrical and Computer Engineering, Beirut, Lebanon, emails: {hn49, chehab, mm14}@aub.edu.lb
[2]Univ. Bourgogne Franche-Comté (UBFC), CNRS, FEMTO-ST Institute, France, emails: {mohamad.noura, raphael.couturier}@univ-fcomte.fr,
*corresponding author

November 23, 2018

Abstract

The impact of confidentiality and privacy breaches are more pronounced when dealing with multimedia contents. One of the obvious techniques to counter these threats is the use of encryption. A number of algorithms for robust image encryption, targeted for real-time applications with tight resource constraints, has been proposed in the literature. In this paper, first, we analyze two recent cipher schemes for image contents, which are based on two rounds. We show that the schemes are designed to ensure maximum avalanche effect in the whole image by employing the chaining block code mode (CBC) in forward and backward directions. However, they do not lend themselves to parallel implementation and they have a problem with error propagation, which is not desirable for wireless multimedia transmission. As such, we propose to redesign the underlying algorithm to make it practical when used with applications that either suffer from a high error percentage or from real-time constraints. The modified cipher employs the counter mode to eliminate the chaining process (forward and backward), which allows for parallel computations and minimizes the effect of error propagation. According to the security and performance results, the proposed scheme can respond better to the applications and/or system requirements and limitations by ensuring a better performance and an equally high level of security compared to both ciphers in addition to minimum error propagation. To the best of our knowledge, the proposed scheme is the first dynamic key-dependent stream cipher scheme with a pseudo-random key-stream generation for re-ordering of sub-matrices.

Keywords: Parallel image cipher scheme, Error Propagation, Visual degradation, Balance between security level and performance.

1 Introduction

The exponential growth of advanced technologies (processing, Telecom, networking, etc.) led to the emergence of new multimedia applications. Such applications rely (among others) on the widespread usage of image-capture sensors in hand-held devices (e.g., camera glasses, smart phones, tablets, etc.), giving the users the ability to capture, store and send multimedia data. This resulted in a huge and voluminous production of images that are daily exchanged/shared among billions of users through social networks (e.g., Instagram, Facebook, Twitter) or simply stored on web/cloud platforms (e.g., Apple iCloud).

Hence, there is an urgent need to *secure the contents of users' images*, especially after the recent celebrity photo hack[1], where more than 100 individual iCloud celebrities accounts have been hacked, making their sensitive private pictures (mostly naked pictures taken by their smart phones) public. Also, similar situations can also occur for ordinary people when their phones are stolen or hacked. This could be very detrimental for those individuals, in particular in conservative countries.

The security requirements are not limited to general public applications; several professional applications have the same requirements such as military and medical ones, where sensitive images need to be protected [1, 2, 3, 4]. As such, image confidentiality becomes important and crucial since there is always the risk of information interception. This is achieved by using an encryption algorithm that keeps the content information protected during transmission or in storage on an insecure medium.

Image encryption algorithms are based on either block or stream cipher. Stream cipher consists of producing a key-stream based on a specific key and an initial vector. The produced key-stream is mixed with plain image to produce the cipher image at the source side, and mixed with the cipher image to produce the decrypted image at the receiver side. On the other hand, block cipher takes a block of the plain-image and the key as inputs and applies a round function for several rounds. Each block is encrypted based on a cryptographic operation mode such as CBC [5]. In addition, image encryption schemes can be realized at the pixel level or at the level of the compressed bit-stream.

1.1 Related work

One major dilemma is the trade-off between efficiency and the security level; it is essential to design an efficient cipher scheme that can strike a good balance between efficiency and robustness. Moreover, a low error propagation is as essential, especially if the cipher is to be used for multimedia wireless communication, which suffers from a high error rate compared to wired communication [6].

The traditional symmetric block cipher algorithms such as 3DES [7] and AES [8] are used currently for encrypting multimedia data [9]. However, traditional ciphers use a static structure: the substitution and diffusion primitives are fixed and independent of the static secret key. Therefore, a higher number of round r is required to reach the desired security level. This introduces consequently an overhead in terms of latency and resources that can be high for tiny devices, for example, or for modern high data rate real-time applications. Therefore, the traditional symmetric cipher schemes are acceptable for powerful devices or for the current data rates. However, these may suffer in the case of tiny devices and/or modern real-time delivery of multimedia streams [10]. This motivated cryptography researchers to target the design of alternative lightweight cipher candidates. Moreover, the static cipher structure gives opportunities to potential attacks such as side channel attacks or fault attacks [11], since the substitution and diffusion primitives are static and known to the attacker. Towards preventing these kinds of attacks, several countermeasures have been presented, but these are associated with major overhead.

From more than two decades, a new class of cryptography was presented, based on the Chaos theory [12], and relying on chaotic maps that can be linear (cat map) or non-linear (logistic,

[1] Celebrity photo hack wikipedia page,
http://en.wikipedia.org/wiki/2014_celebrity_photo_hack

skew tent and PWLCM) [13]. These chaotic maps are pseudo-random and they are extremely sensitive to initial conditions and parameters. In fact, the advantage of chaos theory, compared to traditional cryptographic algorithms, is that it offers a key-dependent structure where substitution and diffusion primitives change according to initial conditions and control parameters of these chaotic maps.

However, the majority of existing chaos-based encryption algorithms suffers from security and performance limitations [14, 15, 16, 17] such as finite computing precision that stems from the periodicity of mapping [18], which is not high for the 1-D chaotic maps. Consequently, this renders the corresponding cryptographic algorithms vulnerable to various kinds of attacks [19, 20]. Another main performance disadvantage is due to the fact that most chaos-based encryption algorithms rely on floating-point arithmetic, which significantly increases the computational complexity (and consequently resources) of any software or hardware implementation and thus, renders practical chaos-based cryptographic solutions inefficient when compared to optimized AES implementation. Note that the standard cryptographic algorithms are based on integer operations [21].

On the other hand, and for more than a decade, several AES optimization techniques were proposed to make it more suitable to support higher data rates, or to be implemented efficiently within tiny devices. The main objective of these optimization techniques is to reduce the required resources and latency and to render its hardware implementation a simple one [10]. Also, a set of optimized assembly AES instructions were introduced to the instruction set of Intel processors [22, 23]. However, AES may not satisfy the future requirements for lightweight cipher schemes according to NIST [24]. This motivated several initiatives targeting the design of new lightweight cryptographic algorithms by simplifying the round function and/or reducing the number of rounds [10]. Examples of such algorithms include LEA [25], FeW [26], Prince [27], TWINE [28], Lblock [29], Piccolo [30], and LED [31]. These ciphers, however, are also based on static substitution and diffusion primitives that require iterating a round function for a large number of rounds, r, which is mandatory to reach the desired security level. Unfortunately, these ciphers can provide the required security level, yet at the expense of high overhead in terms of latency and required computational resources.

Consequently, using such ciphers within limited systems might introduce an overhead that would hinder their proper operation by impacting the overall system performance [24]. As such, designing a new cipher scheme that achieves a high security level with low computational complexity and required resources becomes more than mandatory towards overcoming the limitations for a set of modern systems, applications and devices. In fact, the minimum required number of iterations for recent lightweight cryptographic algorithms is 4 according, which is the case of Hummingbird2 cipher. On the other hand, the recent lightweight chaotic image encryption algorithms such as [32, 33, 34, 35, 36] are also based on a multi-round structure.

The major challenge that limits the majority of chaotic cipher schemes is their use of floating-point calculations and the need to convert operations to integer, which introduces an important overhead in terms of latency and required resources.

Recently, two new lightweight cryptographic solutions were proposed, which are either based on a lightweight round function or on a small number of rounds by relying on the dynamic key approach [37, 38], in order to reduce the required resources and latency [39, 40, 41]. These two recent dynamic key-dependent lightweight cipher schemes use a round function for two iterations ($r = 2$). They apply the Chaining Block Code (CBC) operation mode in the forward and backward directions. The round function of [37] consists of a dynamic substitution operation that uses a dynamic S-box followed by an integer matrix mixing for the diffusion operation. The cipher scheme of [37] is illustrated in Figure 1 and the architecture of [38] is similar compared to [37]. On the other hand, the cipher in [38], achieves a lower latency when compared to [37] since each round iteration applies only a single operation. In fact, the presented cipher in [38] applies the diffusion operation in the first round, the substitution operation in the second round and it uses binary diffusion operations instead of integer ones. However, by analyzing these schemes, several limitations arise such as 1) the effect of error propagation which is doubled, and 2) the inability of parallel computations.

Next, we detail the limitations of [37, 38]:

1. **Error propagation overhead**: using the CBC mode of operation in forward and backward directions leads to error propagation in the corresponding block in addition to the neighboring blocks (previous and next). Moreover, the contents of the corresponding and next neighbor blocks are randomized while specific error(s) occur in the previous block. Therefore, we need to define an efficient cipher scheme that limits the error propagation only to specific byte(s) and more specifically, to be constrained to the same bit-error position(s). Note that the worst error case is when errors are introduced with a uniform distribution, which destroys the whole image contents.

2. **No Parallelism** : The use of the CBC mode in forward and backward directions limits the ability for parallelism. In order to solve this limitation, the encryption algorithm of [37] is redesigned to be realized in counter mode, which reduces the latency according to the obtained results [42].

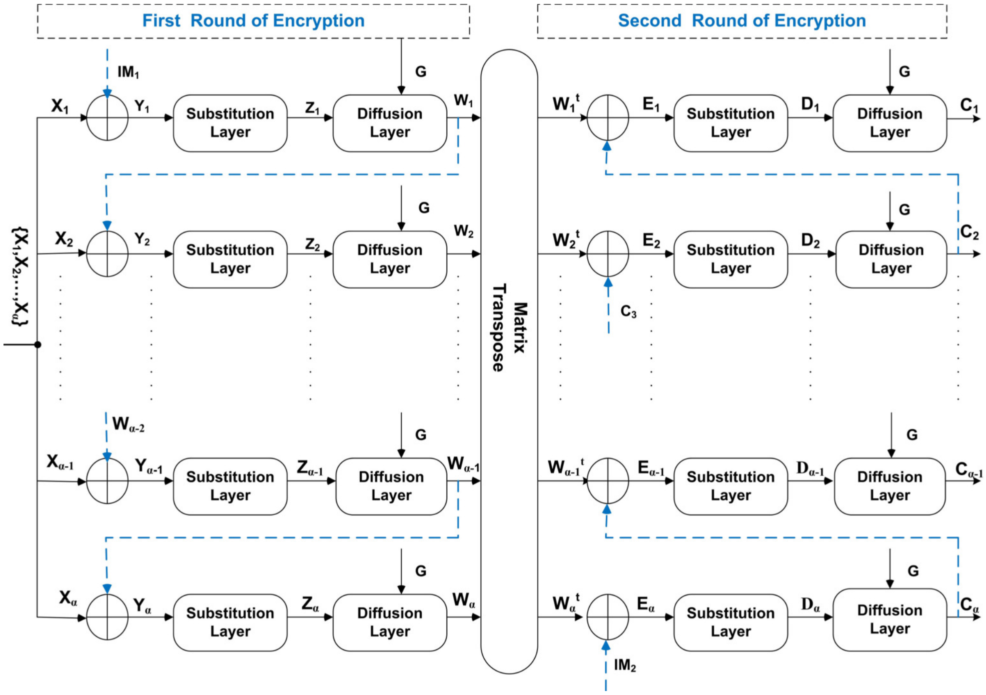


Figure 1: Architecture of the encryption algorithm presented in [37]

1.2 Motivation & Contributions

After highlighting the limitations of the current solutions for recent applications, it becomes evident that there is a need for a new efficient lightweight image cipher scheme. Such cipher has to ensure a high level of security with a low computation complexity, error propagation, simple hardware implementation and parallel computing.

