Multimed Tools Appl (2018) 77:15457–15484
DOI 10.1007/s11042-017-5124-9

# A new efficient lightweight and secure image cipher scheme

Hassan Noura[1] · Lama Sleem[2] · Mohamad Noura[2] ·
Mohammad M. Mansour[1] · Ali Chehab[1] ·
Raphaël Couturier[2]

Received: 16 February 2017 / Revised: 18 July 2017 / Accepted: 16 August 2017 /
Published online: 4 September 2017

**Abstract** The protection of multimedia content has become a key area of research, since very often a user's privacy and confidentiality can be at risk. Although a large number of image encryption algorithms have recently emerged, only a subset of these algorithms are suitable for real applications. These algorithms however use non-integer operations such as chaotic solutions that introduce a sizeable overhead in terms of latency and resources, in addition to floating-point hardware that is costly to implement. Designing an efficient, lightweight, and secure image encryption algorithm is still a hard challenge; yet, it is crucial to have in order to meet the demands of recent multimedia applications running on energy-limited devices. In this paper, an efficient image encryption scheme based on a dynamic structure is proposed. The structure of the proposed cipher consists of two different lightweight rounds (forward and backward chaining blocks) and a block permutation process. In addition, a key derivation function is proposed to produce a dynamic key based on a secret key and a nonce. This key, according to its configuration, can be changed for each validate time (session) or for each new input image. Then, based on this key, the cipher layers are produced, which are an integer or a binary diffusion matrix and a substitution table S-box, together with a permutation table P-box. The proposed dynamic cipher is designed to provide high robustness against contemporary powerful attacks, and permits reducing the required number of rounds for achieving the lightweight property. Experimental simulations demonstrate the efficiency and robustness levels of the proposed scheme.

**Keywords** Lightweight image encryption · Substitution process · Blocks permutation · Key dependent integer and binary diffusion matrix · Forward-backward CBC operation mode

✉ Raphaël Couturier
raphael.couturier@univ-fcomte.fr

[1] Electrical and Computer Engineering, American University of Beirut, Beirut, Lebanon

[2] FEMTO-ST Institute, Université Bourgogne Franche-Comté (UBFC), Belfort, France

# 1 Introduction

Living in an era of technology and communication, many researchers have to focus their attention on security issues. More and more people use the Internet and social media applications. Their personal data is placed on a world wide network at the risk of seeing their privacy invaded. Protecting our personal data has become of major importance and researchers are competing to build different algorithms that can guarantee the safety of our images in particular. Securing images can take place on two levels: a pixel level and a compression level. In this paper, we worked on a pixel level that ensures a higher level of security since every pixel is being processed and encrypted. While on the compression level, pixels are being nominated and chosen if they meet a specific predefined condition and are then subjected to encryption.

In fact, recently, a set of compression algorithms for encrypted images has been presented to enable the reduction of the size of the encrypted image such as [37–39]. This means that encrypting the image in pixel level is preferable ensuring a full protection. Employing the recent compression algorithms over encrypted images permits to reduce the overhead in terms of storage and communication. Based on these works, the proposed solution is presented and realized at the pixel level.

## 1.1 Problem

In fact, the existing symmetric-key standard encryption algorithms such as DES (Data Encryption Standard) [4] and AES (Advanced Encryption Standard) [8, 10, 32] are used for image encryption. However, the traditional techniques are defined for texts and are not suitable for multimedia contents since images and texts possess different characteristics. The intrinsic features of traditional ciphers such as AES or DES are based on applying a round function for multi-times, which requires higher execution time and more resources. This number of rounds is higher and depends on the size of the employed secret key to prevent differential and linear attacks. The reason that $r$ is relatively high is that these algorithms have a static structure and consequently substitution and diffusion layers are known and public. This has a negative impact in terms of execution time and resources, which is a critical problem in the real-time delivery of multimedia streams [9] or for tiny devices that cannot handle high latency.

## 1.2 Related works

Therefore, redesigning the existing symmetric cryptographic algorithms is still in progress. Several recent approaches are presented which follow the dynamic structure which allows the number of rounds $r$ to be reduced and makes the cipher's substitution and diffusion processes variable and unknown to the attackers. One of these recent approaches is the chaotic one which consists of a non-linear dynamic system that apparently looks random. Because of its extreme sensitivity to intrinsic conditions, chaos was implemented extensively to build the cryptographic algorithms of digital images as in [1, 5, 7, 11, 15, 17, 28, 29, 31, 34]. Unfortunately, the majority of chaos-based encryption algorithms are not considered particularly secure, and many of them have been crypt-analyzed successfully as in [19, 20, 27, 36], because of the instability coming from the periodicity of mapping [14, 21] and the finite computing precision that makes the system defenseless against different kinds of attacks [2, 3].

Furthermore, other approaches have recently been introduced and are based on a hash function to ensure authentication, integrity and data confidentiality such as in [6, 33]. A hash function is essential because of its properties that prevent any retrieval of useful information about the secret key used. In [6], a hash function is presented and used to encrypt images. Digital images have been encrypted using the standard Secure Hash Algorithm (SHA-2) along with a compound forward transform and a password provided by the user. Another hash-key-based encryption scheme is discussed in [23], where the salsa20 hash function [13] is used to generate a dynamic secret key. Then, the resultant key is correlated later with the plain text image. Unfortunately, hash functions use floating calculation which makes the hardware implementation complicated and time consuming. Besides, using the pixels of the image itself will raise the probability of the error propagation. In other words, any small error caused by noisy channels (like wireless channels) will prevent the receiver from decrypting the image, thus making this system weak and intolerant to any small error.

To the best of our knowledge, a real, efficient and secure encryption algorithm is needed to meet the requirements of the modern multimedia applications and low cost devices. Therefore, an efficient and secure candidate that can ensure a high level of security with a low computational complexity and simple hardware implementation is necessary for recent applications such as multimedia IoT.

## 1.3 Contribution

The proposed approach ensures several contributions compared to the recent image encryption schemes towards attaining a high level of efficiency and security. These contributions are indicated in the following:

### *1.3.1 System performance*

- The proposed cipher is realized on the block level with a flexible size of blocks (n bytes) that can be adjusted according to the available memory, thus allowing it to be realized with tiny limited devices.
- It is simple to implement in hardware or software solutions which is an important advantage to be considered in real implementations.
- The required processes are reduced for each round iteration from two (mainly substitution then diffusion) to only one process compared to the majority of encryption schemes. In fact, the diffusion operation is applied in the first round, while the substitution operation is realized in the second round. This permits to reduce the required latency for each round.
- A binary diffusion operation is proposed, in addition to an integer one. More important, the binary diffusion permits to reduce the execution time according to the obtained results (see page 13) from 17 to 45% in function of $n$ (dimension of the diffusion matrix).
- Efficient key dependent cryptographic primitives (a substitution table, a diffusion matrix, and a permutation table) are built to ensure a considerable cryptographic performance and can prove to be a real improvement in time and simplicity. This will reduce the required time needed to build these cipher layers and will simplify the hardware implementation. This is essential, since each primitive of these three has its effect and its role in making the proposed cipher scheme secure and efficient.

- In addition, the proposed algorithm ensures the avalanche effect with an acceptable trade-off with error propagation.

### *1.3.2 Security performance*

- The proposed cipher scheme presents an efficient collaboration between substitution, diffusion and block permutation towards ensuring a high level of security and efficiency.
- A block permutation operation is introduced to randomize the sequential order of chained blocks. This operation permits to complexify the procedure of possible future attacks and consequently ensures a better security level compared to the existing cipher approaches that preserve the encryption sequential order. In addition, this step requires lower latency overhead and will not degrade the previous performance contributions.
- The dynamic key approach is employed and the key can be changed for each fixed/chosen time (defined by an application or a user) or for each input image which will render the cryptanalysis task unfeasible. The attacker's task becomes more difficult because of the sensitivity of the unpredicted dynamic key especially if this dynamic key is changed for each input image. This will permit to ensure a high level of security against existing and modern powerful attacks.

As a conclusion, the obtained results, in terms of performance and security, prove that the proposed approach is efficient and can ensure a high level of security compared to other recent image encryption algorithms that have static or dynamic structures. Therefore, the proposed cipher can be considered as a relevant cryptographic candidate since it ensures a positive balance between system performance and security level.

### 1.4 Organization

The paper will be organized as follows: Section 2 shows the derivation of the special parameters needed for encryption. Then, in Section 3 the proposed encryption and decryption schemes are explained to show how the cipher works. In Section 4, the main cipher layers of encryption are detailed. They rely on key dependent substitution layer, diffusion, and permutation. Then, the algorithm is tested under extensive performance evaluation and security analysis in Section 5 to prove the robustness and efficiency of this proposal. In Section 6, the security of this algorithm is discussed. Finally a conclusion summarizes the work and future works are mentioned in Section 7.

## 2 Key derivation

In this section, the proposed key derivation function and its corresponding sub-keys generation scheme are presented and illustrated in Fig. 1. In fact, the cipher layers are dynamic and are changed according to this set of sub-keys. The input of this step is a secret key $K$ and a NONCE $N_o$ (different for each input image or a fixed time) that are Xored to produce a dynamic key $DK$. To ensure that a different $DK$ is produced for each different input image, the SHA-512 cryptographic hash function is chosen and it is known that it has a high degree of collision. This will introduce better strength against any powerful attacks. This dynamic key $DK$ is the basis to form the other required sub-keys as explained in the following.

- **Secret Key** $K$: It is a Secret Key shared only between both entities (transmitter and receiver). To provide better security, the symmetric secret key can be renewed after

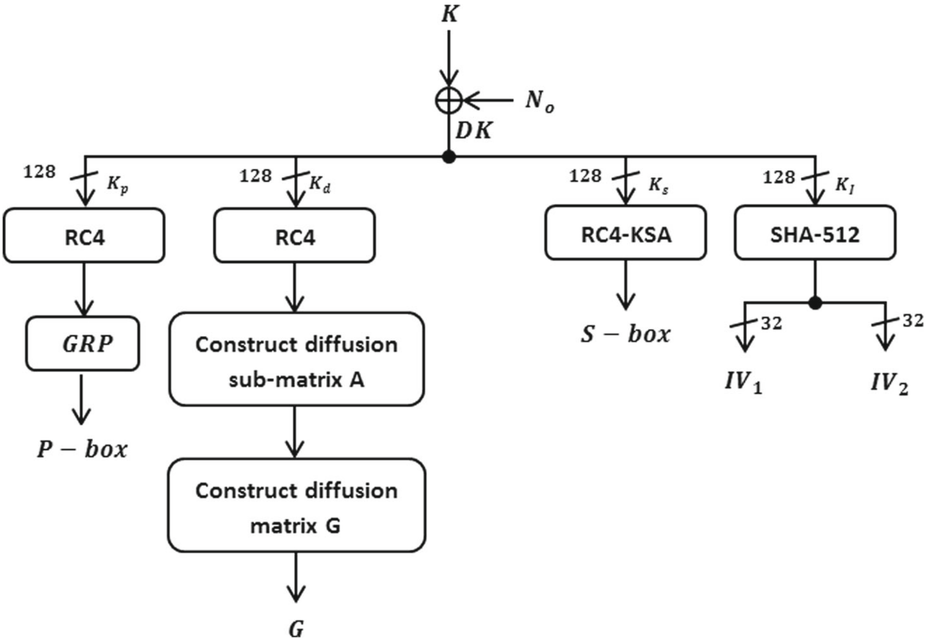


**Fig. 1** Dynamic Key and the requires sub-keys generation scheme. These sub-keys are employed to form the cipher layer's

each periodic interval that is chosen depending on the application. Elliptic curve Diffie Hellman (ECDH) protocols can be a good candidate for this task.

- **Nonce** $N_o$: This Nonce can be produced by a pseudo random generator. Each Nonce should be used only once and should be different for every input image (or session time). Two possible techniques of generation of Nonce can be proposed: i) generated by the emitter and transmitted in an encrypted form to the receiver by employing the secret key or by employing the receiver public key. ii) The second method is to produce the Nonce at the emitter and receiver in a synchronized manner between both entities.
- **Dynamic Key** $DK$: The secret key $K$ will be Xored with $N_o$ and its corresponding output is hashed by employing SHA-512 to produce the dynamic key $DK$, which corresponds to the MAC value and it has a 512 bit length. After this, $DK$ key is split into four sub-keys $\{K_p,\ K_d,\ K_s,\ K_i\}$ in a way that each one of them has 16 bytes (128 bit) length. These sub-keys are described in the following.
- **Permutation sub-key** $K_p$: $K_p$ represents the first most significant 16 bytes of $DK$ and it is used to produce a permutation table ($P - box$) that is employed during the blocks permutation process. In this solution, any key dependent permutation generation algorithm can be employed. The selection of $GRP$ permutation algorithm [30] is done according to its simplicity in hardware or in software implementation. To ensure that a P-box has a good cryptographic performance, $GRP$ should be iterated for multi-iteration $rp$ and for each iteration a control parameter register is required according to [25]. This logic continues in this paper and $K_p$ is used as a seed for the famous stream cipher RC4 to generate the required $rp$ control parameter registers. Furthermore, $RC4$ is not used here in the context of a stream cipher which mixes the key stream with the input plain-text. In fact, this is different since RC4 is iterated with a dynamic sub-key.

This will not introduce any weakness and will not degrade the security level. Indeed, RC4 is selected here since it has the simplest hardware and software implementation and, it is the fastest one. RC4 should be iterated for $\lceil\frac{rp\times nb}{8}\rceil$ times to form the binary control parameters.

- **Diffusion sub-key** $K_d$: $K_d$ represents the second 16 most significant bytes. $K_d$ is used also 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 length, where $q$ represents the precision in bits. $q$ is equal to 1 in the case of a binary diffusion matrix and 8 in case of the byte integer matrix. In addition, the diffusion is realized at the block level and each block has $n$ bytes. This means that $n$ is the dimension of the square diffusion matrix (number of bytes in a block). Therefore, RC4 should be iterated for $nt = \frac{lv}{8}$ times. After that, $VA$ is reshaped into a square sub-matrix called $A$ with size $\frac{n}{2}\times\frac{n}{2}$, which is necessary to form the square diffusion matrix $G$. Moreover, $G$ can be binary or integer. In order to reduce the complexity, a binary diffusion layer is more recommended than using other layers of diffusion such as Matrix Distance Separate (MDS) as in AES or hill cipher (invertible integer square matrix). More details are given in Section 4.
- **Substitution sub-key** $K_s$ :$k_s$ represents the third set of the 16 most significant bytes and it is employed to produce a dynamic substitution table (S-box). The proposed cipher can employ any key dependent substitution generation table algorithm. In fact, in this paper, a simple technique is used and it is based on the Key Setup Algorithm (KSA) of RC4, which represents its initialization step. The output of KSA is a substitution table that is employed as a dynamic S-box in the proposed approach.
- **Initialization key** ($K_I$): $K_I$ represents the first 16 least significant bytes of $DK$ and it is hashed by employing $SHA-512$ [22], which is an un-keyed cryptographic hash function. The output hash value has 64 bytes length and it is divided into two equal parts, each one has a 32 bytes length. These two parts are considered as two initialization vectors ($IV_1$, $IV_2$) that are employed in the forward and backward chaining modes according to the proposed approach.

All the notations are shown in Table 1. These steps are enough to preserve high sensitivity since a little change in the dynamic key will lead to completely different parameters in the encryption process and this is proven in Section 5. The derivation of these parameters is illustrated in Fig. 1.

## 3 The proposed cipher algorithm

An input image (matrix form) is introduced to the proposed cipher and its size is $r\times c\times p$, where $r$ is the number of rows, $c$ is the number of columns and $p$ is the number of planes (in gray scale $P$=1). The number of blocks of the image is padded if necessary to be divisible into $nb$ complete blocks and each block consists of $n$ bytes. In the rest of the paper, $n$ will be set to 16 and can be changed according to the needed application and the memory space. The total number of blocks is $nb$ where $nb = \lceil\frac{r\times c\times p}{n}\rceil$.

### 3.1 Encryption algorithm

The steps of the encryption algorithm are described briefly in this section and its structure is illustrated in Fig. 2.

**Table 1** Table of notations

| Notation | Definition |
|---|---|
| $K$ | Secret key |
| $N_o$ | Nonce |
| $DK$ | Dynamic Key |
| $K_p$ | Permutation sub-key |
| $K_d$ | Diffusion sub-key |
| $K_s$ | Substitution sub-key |
| $K_I$ | Initialization vectors sub-key |
| $S - box$ | A dynamic produced substitution table |
| $S - box^{-1}$ | The inverse corresponding substitution table |
| $P - box$ | A dynamic produced permutation box |
| $P - box^{-1}$ | 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 planes of the input image |
| $n$ | Number of bytes in one image block ($n = 4, 8, \ldots, 32$) |
| $nb$ | Number of blocks in corresponding image |
| $B_i$ | The $i^{th}$ block of plain image |
| $C_i$ | The $i^{th}$ block of encrypted image (without blocks permutation) |
| $q$ | $q$=2 for binary Galois field |
| | and its equal to 8, 16, 32, 64 for integer Galois field |

The proposed scheme is based on a secret symmetric key that is shared between the entities of the communication system. This secret key is employed to produce a set of dynamic keys and each one is employed for one or for a set of input images.

The proposed cipher is divided into **two rounds** and a final block permutation process. Moreover, a **Forward** and a **Backward Chaining CBC** mode are employed to reach the required cryptographic performance such as the avalanche effect in the whole image. In the first round, a linear binary diffusion process is done by using the previously mentioned diffusion matrix $G$ (integer of binary). Whereas in the second round, a non-linear substitution operation and a global permutation are applied. To ensure high randomness and to increase the robustness of the system, pseudo-random chaining is added in both processes. Finally, global permutation will randomize the sequence of blocks as a final step and remove any relation between successive blocks. In the following, the forward and backward processes are explained.