In this paper, we follow the recent work of [37, 38] towards designing new efficient and secure dynamic key-dependent stream cipher, which is based on a key-stream generation algorithm and key-stream sub-matrix permutation. Consequently, the proposed stream cipher is different compared to existing stream ciphers since it uses dynamic substitution and diffusion primitives that change for each new input message. In fact, to the best of our knowledge, the proposed solution is the first dynamic key-dependent stream cipher algorithm with dynamic substitution and diffusion primitives and pseudo-order key-stream generation.

Additionally, the technical contribution of this paper compared to [37, 38] are listed in the following:

- The proposed solution is a stream cipher scheme, while the referenced works are block cipher.
- An update on the simple dynamic key generation is proposed to achieve a high level of security with low overhead. It is based on the variable session key and a Nonce, which changes for each new input message.
- We use the counter mode instead of the chaining mode to enable parallel computations without reducing the cryptographic strength. Equally important, the proposed scheme reduces the effect of error propagation, which consequently ensures lower visual degradation. However, block cipher with ECB, CBC, and OFB operations modes cannot ensure low error propagation. Note that CFB has the worst error propagation effect.
- We introduce a permutation process for the encrypted sub-matrices to achieve a nonlinear (pseudo-random) order of key-stream generation. Such modifications provide better resistance against attacks with low overhead in terms of initialization.
- We use the same technique to generate the substitution table and diffusion matrix as in [38], which is simpler compared to the one in [37]. In addition, the binary diffusion matrix defined in [38] is used in the proposed stream cipher to simplify the hardware implementation and achieve a better performance.

In addition, the proposed key stream generation algorithm is based on a simple flexible key-dependent round function that needs to be iterated for just 2 rounds. This round function consists of two operations, substitution and diffusion. The selected techniques to construct the substitution and diffusion primitives in a dynamic manner guarantee the desired cryptographic strength; the generated key-stream has a high level of randomness and periodicity according to [38].

On the other hand, we conduct tests to assess the performance and security of the proposed solution compared to the recent works [37, 38]. We present a cryptanalysis discussion to verify the immunity of the proposed cipher solution against confidentiality attacks. Note that the existing cryptanalysis techniques target cipher schemes that are based on the use of a static key, and this clearly indicates why the proposed scheme can resist the traditional attacks. Furthermore, this work opens the door for a new kind of cryptanalysis. Even if new modern kind of attacks will be presented in the future for dynamic cipher schemes, the proposed solution is designed with the goal to resist them. The more dynamic is the cipher design, the higher is its security level, which is our main target. Simulation results for a set of security tests such as randomness, uniformity, and sensitivity are introduced in Table 3 to assess better the security of the proposed scheme compared to the recent works of [37, 38]. The obtained results are close to the one obtained in [37, 38].

This makes the scheme a better fit for some applications, especially in wireless communications where the channels are subject to different kinds of errors [43]. On the other hand, the parallel implementation reduces the latency overhead. Moreover, the proposed scheme can be used with tiny devices such as smart phones or sensors because it is based on simple logical operations, and it is flexible (h can be changed) since it can be adapted to the available memory size.

1.3 Organization

The rest of this paper is organized as follows. In Section 2, we analyze the schemes of [37] and [38] and we highlight their limitations. Then, the proposed solution is presented in Section 3, and the description of the proposed stream cipher is presented in details in Section 4. Next, in Section 5, we assess the cryptographic performance of the proposed approach to show that it exhibits the required cryptographic strength. In Section 6, different metrics are analyzed such as latency, error propagation, space complexity and according to the obtained results, better performance is obtained as compared to [37, 38]. Finally, in Section 7, a conclusion summarizes our work and future work directions are presented.

2 Analysis of the [37, 38] Cipher Schemes

The cipher algorithms of [37, 38] operate on an entire image and consist of only two rounds of substitution-diffusion processes combined with a dynamic key changing for every input image as shown in Figure 1. An image is divided into a set of sub-matrices $\{X_1, X_2, \ldots, \ X_\alpha\}$, where each X_i has a square dimension $(h \times h)$, $i = 1, 2, \ldots, \alpha$ and α is the number of sub-matrices of the corresponding image.
The cipher in [38] employs a binary diffusion operation instead of the arithmetic one used in [37], which simplifies the hardware implementation of the proposed cipher and reduces the required latency, energy consumption and simplifies its hardware implementation.

Extensive security tests have demonstrated the high security and efficiency of these ciphers compared to state-of-the-art approaches. However, the authors indicated that the presented ciphers are based on Forward and Backward CBC, which prevents the parallelism process but ensures the avalanche effect in the whole image. This means that each sub-matrix is encrypted and must apply the round function for two iterations (r=2 in [37, 38]). However, ensuring the avalanche effect in the whole image has an impact in terms of error propagation. By quantifying the effect of error propagation (see Section 6.1), an erroneous sub-matrix adds an overhead to the original CBC mode for each erroneous block. This has a negative impact especially for a higher dimension value of sub-matrix h. Moreover, several applications or systems operate within erroneous channels such as the case in wireless communication channels. This limits these solutions and makes them unsuitable for such applications. Thus, an efficient cipher scheme with lower error propagation is necessary, which is the main objective of this paper.

In terms of latency, employing CBC typically limits the parallelism for encryption but allows for parallel decryption. However, by employing CBC-Forward then Backward, the parallel computation is limited in both encryption and decryption. This prevents devices that are capable of parallel computations from reducing the required execution time. For example, all current smart phones have multi-core capabilities. This is why, our proposed solution is based on the counter mode and consequently, the required latency can be reduced compared to the original schemes of [37, 38]. Moreover, theoretically, this reduction can be more pronounced depending on the number of threads. Systems with multi-threads are preferred since they require lower memory and resources.

The diffusion operation of [37] is based on arithmetic addition and multiplication operations, which requires a large number of cycles and hence, high execution time. While [38] uses another kind of diffusion operation, which is based on the binary diffusion matrix and only employs the logical Xor operation. In fact, the integer form of the diffusion matrix of [37] is mapped to a binary diffusion matrix in [38]. Therefore, the proposed cipher scheme in this paper follows [38] by using the binary diffusion matrix. As such, all these modifications are introduced to define a new lightweight image encryption algorithm that has simple hardware and software implementations, lower latency and resources in addition to lower error propagation and a high security level.

In the following section, the proposed cipher scheme and its corresponding dynamic key generation and cipher primitives construction are described in details.

3 The Proposed Dynamic Key-Dependent Cipher Approach

The proposed approach is divided into three steps:

- Dynamic Key Derivation;
- Construction of Cipher primitives;
- Encryption or Decryption algorithm.

In this following, these steps are described in details. In addition, all the notations used are given in Table 1.

3.1 Key Derivation

In order to have a lower complexity and easier implementation on devices, we consider one Secret Key (SK) shared only between both entities (transmitter and receiver). To provide better security, the symmetric secret key can be renewed after a specific periodic interval that can be chosen according to the application. The secret key can be renewed by employing any secure key establishment technique.

In addition, a new Session Secret Key (SSK) is produced from each new session. This is done by hashing the output of the "exlusive or" mixing between SK and Nonce N_o, as described by the following equation:

$$SSK = h(SK \oplus N_o) \tag{1}$$

where h is a cryptographic hash function such as SHA-512. Note that this operation ensures the sensitivity of the cipher key and nonce, simultaneously.

The length of SK can be equal to 128, 256 or 512 bits, while N_o has a fixed size of 512 bits, and it can be produced at the sender and receiver in a synchronized manner by using any secure Deterministic Pseudo Random Generator (DPRG) [44]. The seed of the selected DPRG can be constructed by hashing the secret key with any new public unique parameter. The produced pseudo-random sequence can be divided to form a set of dynamic Nonce values. In principle, the pseudo-random sequence of each of these DPRG ensures a high randomness level and high periodicity (internal update after each maximum "$requested\_number\_of\_bits$"). As such, each Nonce is used only once and it is different for every input image. Alternatively, the Nonce can be generated at the sender side and transmitted in an encrypted form to the receiver by employing the common secret key or by using the receiver public key. In order to achieve a high level of security, each image is encrypted using a dynamic key (DK) of size 512 bits, which is generated for every input image by hashing the output of the "exclusive or" mixing between the session key, which has a 512-bit length (SHA-512 is used) with a counter CT (flexible size) as follows:

$$DK = h(SSK \oplus CT) \tag{2}$$

As such, different dynamic keys can be produced for each new session leveraging the strong collision of the employed cryptographic hash function.

The proposed cipher has a key-dependent structure since each cipher primitive is related to the dynamic key as shown in Figure 2. Then, the dynamic key is divided into 4 dynamic sub-keys ($DK = \{DK_1,\ DK_2,\ DK_3,\ DK_4\}$) such that each one of them has a size of 16 bytes (128 bits). These sub-keys are used to construct 4 different cipher primitives, which are described next.

3.2 Proposed techniques to construct key-dependent cipher primitives

- **Pseudo-Random Matrix Initialization**: DK_1 represents the 16 most significant bytes of DK and it is used as a dynamic seed for a stream cipher. We do not use $RC4$ in the context of a stream cipher, instead, RC4 is iterated with a dynamic sub-key and it is used in this paper to generate dynamic cipher primitives. Therefore, no weakness is introduced and the security level will not be degraded. In fact, any stream cipher can be selected here instead of RC4, which was selected due to its simple hardware and software implementations. RC4 should be iterated for $h \times h$ times to produce the required key-stream, which is reshaped to produce the required initial counter matrix IM.

- **Substitution table (S-box S)**: Dk_2 represents the second set of the 16 most significant bytes and it is used to produce a dynamic substitution table (S-box). The proposed cipher can employ any key-dependent substitution generation table algorithm. In fact, the presented technique is based on the Key Setup Algorithm (KSA) of RC4 as in [38], whereby the KSA of RC4 stream cipher is selected since it possesses the simplest hardware and software implementations. Note that the output of KSA is a substitution table that is used as a dynamic S-box in the proposed approach.

- **Permutation table**: DK_3 represents the second least significant 16 bytes of DK and it is used to produce a permutation table (P-box π) that is employed after encrypting all plain sub-matrices. This produced permutation table is used to permute the encrypted sub-matrices to randomize the linear order of encrypted counter matrices. This will ensure a better security level for the proposed stream cipher. In this paper, the KSA of RC4 is modified to produce flexible P-boxes, where the length of the produced output table of KSA becomes variable (α, which is the number of sub-matrices) and not 256. The input length is introduced as an input for the Modified KSA (MKSA) function. Note that the produced S-boxes have a fixed size of 256 bytes.

- **Diffusion matrix** G: DK_4 represents the first 16 least significant bytes. DK_4 is used to produce a dynamic diffusion matrix G that can be binary or integer. In order to reduce complexity, a binary diffusion layer is recommended in [38] compared to other diffusion techniques such as Matrix Distance Separate (MDS) as in AES or hill cipher (invertible integer square matrix). More details can be seen in [38]. Indeed, DK_4 is used as a seed for the RC4 stream cipher to produce a sequence VA with $lv = \frac{n}{2} \times \frac{n}{2} \times q$ bits in length, where q represents the precision in bits and n is the dimension of the square diffusion matrix. q is equal to 1 in the case of a binary diffusion matrix and 8 in case of the byte integer matrix. Then, VA is reshaped to a square sub-matrix called A with size $\frac{n}{2} \times \frac{n}{2}$, which is necessary to form the square diffusion matrix G as described in [37, 38]. Note that the diffusion operation in this paper is realized at the matrix level and not at the block level (n bytes in a block) as presented in [38].

Moreover, the size of these sub-dynamic keys is large enough to produce a high number of unique cipher primitives that can reach the desired cryptographic performance. All notations are shown in Table 1. These steps are enough to ensure a high sensitivity since any change in the dynamic key will lead to completely different parameters in the encryption process and this is proven is Section 5.3.2. Derivation of these parameters is illustrated in Figure 2.

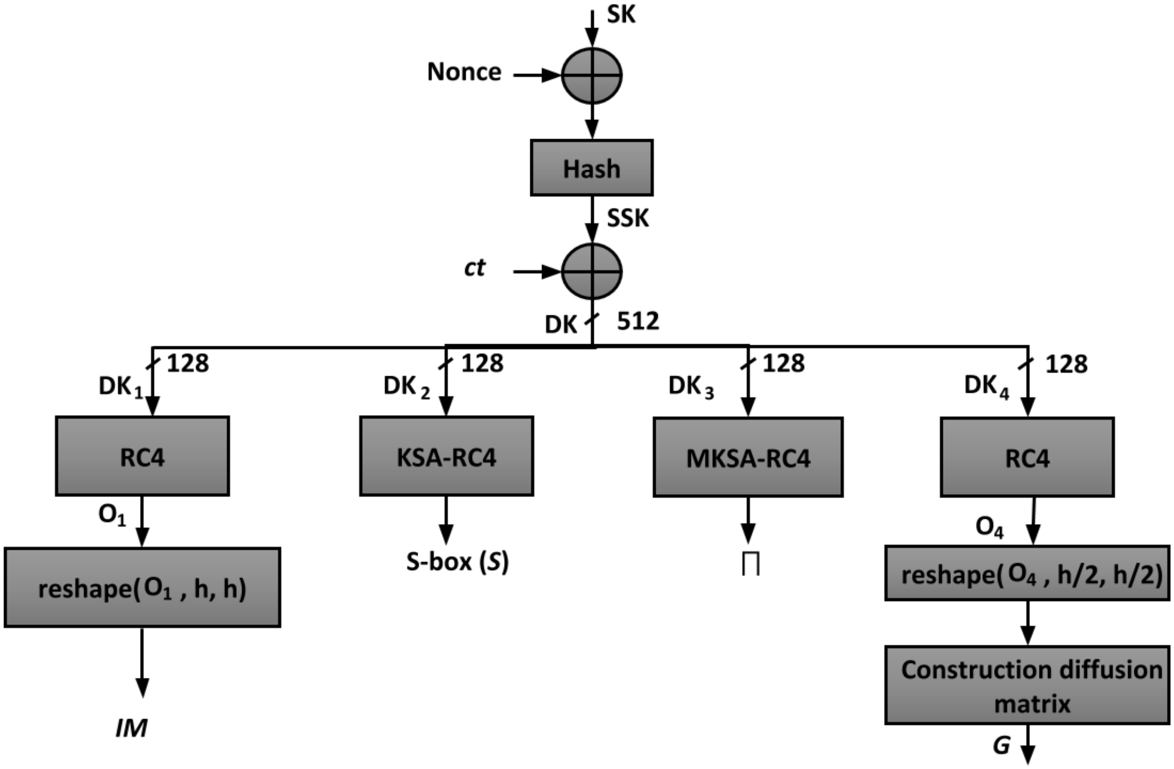


Figure 2: Proposed Dynamic key generation and cipher primitives.

Table 1: Table of notations

Notation	Definition
K	Secret key
N_o	Nonce
DK	Dynamic Key
DK_1	Initialization matrix sub-key
DK_2	Substitution sub-key
DK_3	Permutation sub-key
DK_4	Diffusion sub-key
S	A dynamic produced substitution table
S^{-1}	The inverse corresponding substitution table
π	A dynamic produced permutation box
π^{-1}	The inverse corresponding permutation box
G	A dynamic diffusion matrix (Integer or binary)
G^{-1}	The corresponding inverse diffusion matrix (Integer or binary)
r	Number of rows of the input image
c	Number of columns of the input image
p	Number of plane of the input image
n	Dimension of the square sub-matrix (n=4, 8, . . . , 32)
α	Number of sub-matrices in corresponding image
X_i	The i^{th} sub-matrix of the plain image
C_i	The i^{th} encrypted sub-matrix (without sub-matrices permutation operation)
q	q=1 for binary Galois field and its equal to 8, 16, 32, 64 for integer Galois field

Only the intended receiver that has the secret key and can produce the correct set of Nonce(s) can consequently produce the correct session keys and in consequence a set of dynamic keys for each new session. Each dynamic key is used to decrypt its corresponding encrypted image. Furthermore, the different steps of the proposed scheme at the sender and receiver sides are described below in details.

4 Proposed Stream Cipher Scheme

The proposed stream cipher uses a dynamic key-dependent structure in contrast of the existing stream ciphers. The main target of this solution is to reinforce the security level of existing stream ciphers by using the dynamic key approach but with a minimal possible overhead in the initialization phase. This paper follows the enhancement of our recent previous block ciphers [37, 38]. Furthermore, the main reason to design this stream cipher is due to the advantages in terms of parallel computation, minimum error propagation and high level of security that can be reached by using the proposed enhancements.

The proposed stream cipher scheme uses four dynamic key-dependent cipher primitives, which are:

1. Initial pseudo-random matrix IM;

2. A key-dependent substitution table called S;
3. A key-dependent permutation table π;
4. A dynamic invertible integer or binary diffusion matrix called G;

The input multimedia message (audio, image, video) is divided into α square sub-matrices $\{X_1,\ X_2,\ \ldots,\ X_\alpha\}$, where each one has a size of $(n \times n)$. It should be noted that no padding operation is required since the proposed solution is a stream cipher and not a block cipher [45].

The proposed stream cipher structure has also a new modification compared to a traditional stream cipher, which is to randomize the order of key-stream sub-matrices, which is done via π. The attacker's task becomes very complicated because of the different non-linear order of key-stream sub-matrices in addition to having dynamic key-stream sub-matrices for each new input message (image, video, etc.)

In fact, a traditional stream cipher applies the same operation at both entities. While with this approach, the dynamic permutation process is introduced and applied after the encryption process. This means that the intended receiver should apply a modification by starting with the inverse permutation operation then, decrypting the encrypted sub-matrices. In fact, the encryption and decryption processes (without permutation) are the same. They can be realized by mixing the plain and cipher sub-matrices with the corresponding produced key-stream sub-matrices, respectively. Moreover, the proposed encryption algorithm consists of applying the round function of [37] twice as shown in Figure 3. This round function consists of two main operations, a substitution operation using the S-box S, and a diffusion matrix G for the first round, and G^t for the second iteration. In addition, G can be a binary diffusion matrix according to [38].

4.1 Proposed Key-stream Generation Algorithm

In this part, we describe the techniques to produce the required key-stream sub-matrices $\{W_1,\ W_2,\ \ldots,\ W_\alpha\}$, which can be performed in parallel.

To generate the i^{th} key-stream matrix W_i, several operations are required; first, we apply a conversion of the numeric value i to byte data type and then, the output is reshaped to $(h \times h)$ matrix form (padding with 0 if necessary) to form a counter matrix T_i. Then, the matrix IM is mixed with T_i to produce the sub-matrix Q_i as indicated in the following equation:

$$Q_i = IM \oplus T_i, \qquad i = 1,\ 2,\ \ldots,\ \alpha \tag{3}$$

Next, and to achieve the avalanche effect on the input sub-matrix Q_i, the key-stream generation algorithm is iterated for two rounds, and in between the two rounds, we apply a transpose matrix operation.

Accordingly, each resultant sub-matrix Q_i follows a round function, whereby we perform the substitution operation on Q_i and then, the output is diffused using the diffusion matrix G. This can be summarized in the following equation:

$$Y_i = RF(Q_i, S, G) = G \odot S(Q_i), \ \ i = 1,\ 2,\ \ldots,\ \alpha \tag{4}$$

where Y_i represents the i^{th} substituted diffused sub-matrix of Q_i. Then, the sub-matrix Y_i is transposed ($Z_i = Y_i^t$); this is to increase the sensitivity to the counter value being used. In fact, the first diffusion operation propagates the difference of counter in its specific column, and the transpose propagates the difference to all rows after the second round function.