### *3.1.1 Forward process*

It is the first step of the encryption algorithm, and it starts from the block $B_1$ that should be Xored with $IV_1$. Then, its corresponding output will be subjected to a diffusion operation

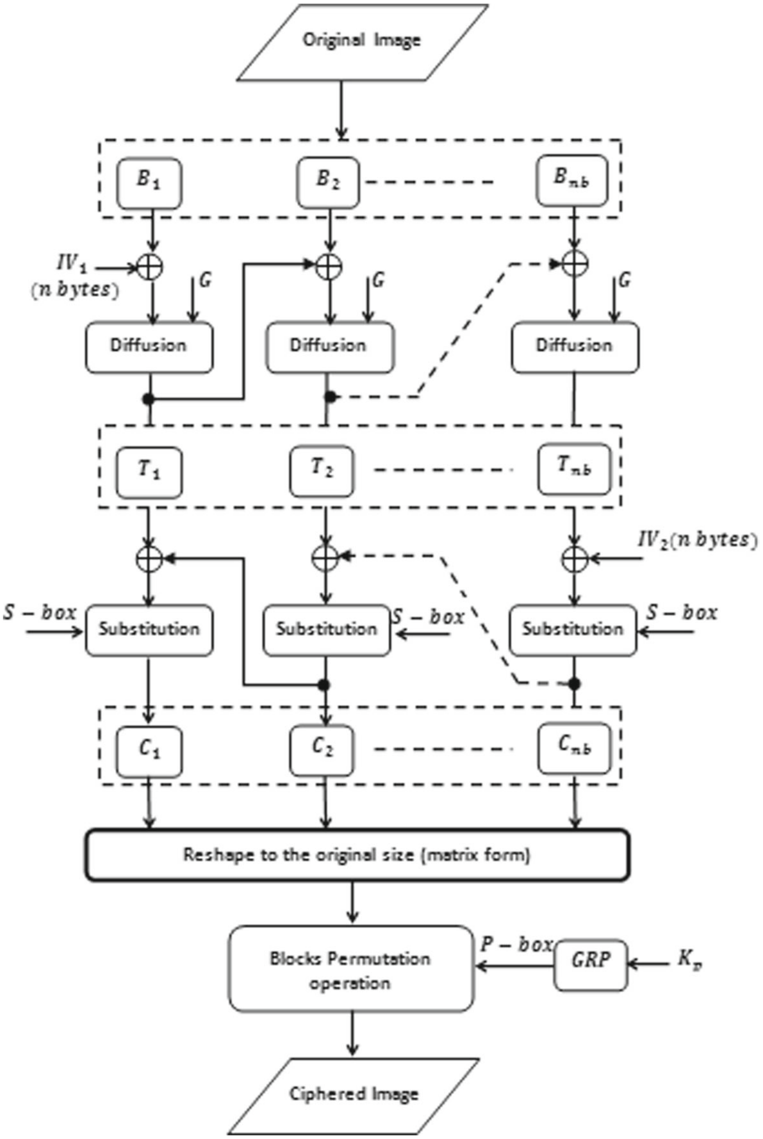


**Fig. 2** Scheme of the proposed lightweight cipher algorithm.

that can be based on a binary or integer diffusion matrix $G$. The result will be $T_1$ and the following equation summarizes this step for $i = \{1,\ 2,\ 3,\ \ldots,\ nb\}$:

$$T_i = G \odot (B_i \oplus T_{i-1}) \tag{1}$$

Where, $T_0 = IV_1$, and $\odot$ represents the diffusion operation that can be represented as an integer multiplication matrix in case of the integer diffusion and binary matrix mixing as described in Algorithm 1 for the binary diffusion operation. In this step, the diffused blocks are chained in a forward way and the process continues till the final block $B_{nb}$ is reached.

### 3.1.2 Backward process

The backward process starts after finishing the forward process. Starting from the last diffused block $T_{nb}$, it is Xored with $IV_2$ that represents the second initial vector. Then, its corresponding output will be subjected to a substitution operation that is based on the produced substitution table S-box. The result will be $C_{nb}$ and the following equation summarizes this step for $i = \{nb,\ nb-1,\ \ldots,\ 2,\ 1\}$:

$$C_i = S \odot (T_i \oplus C_{i+1}) \tag{2}$$

Where $C_{nb+1} = IV_2$, and $S$ represents the produced dynamic S-box. Then, the following block will be subjected to the same process, but with a difference that $IV_2$ will be replaced by $IV_1$ and with a reverse order. This means that in this step, the diffused blocks are chained in a backward way and the process continues until the first diffused block $T_1$ is reached.

### 3.1.3 Blocks permutation

After that, the output blocks $\{C_1,\ C_2,\ \ldots,\ C_{nb}\}$ are subjected to a final block permutation operation. The permutation process is introduced to add randomness and to eliminate the sequential relation order among the diffused substituted blocks. Finally, permuted blocks will be reshaped into the original size of the matrix to form the encrypted image.

## 3.2 Decryption

Similarly to the encryption scheme, the decryption scheme consists of an inverse block permutation process using the inverse $P - box$ called $P - box^{-1}$. In addition, the corresponding two rounds are applied in a reverse order by using the corresponding inverse dynamic substitution table S-box $S - box^{-1}$ and the inverse diffusion matrix ($G^{-1}$).

As a conclusion to this section, this cipher is designed to ensure the main properties that a strong secure cipher must reach. The confusion and diffusion properties are achieved with a lower round number and fast execution time and a high level of security as validated in the coming sections.

# 4 Proposed cipher layers

According to the theorem of Shannon, a successful cipher should achieve the confusion and diffusion properties. Mainly, the majority of the existing standard cryptographic algorithms is based on static keys. However, until now, no standard block cipher based on the dynamic approach is proposed and standardized. In fact, the benefits of the dynamic structure are important and we believe that the modern structure of cryptographic algorithm should be dynamic. Therefore, we follow this logic and this paper is a first step towards our goal. Thus, the proposed cipher is based on a dynamic key and it is designed to be a primary step towards standardizing it as a dynamic block cipher. The objective of this paper is to explain that it is possible to define a cipher with a high security level and with a lower computational complexity.

The dynamicity is reached since all the cipher operations (layers) are variable and depend on the dynamic key as the substitution, diffusion and permutation. The dynamic key can be changed for every input image which ensures the maximum level of security. Another configuration can be realized using $DK$ for a fixed time that means for a set of input images.

This means that for a specific session time, $DK$ is not changed. In both scenarios, the dynamicity prevents any powerful attack or prevents getting any useful information from the collected plain and cipher images.

### 4.1 Proposed diffusion technique

Mainly, the diffusion operation, whether integer or binary, requires an initial step which is the construction of the invertible diffusion matrix $G$. In this paper, an integer and binary technique to build a diffusion matrix $G$ is proposed. The main algebraic rule for a successful diffusion matrix $G$ is to be an invertible matrix with a determinant equal to 1 and has to have full rank.

$G$ is based on an invertible and bijective 2D matrix form. It is represented as follows:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad , det(A) = a \times d - b \times c \tag{3}$$

Then, if $det(A) = 1 \Rightarrow a \times d = 1 + b \times c$. To obtain the proposed structure of invertible key dependent diffusion matrix, we consider that $d$ should be equal to $a$, which leads to

$$a^2 = 1 + bc \Rightarrow bc = a^2 - 1 = (a-1)(a+1).$$

This results in $b$ and $c$ being equal to $(a + 1)$ and $(a - 1)$, respectively. Then, the form of the secret matrix will become in 2D as follows:

$$G = \begin{pmatrix} a & a+1 \\ a-1 & a \end{pmatrix} \tag{4}$$

In the following, the dimension of diffusion matrix can be extended to $n$ dimension for integer and then for binary one, which is a special case of the integer one. In fact, the benefit of the binary diffusion operation compared to the integer one is its simplicity in implementation in hardware or software in addition to a lower execution time.

#### *4.1.1 Integer diffusion matrix form*

It is obvious that only one parameter $a$ is needed to build $G$. To form the diffusion matrix $G$ with $n$ dimension, $a$ is replaced by a sub-matrix $A$ with size $\frac{n}{2} \times \frac{n}{2}$ as presented in the following equation:

$$G = \begin{pmatrix} A & A+I \\ A-I & A \end{pmatrix} \tag{5}$$

Where $I$ is the identity matrix and $A$ is a non-zero square matrix of size $\frac{n}{2}$. The elements of $A$ can be freely chosen from any Galois field ( Binary or integer) such that $G$ is full rank.

To validate that the proposed diffusion matrix $G$ is invertible, the following demonstration is presented.

Suppose that $G$ is built from four sub-matrices $(A,\ B,\ C,\ D)$ as shown in:

$$G = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \tag{6}$$

Its determinant is given by:

$$\begin{aligned}
det(G) &= det(A) \times det(D - CA^{-1}B) \\
&= det(A) \times det(D - CBA^{-1}) \\
&= det(A) \times det(A - A^2A^{-1} + I^2 \times A^{-1}) \\
&= det(A) \times det(A - A + A^{-1}) \\
&= det(A) \times det(A^{-1}) \\
&= det(A \times A^{-1}) \\
&= det(I) \\
&= 1
\end{aligned} \tag{7}$$

where $C \times B = (A^2 - I^2)$, $A^2 \times A^{-1} = A$ and $A^{-1}I^2 = A^{-1}$. Thus, the basic condition is verified and an inverse matrix $G^{-1}$ can be easily calculated at the receiver side according to the following equation.

$$G = \begin{pmatrix} A & -(A+I) \\ -(A-I) & A \end{pmatrix} \tag{8}$$

### 4.1.2 Binary diffusion matrix form

The diffusion process in the binary Galois Field requires to employ a binary diffusion matrix $G$ which enables to reduce the required computational complexity and can consequently reduce the execution time. Here, the scheme of creation of the invertible pseudo-random binary $G$ is described. To obtain the diffusion binary matrix form, the sub-matrix $A$ should be binary and we need to replace the operation of addition and subtraction with the logical operation Xor. Then, the form of the binary matrix is presented below:

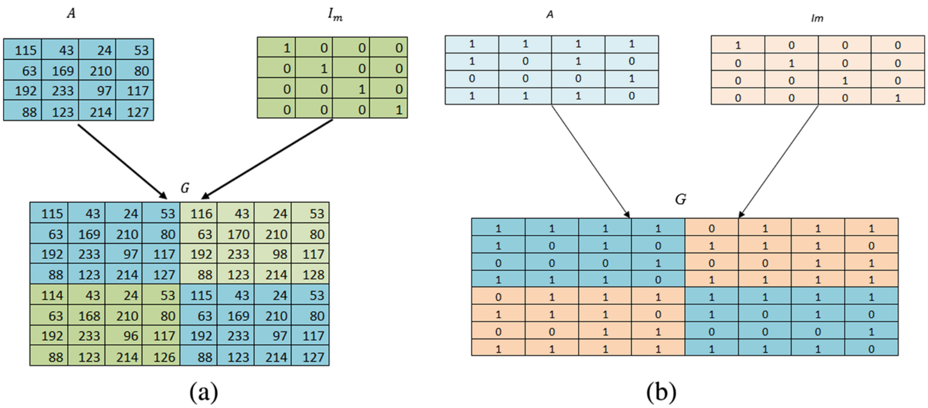
$$G = G^{-1} = \begin{pmatrix} A & A \oplus I_m \\ A \oplus I_m & A \end{pmatrix} \tag{9}$$

According to [18], the calculation of the inverse binary matrix $G$ is possible while the determinant of the binary matrix is 1 which is reached with the defined binary diffusion matrix form. Moreover, the proposed form has another advantage which is $G^{-1}$ is equal to $G$, so the inverse matrix operation at the receiver side is not required and the same required latency of diffusion is close to the inverse diffusion one.

In Fig. 3, two examples of the different steps required to build the secret diffusion matrix $G$ (Integer and Binary) for $n = 8$ are presented. In Fig. 4, a visual integer matrix $G$ (a) and a binary one (b) are shown respectively for $n = 32$. While for the visual binary $G$, the red color means that the index is equal to zero and 1 otherwise. As seen in Fig. 5, a block can be diffused in two different ways: (a) using the integer diffusion operation that employs arithmetic operations such as addition and multiplication mod $2^q$, and (b) using the binary diffusion operation that requires only the logical operation XOR.

### 4.1.3 Binary diffusion matrix mixing operation

The input of the diffusion operation is $n$ bytes. The diffusion process is performed on a series of $n$ bytes $B = \{B_1, \ B_2, \ \ldots, \ B_n\}$, and its output is the diffused block $X =$



(a)

A

| | | | |
|---|---|---|---|
| 115 | 43 | 24 | 53 |
| 63 | 169 | 210 | 80 |
| 192 | 233 | 97 | 117 |
| 88 | 123 | 214 | 127 |

$I_m$

| | | | |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 |

G

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 115 | 43 | 24 | 53 | 116 | 43 | 24 | 53 |
| 63 | 169 | 210 | 80 | 63 | 170 | 210 | 80 |
| 192 | 233 | 97 | 117 | 192 | 233 | 98 | 117 |
| 88 | 123 | 214 | 127 | 88 | 123 | 214 | 128 |
| 114 | 43 | 24 | 53 | 115 | 43 | 24 | 53 |
| 63 | 168 | 210 | 80 | 63 | 169 | 210 | 80 |
| 192 | 233 | 96 | 117 | 192 | 233 | 97 | 117 |
| 88 | 123 | 214 | 126 | 88 | 123 | 214 | 127 |

(b)

A

| | | | |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Im

| | | | |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 |

G

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

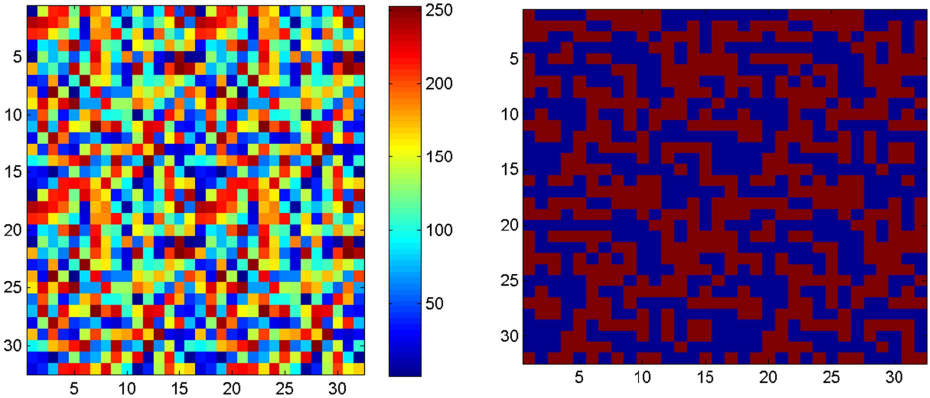
**Fig. 3** An example of constructed secret integer (**a**) and binary (**b**) matrices $G$ for $n = 8$

$\{X_1,\ X_2,\ \ldots,\ X_n\}$. The relationship among an input block $B$ of $n$ bytes, a binary diffusion matrix $G$ and the output block $X$ can be described as follows:

$$X = G \odot (B^t) \tag{10}$$

Where $B$ and $X$ are $n-$dimensional byte vectors and $^t$ is the transpose function. In addition, $G$ is a square $n \times n$ binary matrix and it consists of $n$ diffusion vectors $\{G_1,\ G_2,\ \ldots,\ G_n\}$. Each binary diffusion vector $G_i$ is represented as a sequence of independent random numbers from a binary Galois field, where $G_{i,j}$ is a binary diffusion coefficient of the line $i$ and column $j$ and $i, j = 1,\ 2,\ \ldots,\ n$ and can be equal to 0 or 1. This means that if $G_{i,j}$ is equal to 0, its corresponding byte $j$ is not introduced in the diffusion process of the $i^{th}$ diffused byte.

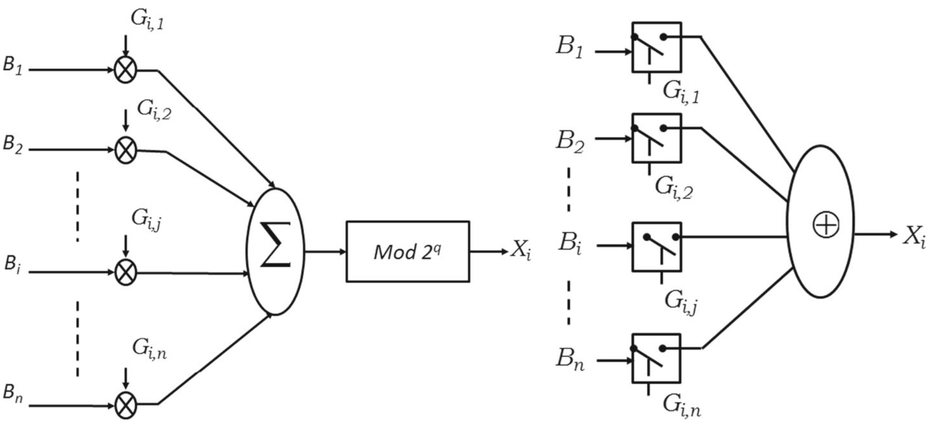
The values of the different indexes in each vector $(G_i)$ equal to 1 correspond to the byte introduced in the diffusion process. The diffused byte is the result of $m$ xoring bytes, where



(a) Integer diffusion matrix

(b) Binary diffusion matrix

**Fig. 4** A visual example of a produced secret matrix $G$ in a binary field (**a**) and in an integer field (**b**) with a pseudo random sub-matrix $A$ for $n = 32$



(a) Integer diffusion operation

(b) Binary diffusion operation

**Fig. 5** Integer (**a**) and binary (**b**) diffusion operation to produce a diffused element $X_i$ by employing the $i^{th}$ vector $G_i$ of the diffusion matrix $G$

the corresponding index of its diffusion vector is 1. The following listing shows the mixing algorithm ($\odot$) in pseudo code in Algorithm 1.