Finally, the round function is applied for a second time with a slight difference, since the transpose of G is used instead of G for the diffusion operation as indicated in the following equation:

$$W_i = RF(Z_i, S, G^t) = G^t \odot S(Z_i) \tag{5}$$

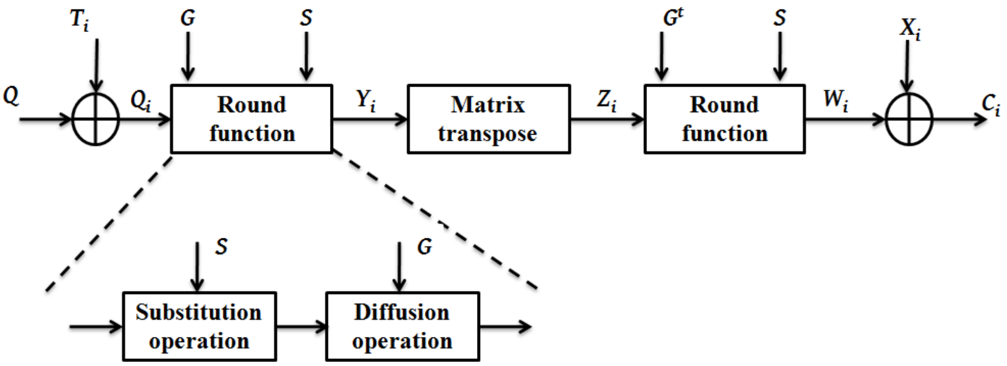


Figure 3: Architecture of the Proposed Stream Cipher

4.2 Encryption Algorithm

The encryption/decryption processes are performed by mixing each produced key-stream sub-matrix W_i for $i = 1,\ 2,\ \ldots,\ \alpha$ with the corresponding plain (X_i) (in encryption) and cipher sub-matrix (C_i) (in decryption). Therefore, to obtain the cipher sub-matrices, the encryption process is performed according to the following equation:

$$C_i = X_i \oplus W_i, \qquad i = 1,\ 2,\ \ldots,\ \alpha \tag{6}$$

After encrypting all the sub-matrices $\{X_1,\ X_2,\ \ldots,\ X_\alpha\}$, a permutation process is applied using the produced permutation table π. This operation permits to randomize the order of encrypted sub-matrices, which leads to an increase in the security level and prevents any future vulnerability in the key-stream generation. Then, a reshaping operation is applied on the cipher permuted sub-matrices $\{C_{\pi(1)},\ C_{\pi(2)},\ \ldots,\ C_{\pi(\alpha)}\}$ to form the cipher image.

4.3 Decryption Algorithm

Similarly to the encryption scheme, the same scheme is used to produce the key-stream sub-matrices $\{W_1,\ W_2,\ \ldots,\ W_\alpha\}$. In addition, the encrypted image is divided into a set of sub-matrices similarly to the sender side. Then, the decryption process starts by applying the inverse permutation process using the inverse permutation table π^{-1}. Next, each output inverse permuted sub-matrix C_i is mixed with its corresponding key-stream sub-matrix W_i to obtain the original sub-matrix X_i, which is described by the following equation:

$$X_i = C_i \oplus W_i, \qquad i = 1,\ 2,\ \ldots,\ \alpha \tag{7}$$

After decrypting all the sub-matrices $\{C_1,\ C_2,\ \dots,\ C_\alpha\}$, a reshaping operation is applied on the decrypted sub-matrices $\{X_1,\ X_2,\ \dots,\ X_\alpha\}$ to recover the original image.

4.4 Binary Diffusion Operation [38]

[38] analyzes the execution time of the presented cipher in [37]. The results show that the diffusion operation consumes the largest percentage of execution time and it increases with the dimension of the diffusion matrix. Hence, to reduce the required execution time of the diffusion operation, the authors recommend to perform the diffusion in the binary Galois field, which permits the use of the logical operation "exclusive or" instead of the arithmetic addition and subtraction operations. Consequently, the latency and energy consumption are reduced while the throughput is increased.

Therefore, in our proposed approach, we use the binary diffusion matrix form of [38], which is presented by the following equation:

$$G = \begin{pmatrix} A & A \oplus I_m \\ A \oplus I_m & A \end{pmatrix} = G^{-1} \tag{8}$$

Where G, as indicated previously, is a square $n \times n$ binary matrix and it consists of n diffusion vectors $\{G_1,\ G_2,\ \dots,\ G_n\}$. Each binary diffusion vector G_j is represented by a sequence of independent random numbers from a binary Galois field, where $G_{j,w}$ is a binary diffusion coefficient of line j and column w and $j, w = 1,\ 2,\ \dots,\ n$ and can be equal to 0 or 1. This means that if $G_{j,w}$ is equal to 0, its corresponding row w is not introduced into the diffusion process of the j^{th} diffused row. The values of the different indices in each vector (G_j) equal to 1 correspond to the row introduced in the diffusion process. The diffused row is the result of m XOR-ing bytes, where the corresponding index of its diffusion vector is 1.

In addition, the advantage of the proposed form is that the transpose of G is equal to G. Consequently, the required execution time of both diffusion operations is the same.

5 Security analysis

The proposed encryption algorithm should be strong enough to guard against the most known types of attacks such as statistical, differential, chosen/known plain-text, and brute-force attacks [46, 47, 48]. In this section, we perform extensive experiments, and we analyze several metrics to demonstrate the efficiency and cryptographic strength of the proposed scheme against these well-known attacks. The statistical results for the listed security metrics are shown in Table 2. In fact, the obtained results in Table 2 are very close to the statistical results of [37, 38] (see Table 2).

5.1 Statistical Analysis

To resist statistical attacks, a cipher must exhibit specific random properties [49]. To assess the randomness degree of the proposed cipher, we carry out the following statistical security tests: (a) Probability Density Function (PDF) analysis, (b) Entropy analysis and (c) Correlation between plain and encrypted images.

5.1.1 Uniformity Analysis

One important randomness property to resist the common statistical attacks is the uniformity of the PDF of the encrypted image. This means that each symbol has an occurrence probability close

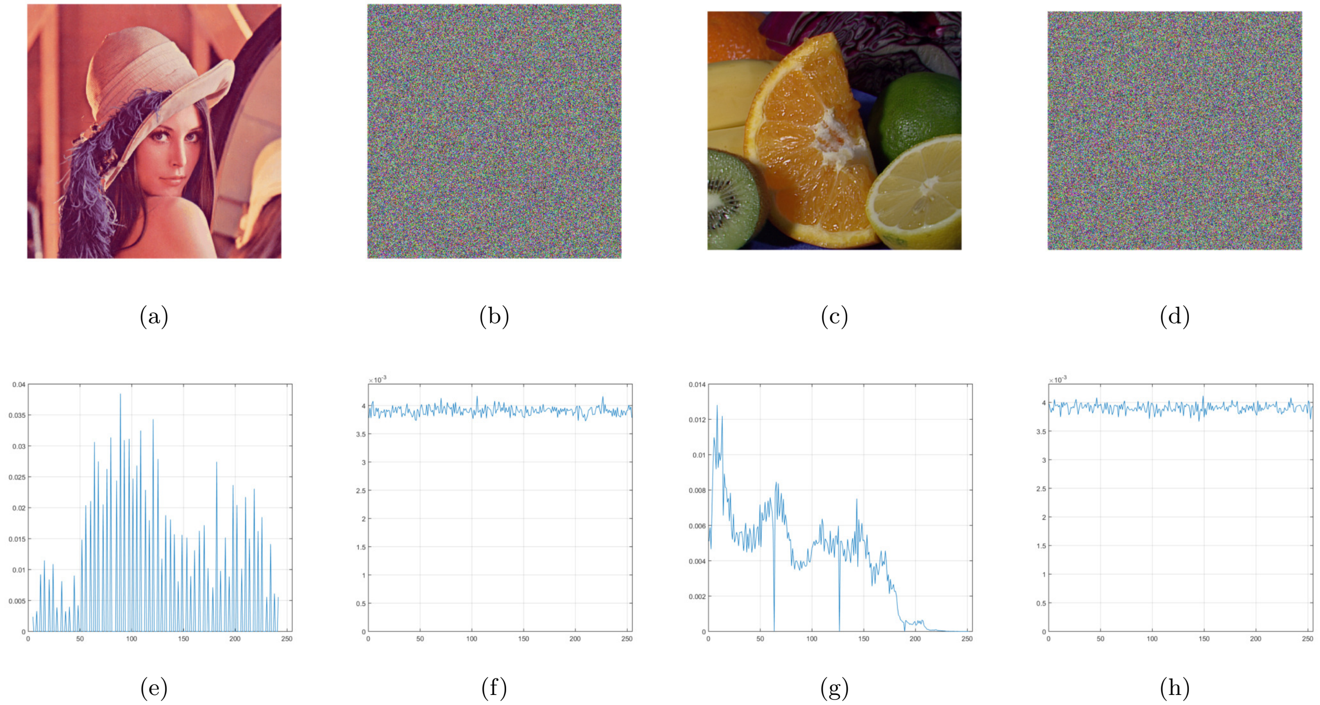


Figure 4: (a) Original Lenna image, (b) encrypted Lenna image, (c) original Fruits image, (d) encrypted Fruits image, (e) PDF of original Lenna image, (f) PDF of encrypted Lenna image, (g) PDF of original Fruits image, and (h) PDF of encrypted Fruits image

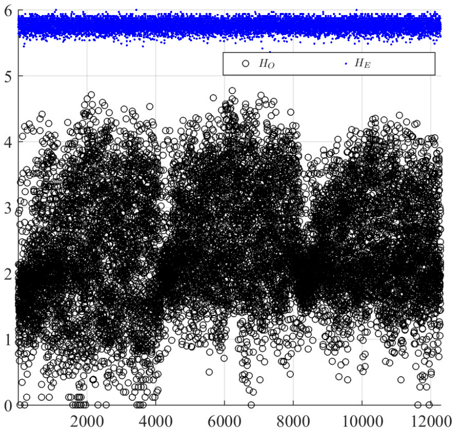
to $\frac{1}{n}$, where n is the number of symbols. Two original plain-images (Lenna and Fruits) and their corresponding cipher images are shown in Figure 4 along with their PDFs. We can see that the PDF of the encrypted images is close to a uniform distribution with a value close to 0.039 that is $\frac{1}{256}$. To validate this result at the sub-matrix level, an entropy test is performed next.

The information entropy of a data sequence M is a parameter that measures the level of uncertainty in a random variable [50], and is defined using the following equation:

$$H(m) = -\sum_{i=1}^{n} p(m_i) \log_2 \frac{1}{p(m_i)} \tag{9}$$

where $p(m_i)$ represents the occurrence probability of the symbol m_i and n is the total number of states of the source information. Note that the entropy is expressed in bits. We calculate the entropy at the sub-matrix level that has h^2 elements. Each sub-matrix can be considered a truly random source with uniform distribution if the value of entropy is close to the desired value, which is $\log_2(h^2)$ for $h^2 \leq n$ or $\log_2(n)$ for $h^2 > n$.

The entropy variation for the original and encrypted images (Lenna and Fruits) at the sub-matrix level with $h = 8$ and under the use of a random dynamic key is shown in Figure 5. The results indicate clearly that the encrypted sub-matrices have always an entropy close to 5.76, which is close to the desired value ($\log_2(64) = 6$) in the case of $h = 8$. This confirms the cipher resistance against statistical attacks.



(a) (b)

Figure 5: Entropy analysis for the sub-matrices of (a) original (black circles) and encrypted (blue points) Lenna images, and (b) Fruits images.

5.1.2 Correlation Test between Original and Cipher Images

Removing the correlation between pixels of an image is a key factor to verify if a scheme is successful or not. Removing spacial redundancy will certainly result in an efficient cipher scheme [51, 52]. Having a correlation coefficient close to zero means that the cipher scheme ensures a high degree of randomness. The correlation test is performed by taking randomly $N = 4,096$ pairs of adjacent pixels from Lenna plain image and their corresponding pixels in the cipher image. The correlation is done in horizontal, vertical and diagonal directions. The correlation coefficient r_{xy} is calculated using the following equation:

$$r_{xy} = \frac{\mathrm{cov}(x,y)}{\sqrt{D(x) \times D(y)}} \tag{10}$$

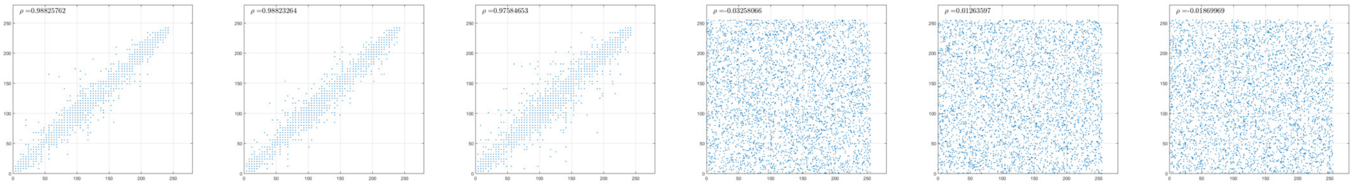
where

$$E_x = \frac{1}{N} \times \sum_{i=1}^{N} x_i$$

$$D_x = \frac{1}{N} \times \sum_{i=1}^{N} (x_i - E(x))^2$$

$$cov(x,y) = \frac{1}{N} \times \sum_{i=1}^{N} (x_i - E(x))(y_i - E(y))$$

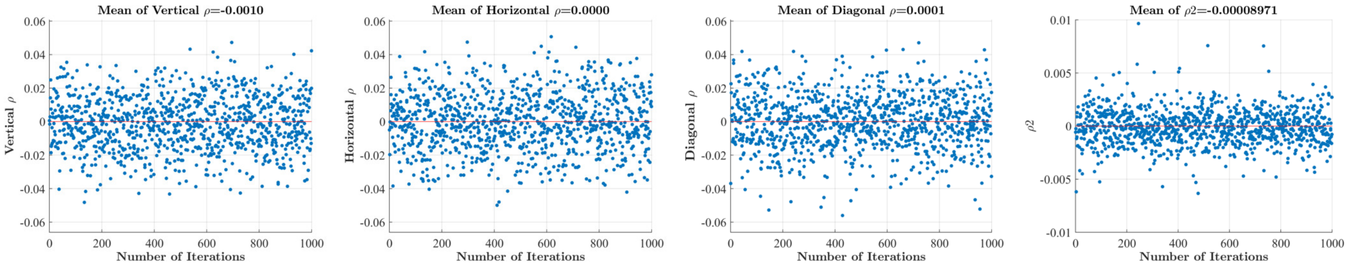
The obtained results for the encrypted Lenna image are presented in Figure 6. The results indicate clearly the high correlation between adjacent pixels in plain image (correlation coefficient is close to 1). However, for the cipher image, the correlation coefficient is very low (close to 0), which clearly shows that the proposed scheme reduces severely the spatial redundancy. In addition, the statistical results of correlation are shown in Table 2.



(a) (b) (c) (d) (e) (f)

Figure 6: Correlation distribution in adjacent pixels in original Lenna: (a) horizontally, (b) vertically and (c) diagonally.
Correlation in adjacent pixels in ciphered Lenna: (d) horizontally, (e) vertically and (f) diagonally.

Moreover, the variation of the correlation coefficient between adjacent pixels (ρ_h, ρ_d, ρ_v) of the encrypted Lenna image versus 1,000 random keys is shown in Figure 7 and its corresponding statistical performance is presented in Table 1. The results are always close to 0, which confirms that spatial redundancy is eliminated and no detectable relation can be found in the encrypted images. Similar results are obtained for the encrypted Fruits image.



(a) (b) (c) (d)

Figure 7: Variation of the coefficient of correlation for adjacent pixels in ciphered Lenna and Fruits: (a) horizontally, (b) vertically, and (c) diagonally, respectively. In addition, the variation of ρ_2 between original and encrypted Lenna image versus 1000 random dynamic keys.

In addition, the coefficient correlation between original and encrypted images (matrices) are obtained by applying the 2-D correlation coefficient. The result is shown in Figure 7-d and its statistical measure is presented in Table 1 (see value distribution of $\rho2$). The results indicate that the 2-D coefficient correlation varies in a small interval around 0. This means that low 2D-correlation coefficient is achieved by employing the proposed cipher approach, which confirms the statistical independence of the original and encrypted matrices.

5.1.3 Difference Between Plain and Cipher Images

The encrypted image must be as different as possible from the original image (at least 50%) at the bit level. The proposed scheme has achieved a high value of difference between both images.

Image Lena was tested to show that at least 50% of the image has been changed by the encryption process. The results are shown in Figure 8 and in Table 2.

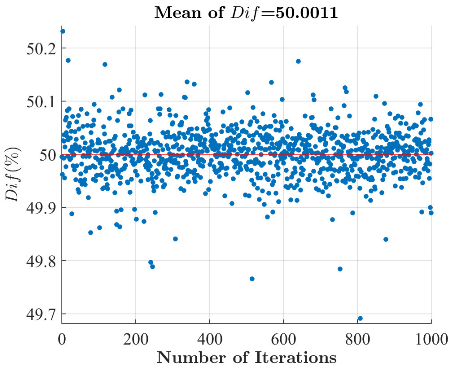


Figure 8: The variation of the difference between plain and cipher image Lenna (percentage of the Hamming distance) against 1000 random keys.

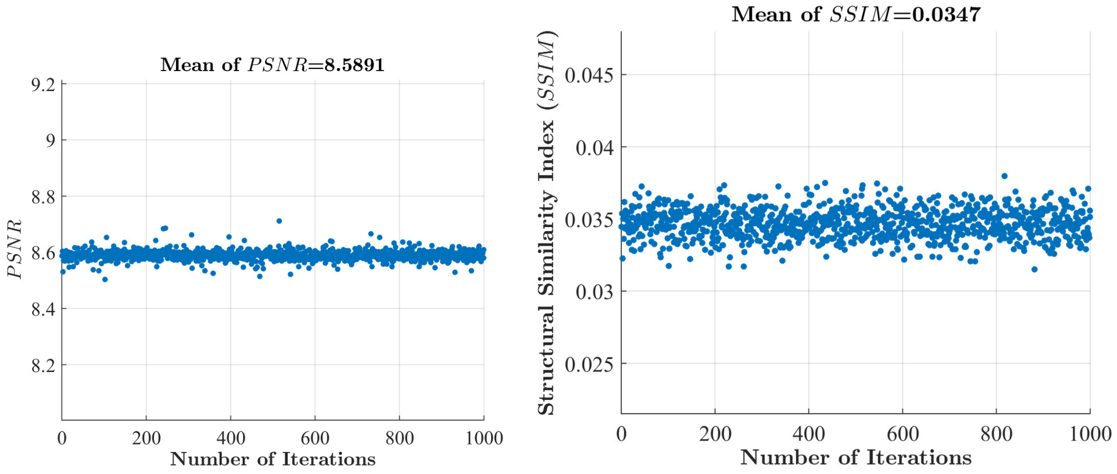
5.2 Visual Degradation

This test is specific for image and video contents and quantifies the level of visual degradation achieved by employing a cipher scheme. Typically, the degradation on the original image must be done such that the visual content in the cipher image must not be recognized. Two well known parameters are studied to measure the encryption visual quality, the Peak Signal-to-Noise Ratio (PSNR) [53] and the Structural Similarity index (SSIM) [54]. The PSNR is derived from the Mean Squared Error (MSE), where MSE represents the cumulative squared error between an original and encrypted image. A low PSNR value indicates that there is a high difference between the original and the cipher images.

Concerning SSIM index [17], it is defined after the Human Visual System (HVS) and has evolved so that we can extract the structural information from the scene. SSIM lies in the interval [0,1]. A value of 0 means that there is no structural similarity between the original and cipher images, while a value close to 1 indicates that the two images are approximately the same. In this context, PSNR and SSIM were measured between the original and the encrypted Lenna images for $1,000$ random dynamic keys and the results are shown in Figure 9-(a) and (b), respectively. We can see that the mean PSNR value is 9.23 dB. Also, the $SSIM$ values are always close to zero, which means that a high and hard visual distortion is obtained using the proposed cipher algorithm and as such, no useful visual information or structure about the original image could be revealed from the cipher image.

5.3 Sensitivity Tests

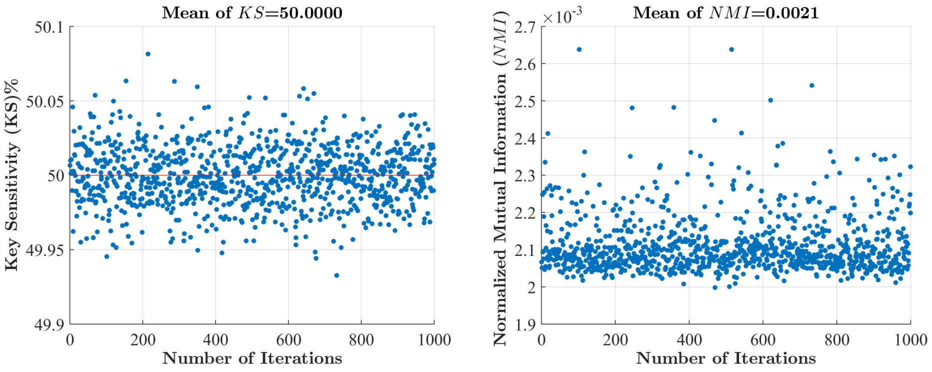
Differential attacks are based on studying the relation between two encrypted images resulting from a slight change on the plain text or secret key, usually one bit difference as compared to the original one. A sensitivity test shows by how much would a slight change in the plain-image or in the key affect the resulting cipher image. In this context, the larger the change, the better the sensitivity of the encryption algorithm. Below, we assess the sensitivity with respect to plain-text and to the key.



(a)

(b)

Figure 9: $PSNR$ and $SSIM$ variation between the original and the encrypted Lenna image versus 1000 keys.



(a) Key sensitivity

(b) Normalized MI

Figure 10: (a) The variation of the key sensitivity and (b) normalized MI, between two ciphered images and against 1,000 random dynamic keys.

5.3.1 Plain-text Sensitivity

This particular test is not a necessary one for the proposed approach since different dynamic keys are employed for each input image, which leads to different substitution and diffusion primitives in addition to the initial round and counter matrices. Consequently, this leads to totally different cipher images for the same plain image.

5.3.2 Key Sensitivity test

This is one of the most important tests that quantifies the sensitivity against any slight change in the secret key. In fact, the proposed key derivation function is based on a secret key and a Nonce.