**Algorithm 1** Integer & binary diffusion algorithms

```
procedure INTEGER_MIXING($Y$, $G_i$)
    $n \leftarrow length(Y)$
    for $i \leftarrow 1$ to $n$ do
        $z_i \leftarrow= 0$
        for $j \leftarrow 1$ to $n$ do
            $z_i \leftarrow (z_i + G_{i,j} \times y_j) mod\, 256$
        end for
    end for
    return $z = \{z_1, z_2, \ldots, z_n\}$
end procedure
```

```
procedure BINARY_MIXING($Y$, $G_i$)
    $n \leftarrow length(Y)$
    for $i \leftarrow 1$ to $n$ do
        $z_i \leftarrow= 0$
        for $j \leftarrow 1$ to $n$ do
            if $G_{i,j} \neq 0$ then
                $z_i \leftarrow z_i \oplus y_j$
            end if
        end for
    end for
    return $z = \{z_1, z_2, \ldots, z_n\}$
end procedure
```

The average of the execution time for the diffusion operation is done for 250000 iteration and for different sizes of block $n$ using “GCC” with optimization “O2” for the proposed integer and binary diffusion techniques. Therefore, the ratio between their execution times is shown in Fig. 6. The obtained results indicate that the binary approach can ensure a reduction in execution time compared to the integer one from to 17 to 45% in function of $n$. More important, increasing $n$ permits to reduce more and more the execution time. This proves that the proposed binary approach is significant since it permits to reduce the execution time and consequently less latency and resources can be achieved compared to the integer diffusion operation.

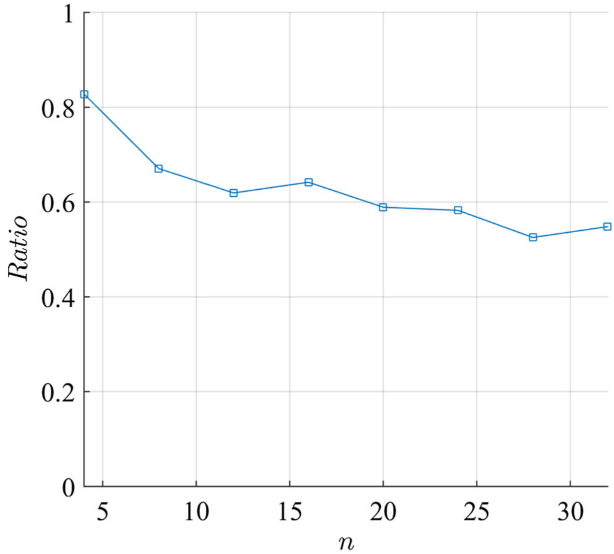


**Fig. 6** The ratio time analysis between the proposed binary diffusion approach and of the integer one in function of $n$ and for 25 0000 iterations

## 4.2 Key dependent substitution operation

An important property that should be ensured during the design of a cipher system is the confusion property which is ensured by employing a substitution technique that is a non linear operation. This operation makes the algorithm immune against differential and linear attacks. The existing substitution techniques use a lookup table called an S-box or employ a nonlinear function.

In this paper, the selected technique is to use a dynamic S-box that can ensure a lower latency compared to other ones. The proposed algorithm of construction dynamic S-boxes is based on the KSA of RC4. Moreover, the KSA algorithm is stated and explained in Algorithm 2. The cryptographic performance (LPF (Linear Probability Approximation Function), SAC( Strict Avalanche Criterion), BIC (Output Bit Independence Criterion) and DPF (Differential Probability Approximation Function)) of the produced substitution table is quantified after using each element of the substitution key $K_s$. The corresponding results are shown in Fig. 7 and clearly indicate that a good performance has been reached attained such as lower LPF, lower DPF, SAC and BIC are close to the ideal value = 0.5. In addition, the produced S-box is bijective which consequently means that the inverse S-box, S-box$^{-1}$ can be obtained. Hence, it can be generated easily by the following operation S-box$^{-1}$[S-box(i)] = i. Indeed, these criteria are explained briefly in the following.

- LPF: Quantifies the nonlinear degree of a given substitution table and it must be as low as possible. The variation of LPF versus the size of sub-substitution in byte length is shown in Fig. 7a. LPF is so stable and close to the minimum value from the third iteration (LPF close to $2^{-4.8}$).
- DPF: Quantifies the differential uniformity of a given substitution table. It must reach its minimum value and in our case it reaches the minimum in a stable value after 3 iterations according to Fig. 7b.

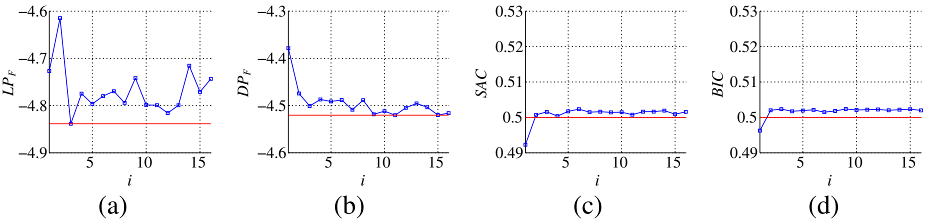


**Fig. 7** Variation of the average $LPF$ (**a**), $DPF$ (**b**), (**c**) SAC, (**d**) BIC versus the number of iterations $i$

- SAC: A S-box satisfies SAC whenever a single input bit is complemented, the output substituted bit should be changed at least with a probability of half. According to Fig. 7c, the SAC is attained and becomes close to its ideal value (0.5) from the third iteration.
- BIC: Specifies that two output bits $j$ and $k$ should be changed independently when a single input bit $i$ is changed. It can also be seen that BIC reached its desirable value 0.5 after two iterations according to Fig. 7d.

**Algorithm 2** KSA for RC4

```
procedure RC4_KSA($K = \{k_1, k_2, \ldots, k_L\}, L$)
    for $i \leftarrow 0$ to 255 do
        $S[i] \leftarrow i$
    end for
    $j \leftarrow 0$
    for $i \leftarrow 0$ to 255 do
        $j \leftarrow (j + S[i] + k[j \bmod L]) \mod 256$
        $swap(S[i], S[j])$
    end for
    return $S$
end procedure
```

According to all these criteria, the produced dynamic S-box achieves a good cryptographic performance from the third iteration. However, to ensure a high number of unique S-boxes, the size of the dynamic substitution sub-key $K_s$ is set to 16 bytes (16 iterations and for each iteration a byte is selected) in order to reach this objective.

### 4.3 Key dependent permutation layer

This step consists in applying a block permutation among the diffused substituted blocks. It is designed to remove and to randomize the order relation of blocks. The block permutation operation is performed by using the produced dynamic $P - box$ that has $nb$ length. This permutation table is built based on the GRP permutation algorithm that is described in [30].

GRP is chosen here since it is known to be simple and efficient in hardware and software implementations. The pseudo-code of GRP algorithm is shown in Algorithm 3. The basic idea of GRP is to split the indexes into two groups according to a pseudo random bit sequence which is $CR$. If the bit in $CR$ is 0, this index is placed in the first group. Otherwise, the element is placed in the second group. These three vectors have the same length, which will be equal to the number of blocks $nb$ of the image. For further details about the permutation construction and realization, we can refer to [24] that describes GRP with variable control registers.

**Algorithm 3** GRP permutation algorithm

```
procedure GRP($R\_src$, $CR$, $l$)
    $j \leftarrow 0$

    for $i \leftarrow 0$ to $l - 1$ do
        if $CR[i] == 0$ then
            $R\_dest[j++] \leftarrow R\_src[i]$
        end if
    end for
    ▷ After that, if the control register bit is one, put its corresponding index on the right
    for $i \leftarrow 0$ to $l - 1$ do
        if $CR[i] == 1$ then
            $R\_dest[j++] \leftarrow R\_src[i]$
        end if
    end for

    Return $R\_dest$
end procedure
```

## 5 Statistical analysis

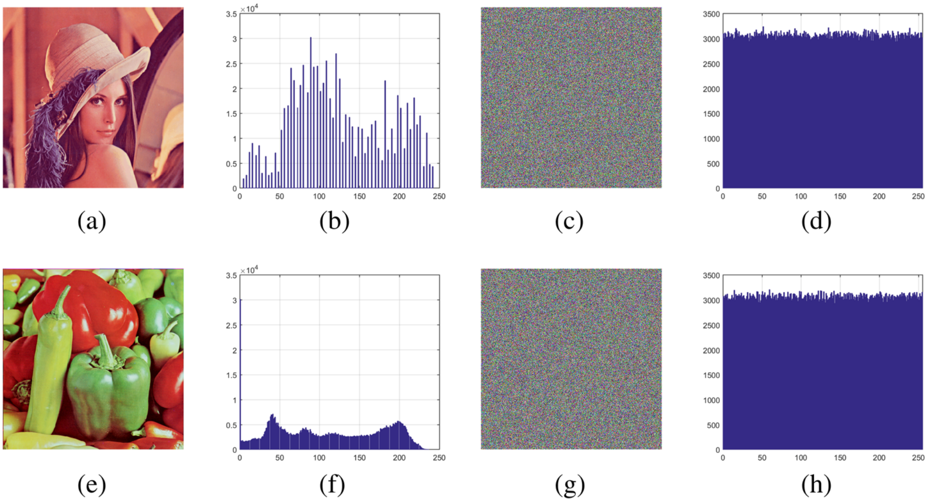
In order to have a strong immunity against statistical attacks, histogram and entropy analysis are performed to validate the uniformity property of the encrypted image. The correlation among the adjacent pixels of the plain and encrypted images in addition to the coefficient correlation between plain and encrypted images is also computed to check the independence property. These properties allow us to validate that the proposed cipher can ensure a high degree of randomness.