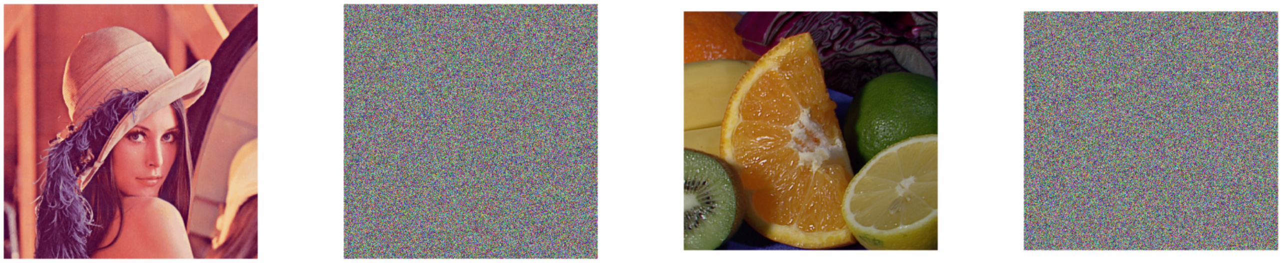
To study the key sensitivity, two dynamic keys are used: SK_1 and SK_2 that differ by only one random bit. Two plain-images are encrypted separately and then, the Hamming distance (in bits) between the two cipher-images is calculated as follows:

$$\begin{aligned} KS =& \frac{\sum_{k=1}^{T} C_1 \oplus C_2}{T} \times 100\% \\ =& \frac{\sum_{k=1}^{T} (E_{SK_1}(P)) \oplus (E_{SK_2}(P))}{T} \times 100\% \end{aligned} \tag{11}$$

Where T is the length in bits of the plain and cipher data.

The computed Hamming distance of the corresponding encrypted cipher-images C_1 and C_2 is illustrated in Figure 10-(a) against 1,000 random dynamic keys.

We can see that the majority of values is close to the optimal value (50 %), indicating that the proposed encryption scheme is robust and has enough strength against any little change in the dynamic key. Additionally, the obtained result of 49.9970 is sufficient and similar to the ones reported in [51, 55, 56, 57], which have values of 49.98, 49.97, 49.99 and 49.999, respectively. On the other hand, an example of decrypted original images (Lenna and Fruits) with the correct key are shown in Figure 11-(a) and (c), and with a modification of one random bit of the dynamic key are shown in Figure 11-(b) and (d).



(a) (b) (c) (d)

Figure 11: Decrypted original images (Lenna and Fruits) with the correct key (a) and (c), and decrypted ones with a modification of one random bit of the dynamic key (b) and (d), respectively.

5.3.3 Normalized Mutual Information (NMI)

MI is used to measure the similarity between two sequences S and S'. MI is based on the individual entropy $H(S)$ and the conditional entropy $H(S|S')$ defined as follows:

$$MI = H(C) - H(C|C') \tag{12}$$

$$H(S) = -\sum_{i=1}^{ns} p(s_i) \log p(s_i) \tag{13}$$

$$H(S|S') = -\sum_{i=1}^{ns}\sum_{j=1}^{ns'} p_{s_i,s'_j} \log p(s_i|s'_j) \qquad (14)$$

Where $p(s_i)$ represents the probability of occurrence of the symbol s_i. ns and ns' are the total number of states of information in S and S', respectively. p_{s_i,s'_j} denotes the joint probability that C is in state s_i and C' is in state s'_j. The values vary between 0 and 1. In order to confirm the Independence between encrypted images (with a slight change in the dynamic key), we applied NMI [38] between the encrypted images versus 1,000 random dynamic keys and the results are shown in Figure 10-(b) for Lenna image and its statistical performance is presented in Table 2. The results indicate that NMI is always close to 0 (with a mean equal to 0.0192). This indicates clearly that no meaningful information can be extracted from the encrypted images.
Note that in Tables 2 and 3, H_O and H_E represent the entropy of original and encrypted sub-matrices, respectively. In addition, in Table 2 ρ_h, ρ_d, and ρ_v represent the correlation coefficient between adjacent pixels in horizontal, vertical and diagonal directions, respectively.

Table 2: Statistical results of sensitivity for Lenna image using integer diffusion matrix (left) and binary one (right) for 1,000 random keys.

Integer Diffusion matrices					Binary diffusion matrices				
	Min	Mean	Max	Std		Min	Mean	Max	Std
Dif	49.69	50.001	50.23	0.0471	Dif	49.8859	50.0017	50.1239	0.0346
KS	49.932	49.99	0.0501	0.0199	KS	49.9012	49.9990	50.0941	0.0333
H_O	4.2014	4.2014	4.2014	0.0001	$H-O$	4.2014	4.2014	4.2014	0.0001
H_E	5.73	5.763	5.7678	0.005	$H-E$	5.7623	5.7657	5.7691	0.0012
NMI	0.0189	0.0193	0.0196	0.0001	NMI	0.0190	0.0192	0.0196	0.0001
$\rho 2$	-0.0056	0.0000	0.0063	0.0020	$\rho 2$	-0.0066	-0.0000	0.0066	0.0020
ρ_h	-0.0499	0.0000	0.0507	0.0169	$\rho-h$	-0.0523	0.0003	0.0589	0.0163
ρ_v	-0.0483	-0.0010	0.0472	0.0158	$\rho-v$	-0.0467	0.0001	0.0530	0.0163
ρ_d	-0.0561	0.0001	0.0471	0.0160	$\rho-d$	-0.0467	-0.0006	0.0577	0.0155
PSNR	8.5025	8.5891	8.7121	0.0177	PSNR	9.2031	9.2303	9.2626	0.0092
SSIM	0.0315	0.0347	0.0380	0.0010	SSIM	0.0310	0.0359	0.0422	0.0017

Table 3: Statistical results of [58] and of [38] for 1000 random keys with Lena as plain image. ID represents the integer diffusion matrix, where BD represents the binary diffusion one.

Tests	Mean of [58]	Std of [58]	Mean of [38] (ID)	Std of [38] (ID)	Mean of [38] (BD)	Std of [38] (BD)
Dif	49.8875	0.034	49.9998	0.0195	50.0007	0.0202
PS	45.6381	1.9728	49.9533	1.0992	50.041	1.185
KS	49.8741	0.0342	50.0003	0.0198	49.9992	0.0199
$H-O$	4.2014	0.79914	4.2014	0.79914	4.2014	0.79914
$H-E$	5.7657	0.0012	5.7658	0.0024	5.7657	0.00239
PSNR	9.2306	0.0096	8.5896	0.0054	8.5894	0.0054
SSIM	0.0359	0.0017	0.0359	0.0016	0.0359	0.0017

5.4 Cryptanalysis

In this section, we present typical cryptanalytic use cases with a brief analysis of the proposed cipher against several cryptanalytic attacks. We assume that the cryptanalyst has knowledge of the employed technique to generate the employed substitution and diffusion primitives but no knowledge of the secret and Nonce, which means that also no knowledge about the produced dynamic key. The previous scheme of [37] is resistant to different types of attacks since a dynamic key approach is employed. The proposed work ensures immunity against statistical, chosen/known plain text attacks. The proposed scheme can ensure a good statistical performance since uniformity and independence are attained in addition to key and initial matrix sensitivity. A successful sensitivity test shows how much a slight change in the key or initial matrix will affect the resulted cipher image. Moreover, the key space of the secret key can be 2^{128}, 2^{196} or 2^{256}, while the key space of the dynamic key is 2^{512}. Therefore, the key space for secret and dynamic keys are large enough to prevent brute-force attacks. Finally, the problem of single image failure and accidental key disclosure is avoided since the dynamic key is changed for every input image. Moreover, the size of dynamic keys complicates the recovery process of the dynamic key, and differential and linear attacks are inefficient. As such, this cipher can resist the different well-known attacks and its main goal is to benefit from the dynamic approach as a lightweight secure image encryption scheme.

6 Performance Analysis

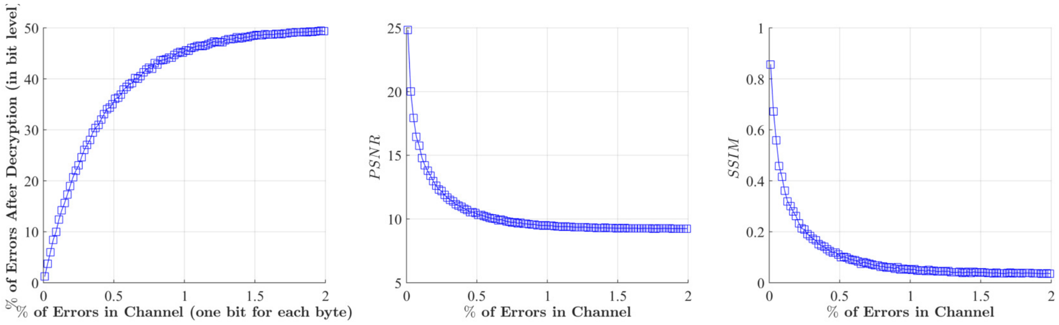
In this section, several criteria and tests, such as space complexity and execution time, are performed to show that the proposed solution can outperform [37, 38], and thus, it can cater for practical applications in a more efficient manner. In practice, the cipher scheme is efficient if it ensures low latency, memory and required resources during the ciphering/deciphering process whilst maintaining a high security level. The proposed cipher scheme employs the counter mode with a dynamic lightweight round function that is applied for two rounds. In addition, the proposed cipher can be implemented in parallel, where each sub-matrix can be encrypted independently from other sub-matrices.

6.1 Propagation of errors

An important criterion to guarantee is error tolerance, which means that an error in a block is not propagated to other blocks. Interference and noise within the transmission channels are the main causes of errors. We reefer to a bit error as the substitution of '0' bit by '1' bit or vice versa. Such an error may propagate and lead to the corruption of data, which is a challenge due to the trade-off between the Avalanche effect and error propagation as shown in [6].

If a bit error takes place in any byte of the encrypted sub-matrix C_i, it will affect three sub-matrices $\{\hat{X_{i-1}},\ \hat{X_i},\ \hat{X_{i+1}}\}$ in the decrypted image for [37, 38]. Two of them $\{\hat{X_i},\ \hat{X_{i+1}}\}$ have random bit errors that occur independently of the bit position with an expected probability of $\frac{1}{2}$ and the third sub-matrix $\hat{X_{i-1}}$ has only one specific bit error in the same error bit position. While, for the proposed scheme, the effect of bit error introduces only a specific bit error at the same error bit position in the decrypted image. This means that the error in the proposed scheme is not propagated and will not destroy the neighboring sub-matrices compared to [37, 38].

The difference between both decrypted images, as a function of the channel error percentage, is calculated and presented in Figure 12 (uniform distribution) for both approaches. The results indicate that the error in the decrypted image for [37] is close to 50% for a channel error $\geq 2\%$. The decrypted image versus the channel error values is presented in Figure 14. In addition, to quantify the visual degradation, the two well known parameters are used, the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM). We measure the $PSNR$ and $SSIM$ between two decrypted Lenna images (where the second decrypted image corresponds to the encrypted image with a specific error percentage). The variations of SSIM and PSNR versus the percentage of errors are presented for [37] in Figure 12-(b),(c). The results indicate that 0.5% of random errors in the channel (with uniform distribution) can destroy the image contents. Additionally, similar results are obtained for [38]. On the other hand, the same tests are applied for the proposed scheme and the results are shown in Figure 13. The proposed solution shows a linear difference. In addition, the variations of SSIM and PSNR for the proposed approach show that it can resist a high value of channel error compared to [37]. This conclusion can be justified by showing the decrypted image with high values of channel errors that are shown in Figure 15. Therefore, we can deduce that the approaches of [37, 38] are inefficient compared to the proposed one from an error propagation viewpoint.



(a) (b) (c)

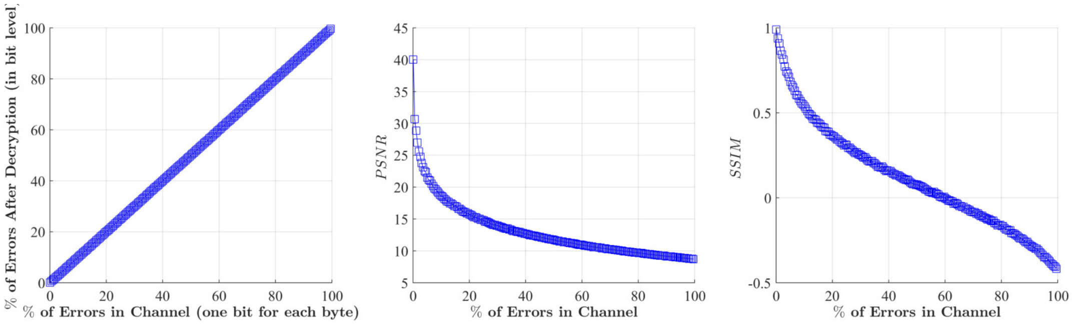
Figure 12: Variation of the impact of error propagation (% of bits difference between both decrypted images) (a) and the variation of $SSIM$ (b) and PSNR (c) compared to the percentage of errors in channel for [37].

6.2 Space Complexity

To perform the encryption of one sub-matrix, we only need the current sub-matrix without the previously encrypted one as in [37]. Hence, the total space complexity is reduced to a much lower value than $2 \times O(h^2)$ where $O(h^2)$ designates the space complexity for one sub-matrix of size $h \times h$ to $O(h^2)$.

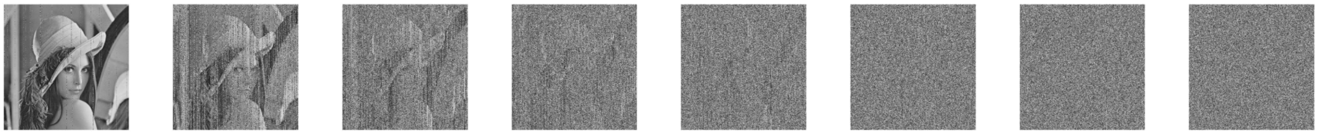
6.3 Execution Time

The main motivation of this paper is to re-design the algorithm proposed in [37] in order to make it lightweight with minimum execution time to address the time limitations of real-time applications and to reduce the required resources, especially, the low energy consumption of calculation, which is critical for constrained devices using a battery. Note that the recent algorithm of [37] was compared to recent cipher schemes and it was verified that it achieved lower execution time compared to the



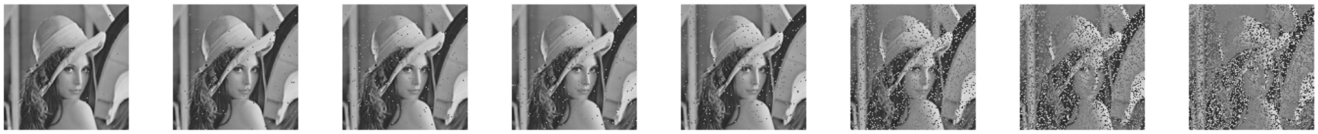
(a) (b) (c)

Figure 13: Variation of the impact of error propagation (% of bits difference between both decrypted images) (a) and the variation of $SSIM$ (b) and PSNR (c) versus the percentage of errors in channel for the proposed approach.



(a) 0.01% (b) 0.25% (c) 0.5% (d) 0.75% (e) 1% (f) 1.5% (g) 1.75% (h) 2%

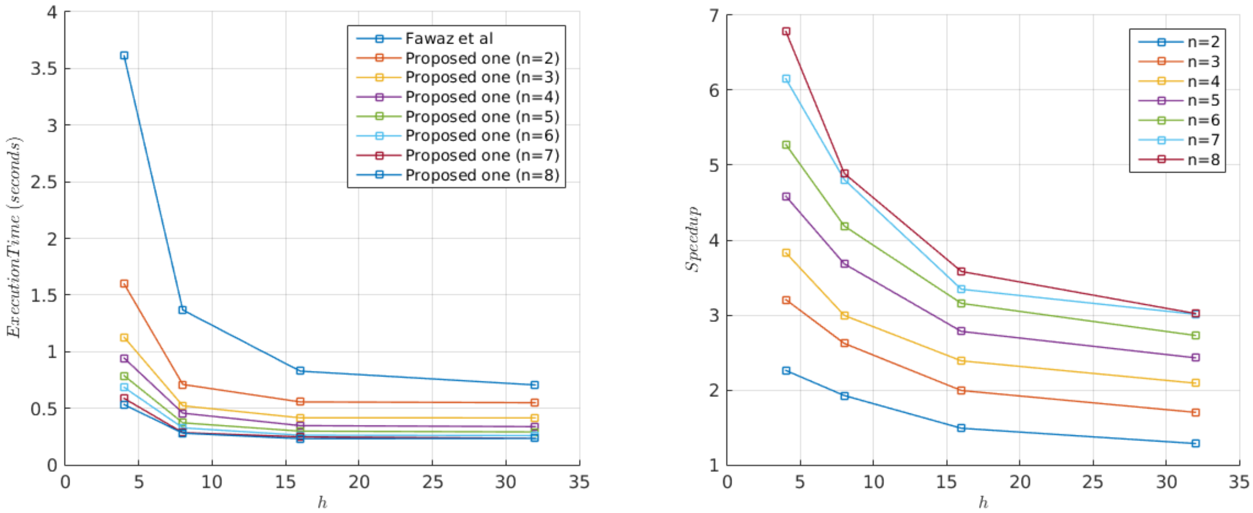
Figure 14: Decrypted images as a function of the percentage of channel errors for [37].



(a) 0.01% (b) 0.6% (c) 1.1% (d) 1.6% (e) 2.1% (f) 10.1% (g) 20.1% (h) 30.6%

Figure 15: Decrypted images as a function of the percentage of channel errors for the proposed cipher.

selected schemes. The average calculation time (within 0 iterations) for encrypting a color image of size 1024×1024×3 is performed using the following software and hardware environment: MATLAB R2015a simulator, 2 Intel(R) Xeon(R) CPU E5-2623 v4 @ 2.60GHz running Debian Linux. The time analysis is applied for the recent cipher algorithm of [37] and the proposed one with the integer diffusion operation. The results are shown in Figure 16-a for different h values (4, 8, 16 and 32) and for different number of processes for our approach $n = 2, 3, 4, 5, 6, 7, 8$. Moreover, the ratio between the scheme of [37] and the proposed one with parallel computing for the different values of n is shown in Figure 16-b. This test is performed using the same machine with the same environmental conditions. First of all, the required execution time of the proposed scheme without parallel computing is really close to the execution time of [37] with a binary diffusion operation. Accordingly, the proposed scheme without parallel computing attains minimum execution time for a high value of h, while a lower value of h requires more execution time. However, this requires in



(a) Execution times between our approach and the approach of [37]

(b) Speed up of the new cipher operation

Figure 16: Average of execution times between our approach and the approach of [37] in function of h and number of (n) processes for 100 iterations (a) and the corresponding speedup between the parallel implementation and the implementation of [37].

contrast more memory storage to reduce the execution time.

Then, we perform a comparison with the proposed scheme while performing parallel computations, and obviously, the proposed approach is outperforms [37], which cannot be parallelized. In addition, the results clearly indicate that increasing the number of threads (n) permits to reduce further the execution time. On the other hand, for a small value of h, the reduction of the execution time is significant. This means that the proposed scheme with parallel computing has the required stable low execution time for different dimensions of sub-matrix h. As such, a significant reduction in terms of execution time is ensured by employing the binary diffusion operation and it becomes smaller by introducing parallel computing. This makes the proposed scheme more suitable for real-time applications compared to [37].

6.4 The choice of one sub-matrix size h^2

There is often a time-space trade-off involved in each proposed scheme. Hence, a compromise must be made to exchange computing time for memory consumption or vice-versa, depending on the application requirements and the user's priority. The parameter h has a significant impact on the required memory size and no effect on the execution time when parallel processing is possible. However, increasing the value of h necessitates a larger memory storage. For this reason, the proposed approach is flexible, and h can be chosen according to the application requirements and the possibility of applying parallel computations. For example, in the cloud computing area, no memory constraints exist and parallel computing is possible. Therefore, a higher value of h can be employed (16 or 32). However, for tiny devices, where the memory capacity is really limited, a low

h value must be chosen (2 or 4).

7 Conclusion

In this paper, we have analyzed the cipher presented in [37, 38]. We demonstrated that these schemes suffer from two limitations, namely the effect of error propagation and the lack of inherent parallelism. These shortcomings limit the applicability of the original scheme of [37] to many real-time applications as well as its deployment in wireless communications systems. A modified scheme that mitigates these disadvantages has been proposed, which is more efficient since 1) it requires a lower latency by applying parallel processing, and 2) it is more resilient to channel errors. These results are presented in order to prove the credibility and the safe deployment of the proposed scheme. Moreover, its security level is equivalent to those of [37, 38] due to inherent key and Nonce sensitivities. Furthermore, the same technique to generate the substitution and diffusion (binary) primitives of [37, 38] is employed.

Acknowledgement

This paper is partially funded from the Maroun Semaan Faculty of Engineering and Architecture at the American University of Beirut and by the EIPHI Graduate School (contract "ANR-17-EURE-0002").

References

[1] Brij Gupta, Dharma P Agrawal, and Shingo Yamaguchi. *Handbook of research on modern cryptographic solutions for computer and cyber security*. IGI Global, 2016.

[2] William Davros. Digital image processing for medical applications. *Medical Physics*, 37(2):948–949, 2010.

[3] Tinku Acharya and Ajoy K Ray. *Image processing: principles and applications*. John Wiley & Sons, 2005.

[4] Samer Atawneh, Ammar Almomani, Hussein Al Bazar, Putra Sumari, and Brij Gupta. Secure and imperceptible digital image steganographic algorithm based on diamond encoding in dwt domain. *Multimedia tools and applications*, 76(18):18451–18472, 2017.