### 5.1 Histogram analysis

A uniform histogram for the encrypted image is necessary to ensure that this cipher meets the uniformity property. This means that each symbol has an occurrence frequency close to $\frac{r \times c \times p}{n}$, where $n$ is the number of symbols, $r$, $c$ and $p$ are respectively the number of rows, columns and plane of the input image. The histogram of two original plain-images ($512 \times 512 \times 3$) and their corresponding cipher images are shown in Fig. 8. It is shown that the histogram of the encrypted images is close to a uniform distribution (close to 3072).

### 5.2 Information entropy analysis

The information entropy of a data sequence $m$ is a parameter that measures the level of uncertainty [35]. Note that the entropy is expressed in $q$ bits and it quantifies the amount



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

(e) (f) (g) (h)

**Fig. 8** **a** Original Lena **a** and Pepper **e** and its corresponding histogram (**b**) and (**f**), respectively. Encrypted Lena (**c**) and Pepper (**g**) and its corresponding histogram (**d**) and (**h**), respectively

of information which can be coded by a compression algorithm. The information entropy is calculated according to the following equation:

$$H(m) = -\sum_{i=1}^{2^q} p(m_i) \log_2 \frac{1}{p(m_i)} \tag{11}$$

Where $p(m_i)$ represents the occurrence probability of the symbol $m_i$ and $2^q$ is the total number of symbols for $q$ bits. In this paper, towards quantifying the uniformity between adjacent pixels, the entropy is selected to be calculated on the level of sub-matrix.

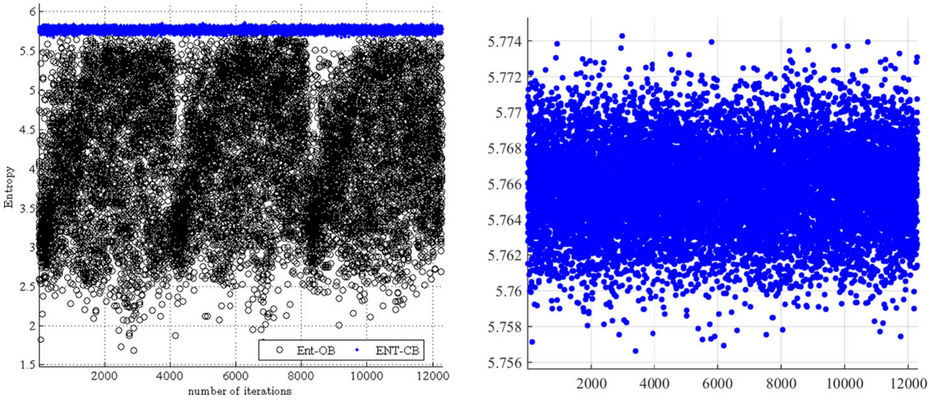
This test divides the image into a set of sub-matrices, where each one has a size $h \times h$. Each sub-matrix can be considered a truly random source with uniform distribution if it has an entropy equal or close to $\log_2(h^2)$ for $h^2 < 2^q$.

$$H(m) = -\sum_{i=1}^{h^2} \frac{1}{h^2} \log_2 \frac{1}{h^2} = log_2(h^2) \tag{12}$$

The entropy variation of plain and cipher sub-matrices of the Lena image under the usage of a random secret key and a Nonce for $h = 8$ is shown in Fig. 9. It is shown that the encrypted blocks always have an entropy close to the desired value 6 ($\log_2(8 \times 8) = \log_2(2^6) = 6$) in case $h = 8$. According to this, the proposed cipher ensures the uniformity and eliminates the redundancy between adjacent pixels.

## 5.3 Test correlation between original and cipher image

The high linear correlation between adjacent pixels must be removed after encryption which is a mandatory principal to resist statistical attacks [23, 26]. Having a correlation coefficient close to zero means that the cipher scheme ensures a high degree of randomness. The



(a) Ent-OB and ENT-CB

(b) Only ENT-CB

**Fig. 9** The Entropy analysis for the sub-matrices of encrypted Lena image under the usage of a random dynamic key for $h = 8$ and **b** a zoom of the entropy of the encrypted image. Ent-OB and ENT-CB are the entropy of the plain and cipher sub-matrices, respectively

correlation test is realized by randomly taking $N = 2,000$ pairs of two adjacent pixels of the defined known two plain images and their corresponding cipher images. Correlation is done in three directions: horizontal, vertical and diagonal. The correlation coefficient $r_{xy}$ is calculated using the following equations:

$$r_{xy} = \frac{cov(x, y)}{\sqrt{D(x) \times D(y)}} \tag{13}$$

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 both encrypted images (Lena and pepper) are presented in Fig. 10.

The obtained results indicate that the correlation coefficients of a plain image in horizontal, vertical and diagonal direction are higher and close to 1 where it is very low and close to 0 for the ciphered images. This shows that the proposed cipher scheme eliminates spatial redundancy.

## 5.4 Difference between plain and ciphered images

The encrypted image must be different by at least 50% from the original image in the bit level. Figure 11a shows the variation of the percentage of bit difference between the plain and cipher Lena images for 1,000 random dynamic keys. According to this result, the obtained value of the percentage difference is always close to 50%, which means that

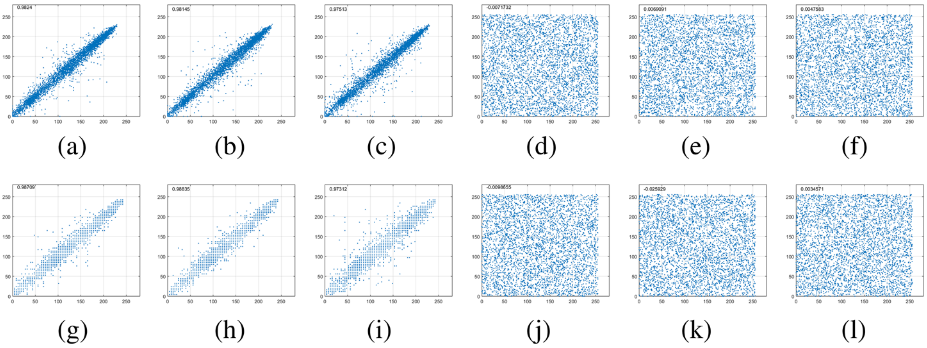


**Fig. 10** Correlation in adjacent pixels in original Lena: **a** horizontally, **b** vertically and **c** diagonally. Correlation in adjacent pixels in ciphered Lena: **d** horizontally, **e** vertically and **f** diagonally

the proposed cipher scheme ensures the independence. This test, in addition to the previous ones, validates that the proposed cipher ensures a high randomness degree.

### 5.5 Sensitivity test

Differential attacks are based on studying the relation between two encrypted images resulting from a slight change, usually one bit difference compared to the original one. A successful sensitivity test shows how much a slight change in the original-image or in the key will affect the resulted cipher image. The higher the encryption change, the better the sensitivity of the encryption algorithm is proven. There are two types of sensitivity: a **Plain Sensitivity** ($PS$) and a **Key Sensitivity** ($KS$). A key sensitivity shows how much the plain image is affected by a one bit change in the key. The result must be a change with at least 50% in the ciphered image. In Fig. 11b, the key sensitivity versus 1,000 random keys is always close to the optimal value (50% ). This indicates that any little change in the dynamic key will certainly affect the ciphered image.

Concerning plain-text Sensitivity, a dynamic key is used and the avalanche effect of the cipher is tested. One bit in two plain images is changed and it is clear that the images differ by at least 50%. Indeed, the obtained two ciphered images are totally different as illustrated in Fig. 11c where plain sensitivity was tested for 1,000 random dynamic keys. The cost of reducing the number of rounds and latency is given, since some values are far from 50% but still in an accepted manner. Therefore, the cipher has successfully met the avalanche effect and is considered to have enough sensitivity against any change in the plain image.

In addition to that, the sensitivity of the deciphered images is also tested. Changing one bit in the decrypted images must lead into a different image from the original one. The difference test between two decrypted images is done with the LSB bit of a chosen byte changed in the ciphered image. The result obtained is at least 50% different deciphered image. Then, the proposed algorithm has also a deciphering sensitivity. The result is shown in Fig. 11d. Note that in Fig. 11, all the results are obtained with a binary diffusion matrix. The results are similar using an integer diffusion matrix.