[5] Morris Dworkin, Morris Dworkin, Patrick D. Gallagher, and Director Nist Special Publication f. Recommendation for block cipher modes of operation: Methods and techniques, 2001.

[6] Ayoub Massoudi, Frédéric Lefebvre, Christophe De Vleeschouwer, Benoit Macq, and J-J Quisquater. Overview on selective encryption of image and video: challenges and perspectives. *Eurasip Journal on information security*, 2008:5, 2008.

[7] Eli Biham and Adi Shamir. *Differential cryptanalysis of the data encryption standard*, volume 28. Springer-Verlag New York, 1993.

[8] Joan Daemen and Vincent Rijmen. *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media, 2002.

[9] Nahla A Flayh, Rafat Parveen, and Syed I Ahson. Wavelet based partial image encryption. In *Multimedia, Signal Processing and Communication Technologies, 2009. IMPACT'09. International*, pages 32–35. IEEE, 2009.

[10] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. Simon and speck: Block ciphers for the internet of things. *IACR Cryptology ePrint Archive*, 2015:585, 2015.

[11] Hannes Gross, Stefan Mangard, and Thomas Korak. An efficient side-channel protected aes implementation with arbitrary protection order. In *Cryptographers' Track at the RSA Conference*, pages 95–112. Springer, 2017.

[12] Seyed Mohammad Seyedzadeh and Sattar Mirzakuchaki. A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map. *Signal Processing*, 92(5):1202–1215, 2012.

[13] Yuping Hu, Congxu Zhu, and Zhijian Wang. An improved piecewise linear chaotic map based image encryption algorithm. *The Scientific World Journal*, 2014, 2014.

[14] Ying-Qian Zhang and Xing-Yuan Wang. Analysis and improvement of a chaos-based symmetric image encryption scheme using a bit-level permutation. *Nonlinear Dynamics*, 77(3):687–698, 2014.

[15] Chengqing Li, Michael ZQ Chen, and Kwok-Tung Lo. Breaking an image encryption algorithm based on chaos. *International Journal of Bifurcation and Chaos*, 21(07):2067–2076, 2011.

[16] Rhouma Rhouma, Ercan Solak, and Safya Belghith. Cryptanalysis of a new substitution–diffusion based image cipher. *Communications in Nonlinear Science and Numerical Simulation*, 15(7):1887–1892, 2010.

[17] Shujun Li and Xuan Zheng. Cryptanalysis of a chaotic image encryption method. In *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, volume 2, pages II–708. IEEE, 2002.

[18] Feng Huang and Yong Feng. Security analysis of image encryption based on twodimensional chaotic maps and improved algorithm. *Frontiers of Electrical and Electronic Engineering in China*, 4(1):5–9, 2009.

[19] David Arroyo, Chengqing Li, Shujun Li, Gonzalo Alvarez, and Wolfgang A Halang. Cryptanalysis of an image encryption scheme based on a new total shuffling algorithm. *Chaos, Solitons & Fractals*, 41(5):2613–2616, 2009.

[20] Gonzalo Alvarez and Shujun Li. Cryptanalyzing a nonlinear chaotic algorithm (NCA) for image encryption. *Communications in Nonlinear Science and Numerical Simulation*, 14(11):3743–3749, 2009.

[21] Joan Daemen and Vincent Rijmen. *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media, 2013.

[22] Shay Gueron. Intel's new aes instructions for enhanced performance and security. In *FSE*, volume 5665, pages 51–66. Springer, 2009.

[23] Sean O'Melia and Adam J Elbirt. Enhancing the performance of symmetric-key cryptography via instruction set extensions. *IEEE transactions on very large scale integration (VLSI) systems*, 18(11):1505–1518, 2010.

[24] Kerry A McKay, Lawrence E Bassham, Meltem Sonmez Turan, and Nicky W Mouha. Report on lightweight cryptography. *NIST Interagency/Internal Report (NISTIR)-8114*, 2017.

[25] Deukjo Hong, Jung-Keun Lee, Dong-Chan Kim, Daesung Kwon, Kwon Ho Ryu, and Dong-Geon Lee. LEA: A 128-Bit Block Cipher for Fast Encryption on Common Processors. In *Information Security Applications*, pages 3–27. Springer, 2014.

[26] Manoj Kumar, Saibal K Pal, and Anupama Panigrahi. FeW: A Lightweight Block Cipher. *IACR Cryptology ePrint Archive*, 2014:326, 2014.

[27] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, et al. Prince–a low-latency block cipher for pervasive computing applications. In *Advances in Cryptology–ASIACRYPT 2012*, pages 208–225. Springer, 2012.

[28] Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi. TWINE: A Lightweight Block Cipher for Multiple Platforms. In LarsR. Knudsen and Huapeng Wu, editors, *Selected Areas in Cryptography*, volume 7707 of *Lecture Notes in Computer Science*, pages 339–354. Springer Berlin Heidelberg, 2013.

[29] Wenling Wu and Lei Zhang. LBlock: a lightweight block cipher. In *Applied Cryptography and Network Security*, pages 327–344. Springer, 2011.

[30] Kyoji Shibutani, Takanori Isobe, Harunaga Hiwatari, Atsushi Mitsuda, Toru Akishita, and Taizo Shirai. Piccolo: an ultra-lightweight blockcipher. In *Cryptographic Hardware and Embedded Systems–CHES 2011*, pages 342–357. Springer, 2011.

[31] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matt Robshaw. The LED block cipher. In *Cryptographic Hardware and Embedded Systems–CHES 2011*, pages 326–341. Springer, 2011.

[32] Radu Boriga, Ana Cristina Dăscălescu, and Iustin Priescu. A new hyperchaotic map and its application in an image encryption scheme. *Signal Processing: Image Communication*, 29(8):887 – 901, 2014.

[33] Dolendro Singh Laiphrakpam and Manglem Singh Khumanthem. A robust image encryption scheme based on chaotic system and elliptic curve over finite field. *Multimedia Tools and Applications*, pages 1–24, 2017.

[34] Mohammad Ghebleh, Ali Kanso, and Hassan Noura. An image encryption scheme based on irregularly decimated chaotic maps. *Signal Processing: Image Communication*, 29(5):618–627, 2014.

[35] Siva Janakiraman, K. Thenmozhi, John Bosco Balaguru Rayappan, and Rengarajan Amirtharajan. Lightweight chaotic image encryption algorithm for real-time embedded system: Implementation and analysis on 32-bit microcontroller. *Microprocessors and Microsystems*, 56(Supplement C):1 – 12, 2018.

[36] Bhaskar Mondal and Tarni Mandal. A light weight secure image encryption scheme based on chaos & dna computing. *Journal of King Saud University - Computer and Information Sciences*, 29(4):499 – 504, 2017.

[37] Zeinab Fawaz, Hassan Noura, and Ahmed Mostefaoui. An efficient and secure cipher scheme for images confidentiality preservation. *Signal Processing: Image Communication*, 42:90 – 108, 2016.

[38] Hassan Noura, Lama Sleem, Mohamad Noura, Mohammad M. Mansour, Ali Chehab, and Raphaël Couturier. A new efficient lightweight and secure image cipher scheme. *Multimedia Tools and Applications*, Sep 2017.

[39] H. Noura, S. Hussein, S. Martin, L. Boukhatem, and K. A. Agha. Erdia: An efficient and robust data integrity algorithm for mobile and wireless networks. In *2015 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2103–2108, March 2015.

[40] H. Noura, S. Martin, and K. A. Agha. Edca: Efficient diffusion cipher and authentication scheme for wireless sensor networks. In *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2635–2640, April 2014.

[41] H. Noura, S. Martin, and K. A. Agha. E3sn: Efficient security scheme for sensor networks. In *2013 International Conference on Security and Cryptography (SECRYPT)*, pages 1–7, July 2013.

[42] Yaser Jararweh, Mahmoud Al-Ayyoub, Maged Fakirah, Luay Alawneh, and Brij B Gupta. Improving the performance of the needleman-wunsch algorithm using parallelization and vectorization techniques. *Multimedia Tools and Applications*, pages 1–17, 2017.

[43] Khalid Mohamed Alajel, Wei Xiang, and John Leis. Error resilience performance evaluation of h. 264 i-frame and jpwl for wireless image transmission. In *Signal Processing and Communication Systems (ICSPCS), 2010 4th International Conference on*, pages 1–7. IEEE, 2010.

[44] Elaine B Barker and John Michael Kelsey. *Recommendation for random number generation using deterministic random bit generators (revised)*. US Department of Commerce, Technology Administration, National Institute of Standards and Technology, Computer Security Division, Information Technology Laboratory, 2011.

[45] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. john wiley & sons, 2007.

[46] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in Cryptology*, pages 10–18. Springer, 1985.

[47] Kaisa Nyberg and Lars Ramkilde Knudsen. Provable security against a differential attack. *Journal of Cryptology*, 8(1):27–37, 1995.

[48] Jung-Sik Cho, Sang-Soo Yeo, and Sung Kwon Kim. Securing against brute-force attack: A hash-based RFID mutual authentication protocol using a secret value. *Computer Communications*, 34(3):391–397, 2011.

[49] Shujiang Xu, Yinglong Wang, Jizhi Wang, and Min Tian. Cryptanalysis of two chaotic image encryption schemes based on permutation and xor operations. In *Computational Intelligence and Security, 2008. CIS'08. International Conference on*, volume 2, pages 433–437. IEEE, 2008.

[50] Guoji Zhang and Qing Liu. A novel image encryption method based on total shuffling scheme. *Optics Communications*, 284(12):2775–2780, 2011.

[51] Benyamin Norouzi, Seyed Mohammad Seyedzadeh, Sattar Mirzakuchaki, and Mohammad Reza Mosavi. A novel image encryption based on hash function with only two-round diffusion process. *Multimedia systems*, 20(1):45–64, 2014.

[52] Rhouma Rhouma and Safya Belghith. Cryptanalysis of a new image encryption algorithm based on hyper-chaos. *Physics Letters A*, 372(38):5973–5978, 2008.

[53] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of PSNR in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008.

[54] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, 2004.

[55] A Akhshani, S Behnia, A Akhavan, H Abu Hassan, and Z Hassan. A novel scheme for image encryption based on 2D piecewise chaotic maps. *Optics Communications*, 283(17):3259–3266, 2010.

[56] Anil Kumar and MK Ghose. Extended substitution-diffusion based image cipher using chaotic standard map. *Communications in Nonlinear Science and Numerical Simulation*, 16(1):372–382, 2011.

[57] Xiaojun Tong, Minggen Cui, and Zhu Wang. A new feedback image encryption scheme based on perturbation with dynamical compound chaotic sequence cipher generator. *Optics Communications*, 282(14):2722–2728, 2009.

[58] Zeinab Fawaz, Hassan Noura, and Ahmed Mostefaoui. An efficient and secure cipher scheme for images confidentiality preservation. *Signal Processing: Image Communication*, 42:90–108, 2016.