In fact, a statistical comparison between the proposed scheme with integer diffusion operation and the approach of [12] is presented in Table 2. Based on the obtained result, we can conclude that the proposed approach achieves a similar cryptographic performance compared to [12]. In addition, in Table 3, a statistical cryptographic performance of the

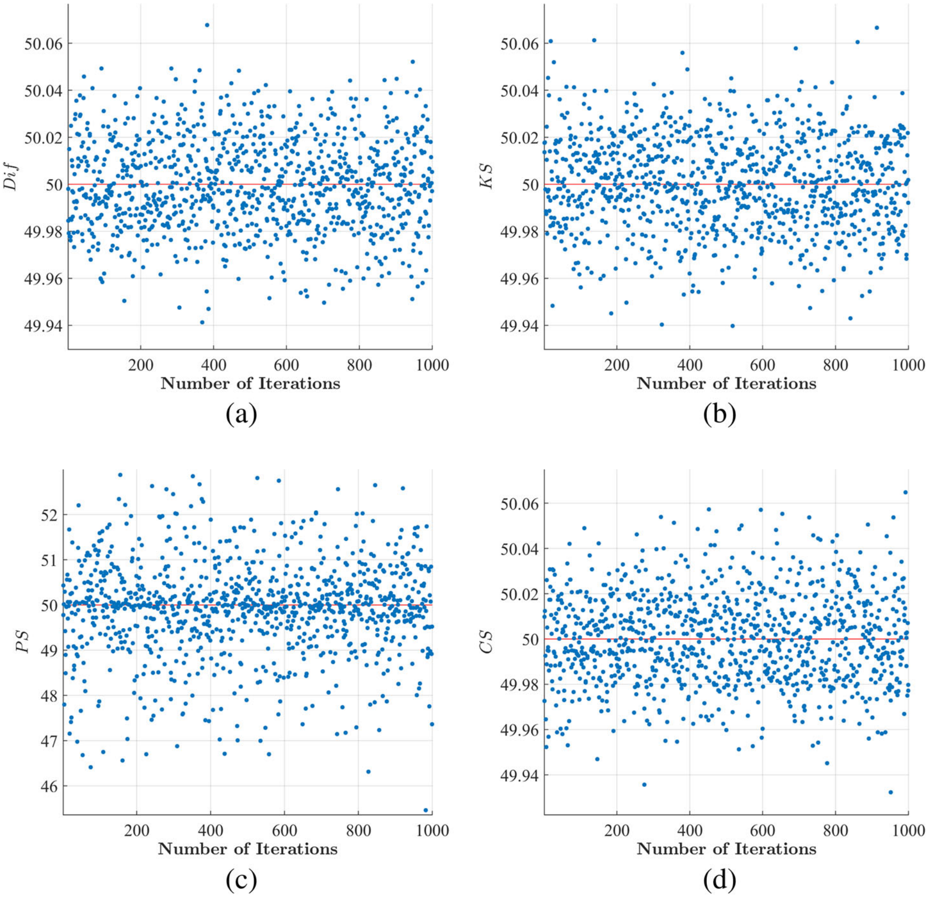


**Fig. 11** **a** Difference between plain Lena and ciphered ($Dif$), **b** Key Sensitivity ($KS$), **c** Plain Sensitivity ($PS$) and **d** Cipher Sensitivity (CS)

**Table 2** Statistical results of [12] and of the proposed scheme by using the integer diffusion matrix for 1000 random keys with Lena as plain image

| | Approach of [12] | | | | Proposed approach with "integer" diffusion matrices | | | |
|---|---|---|---|---|---|---|---|---|
| | Min | Mean | Max | Std | Min | Mean | Max | Std |
| $Dif$ | 49.8875 | 50.001 | 50.12 | 0.034 | 49.9414 | 49.9998 | 50.0677 | 0.0195 |
| $PS$ | 45.6381 | 48.0975 | 50.9305 | 1.9728 | 45.4594 | 49.9533 | 53.9893 | 1.0992 |
| $KS$ | 49.8741 | 49.9993 | 50.1143 | 0.0342 | 49.9398 | 50.0003 | 50.0666 | 0.0198 |
| $H-O$ | 2.1823 | 4.2014 | 5.7813 | 0.79914 | 2.1823 | 4.2014 | 5.7813 | 0.79914 |
| $H-E$ | 5.7623 | 5.7657 | 5.7701 | 0.0012 | 5.7566 | 5.7658 | 5.7743 | 0.0024 |
| PSNR | 9.1919 | 9.2306 | 9.2604 | 0.0096 | 8.5720 | 8.5896 | 8.6054 | 0.0054 |

**Table 3** Statistical results of the proposed scheme by using the proposed binary modification for 1000 random keys with Lena as plain image

| Proposed approach with "binary" diffusion matrices | | | | |
|---|---|---|---|---|
| | Min | Mean | Max | Std |
| $Dif$ | 49.941 | 50.0007 | 50.0623 | 0.0202 |
| $PS$ | 44.9621 | 50.041 | 55.131 | 1.185 |
| $KS$ | 49.9439 | 49.9992 | 50.0651 | 0.0199 |
| $H-O$ | 2.1823 | 4.2014 | 5.7813 | 0.79914 |
| $H-E$ | 5.756 | 5.7657 | 5.775 | 0.00239 |
| PSNR | 8.573 | 8.5894 | 8.6093 | 0.0054 |

proposed approach with a binary diffusion operation is presented. Also, the obtained results are similar to [12]. This indicates that the binary diffusion operation preserves the required cryptographic performance and consequently it can be safely employed in the proposed cipher.

### 5.6 Visual degradation

This test is specific for image and video contents and enables to quantify the visual degradation reached by employing the cipher scheme. In fact, the degradation operated on the encrypted image prevents the contents of an image from being recognized. To measure the visual degradation, a well known parameter is studied to measure the encryption visual quality, which is the Peak Signal-to-Noise Ratio (PSNR) [16].

Indeed, PSNR is derived from the Mean Squared Error (MSE), which represents the cumulative squared error between an original and encrypted image. A lower PSNR value indicates that there is a high difference between the original and the cipher images. In the proposed algorithm PSNR was measured between the original and the encrypted image for 1,000 random dynamic keys and is presented in Fig. 12. As shown, the mean PSNR value is 8.621 dB, which is close to the desirable value. This low value validates that the proposed encryption technique provides a high difference between the original and the encrypted image. As a conclusion, the proposed cipher scheme ensures a hard visual degradation so that no useful information can be revealed about the contents of the original image from the ciphered image.

### 5.7 Propagation of errors

Error propagation is an important criterion that should be low (error is not propagated) to consider the proposed cipher as efficient. Interference and noise existing in the transmission channel are the main causes of error. However, a bit error means that a substitution of '0' bit into '1' bit or vice versa will take place. In the proposed algorithm, if any block is affected in an image, it will only affect its previous and next neighboring blocks. This presents an acceptable error propagation. In fact, this is the cost of ensuring the avalanche effect in the whole image.

### 5.8 Execution time

The main motivation of this paper is to design a lightweight cipher scheme that can reach the required level of security, with a lower execution time, and respect the time limitations

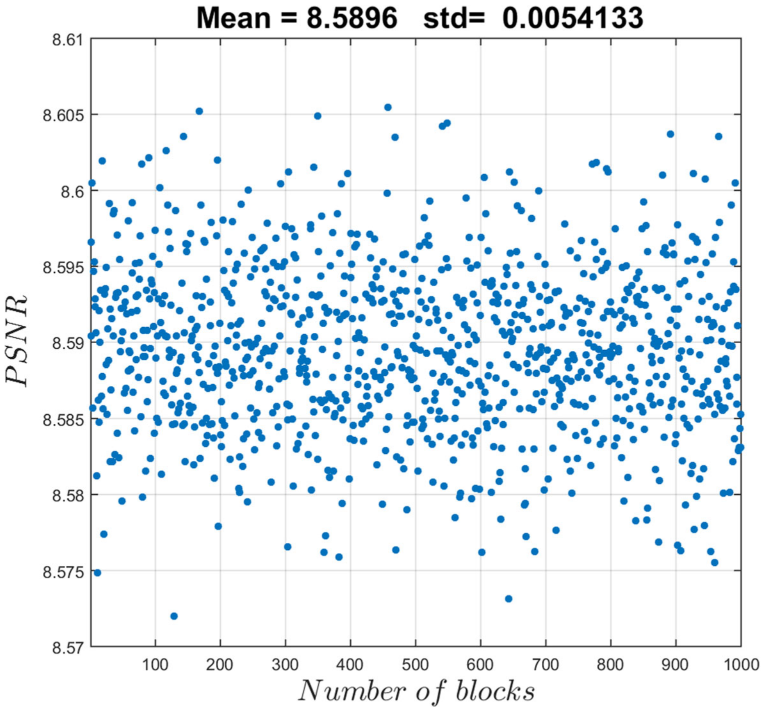


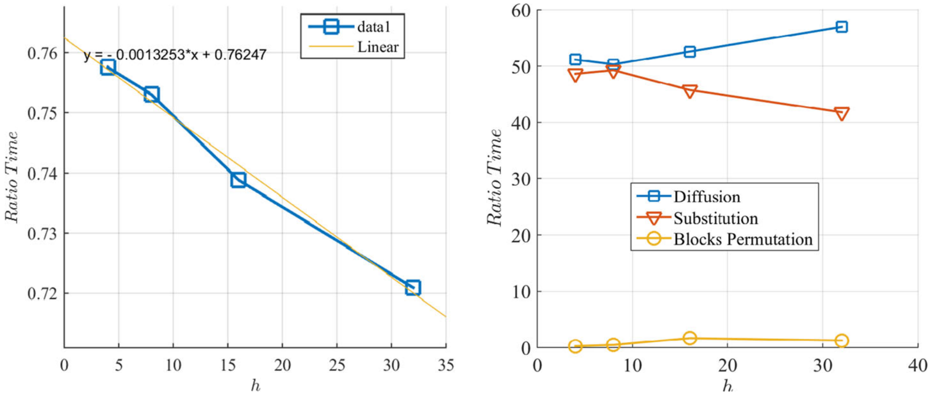
**Fig. 12** $PSNR$ variation between the original and the encrypted Lena image versus 1000 dynamic keys

of real time and reduce the required resources. In fact, a lower time of execution leads to a lower energy consumption of calculation, which is critical for limited devices that use a battery. The average calculation time (within 1,000 iterations) to encrypt the plain Lena image of size 256 × 256 × 3 is performed using the following software and hardware environment : Matlab R2013b simulator, micro-computer Intel Core i5, 3 GHZ CPU, 2 GB RAM Intel and the Microsoft Windows 7 operating system.

A comparison in execution time is done with the recent algorithm of [12], which was compared with recent cipher schemes and indicates that it can ensure a lower execution time compared to the selected recent state of the art. The average of the execution time for 1,000 iterations and for different sizes of block $n$ is obtained for the proposed approach (integer diffusion operation) and for [12]. Therefore, the ratio between the execution time of the proposed cipher and of [12] is shown in Fig. 13. In addition, the linear interpolation of the variation of ration versus $n$ is also calculated, and it is shown in Fig. 13-a and presented in the following equation:

$$Ratio = -0.0013253 \times n + 0.76247 \tag{14}$$

According to this result, the proposed approach ensures a reduction in execution time from to 22 to 26% in function of $n$ compared to [12]. In addition, increasing $n$ permits to reduce more and more the execution time. This proves that the proposed scheme requires a lower execution time and consequently less latency and resources compared to [12]. Consequently, the execution time of the proposed cipher is lower compared to [5, 15, 17, 23, 29, 31, 35, 40]. On the other hand, in Fig. 13b, the ratio of the execution time of each operation



(a) Ratio between the proposed approach and of [12]

(b) Ratio of the employed cipher operation

**Fig. 13** **a** the ratio time analysis between the proposed approach and the one in [12] in function of $n$ and for 1000 iterations and **b** the ratio of each operation (integer diffusion, substitution, and blocks permutation) of the proposed approach

of the proposed approach is presented. The obtained result indicates that the diffusion operation requires the higher percentage of execution time in function of $n$. Moreover, employing the binary diffusion operation will permit to reduce more the execution time according to Fig. 6. Therefore, and according to the previous results, the execution time is reduced to 50% for $n \geq 16$ by employing the binary diffusion operation compared to [12].

## 6 Discussion and cryptanalysis: resistance against the well-known types of attacks

All the previous tests were carried out to prove that the proposed scheme is efficient. In this section, a brief cryptanalysis discussion is presented to validate its secure employment and to prove that it can resist the different kinds of existing attacks. Uniformity was proven by getting the histograms of the ciphered images and by applying the entropy analysis. Then, independence between plain and cipher images is verified based on the correlation test among the adjacent pixels and between the original and cipher images in addition to measuring the difference in bit level between the plain and cipher images, which is close to 50%. Therefore, the encrypted images ensure a high degree of randomness and consequently provides it with a high resistance degree against statistical attacks.

The sensitivity of the plain image is accomplished with an acceptable value (always $> 45$) and the avalanche effect is reached in the whole image. A small reduction in terms of avalanche effect is introduced towards reducing the execution time. However, this will not degrade the security level since the lower value of plain image sensitivity is always greater than 45%. Therefore, any change in any bit of the plain image will provide a different deciphered image.This in turn will prevent chosen/known plain/cipher image attacks.

The sensitivity of secret and dynamic key in addition to a Nonce is attained and any change in any bit of these parameters provides a different cipher image on the emitter side or a different decrypted image on the receiver side with a 50% changed, which is a very satisfying result. These results validate that the proposed scheme can resist the related key attacks.

On the other hand, the size of the secret key can be 128, 196, and 256 bits as AES and the size of the dynamic key and Nonce are 512 bits, which are sufficient enough to protect the proposed cipher against the brute force attacks.

More important, the proposed dynamic key approach played a massive role in making this scheme more secure compared to the existing and modern powerful attacks. In addition, the variation of the secret key permits to overcome the problem of accident key disclosure.

This discussion will enable the proposed cipher to be considered as a good candidate for lightweight modern image encryption.

## 7 Conclusion

Our aim was to ensure a better security level and to meet the limitations of modern multimedia applications and low cost devices that are used today with limited resources. Therefore, the objective of this paper is to propose an efficient and robust lightweight cipher candidate that can ensure lower latency for the real time applications and to reduce the required resources for the low cost devices.

In contrast with the existing image encryption ciphers, the proposed scheme uses the substitution and diffusion only once but in two different rounds towards reducing the required latency and resources and without any degradation in the level of security.

In this paper, the proposed cipher is based on the dynamic key approach which permits to produce a dynamic key for each validate time (session) or for each input image (depending on the configuration). Then, based on this dynamic key, a set of sub-keys are obtained to produce the required dynamic initialization vectors, substitution and permutation tables in addition to a diffusion matrix, which are the basic elements of the proposed dynamic cipher scheme. Furthermore, the structure of the proposed cipher employs two different round functions using the CBC operation mode in forward and backward directions. After that, a block permutation operation is applied to randomize the sequential order of blocks. This operation permits to complicate the procedure of possible future attacks. Furthermore, the first round employs a dynamic diffusion operation that can be an integer or binary, while the second round uses a substitution operation. Therefore, both rounds permit to ensure the confusion and diffusion properties. In addition, the advantage of employing the CBC with forward and backward directions is to achieve the avalanche effect in the whole image.

Extensive tests were done to prove that the proposed candidate ensures the desired cryptographic performances. As a conclusion, due to its flexibility, high security as well as fast execution time achieved, it can be considered as a good cipher candidate. This cipher can compete with the available image cryptographic algorithms that can ensure image data confidentiality and privacy in real implementations for modern applications.

**Acknowledgements** This paper is partially funded from the Faculty of Engineering and Architecture at the American University of Beirut and from the Labex ACTION program (contract ANR-11-LABX-01-01).

## References

1. Akhshani A, Behnia S, Akhavan A, Abu Hassan H, Hassan Z (2010) A novel scheme for image encryption based on 2d piecewise chaotic maps. Opt Commun 283(17):3259–3266
2. Alvarez G, Li S (2009) Cryptanalyzing a nonlinear chaotic algorithm (nca) for image encryption. Commun Nonlinear Sci Numer Simul 14(11):3743–3749

3. Arroyo D, Li C, Li S, Alvarez G, Halang WA (2009) Cryptanalysis of an image encryption scheme based on a new total shuffling algorithm. Chaos, Solitons Fractals 41(5):2613–2616
4. Biham E, Shamir A (1993) Differential cryptanalysis of the data encryption standard, vol 28. Springer, New York
5. Borujeni SE, Eshghi M (2013) Chaotic image encryption system using phase-magnitude transformation and pixel substitution. Telecommun Syst 52(2):525–537
6. Cheddad A, Condell J, Curran K, McKevitt P (2010) A hash-based image encryption algorithm. Opt Commun 283(6):879–893
7. Chen J-X, Zhu Z-L, Yu H (2014) A fast chaos-based symmetric image cryptosystem with an improved diffusion scheme. Optik-International Journal for Light and Electron Optics 125(11):2472–2478
8. Daemen J, Rijmen V (2002) The design of Rijndael: AES-the advanced encryption standard. Springer Science & Business Media
9. Dan T, Xiaojing W (2008) Image encryption based on bivariate polynomials. In: International conference on computer science and software engineering, 2008, vol 6. IEEE, pp 193–196
10. De S, Bhaumik J (2015) An aes-based robust image encryption scheme. Int J Comput Appl 109(12)
11. Enayatifar R, Abdullah AH, Isnin If (2014) Chaos-based Image encryption using a hybrid genetic algorithm and a dna sequence. Opt Lasers Eng 56:83–93
12. Fawaz Z, Noura H, Mostefaoui A (2016) An efficient and secure cipher scheme for images confidentiality preservation. Signal Process Image Commun 42:90–108
13. Hernandez-Castro JC, Tapiador JME, Quisquater J-J (2008) On the salsa20 core function. In: Fast software encryption. Springer, pp 462–469
14. Huang F, Feng Y (2009) Security analysis of image encryption based on twodimensional chaotic maps and improved algorithm. Front Electr Electron Eng Chin 4(1):5–9
15. Huang CK, Liao C-W, Hsu SL, Jeng YC (2013) Implementation Of gray image encryption with pixel shuffling and gray-level encryption by single chaotic system. Telecommun Syst 52(2):563–571
16. Huynh-Thu Q, Ghanbari M (2008) Scope of validity of psnr in image/video quality assessment. Electron Lett 44(13):800–801
17. Kumar A, Ghose MK (2011) Extended substitution–diffusion based image cipher using chaotic standard map. Commun Nonlinear Sci Numer Simul 16(1):372–382
18. Ledley RS (1965) The inverse of a boolean matrix. Technical report, DTIC Document
19. Li S, Zheng X (2002) Cryptanalysis of a chaotic image encryption method. In: IEEE international symposium on circuits and systems, 2002. ISCAS 2002, vol 2. IEEE, pp II–708
20. Li C, Chen MZQ, Lo K-T (2011) Breaking an image encryption algorithm based on chaos. Int J Bifurcation Chaos 21(07):2067–2076
21. Lou D-C, Sung C-H (2004) A steganographic scheme for secure communications based on the chaos and euler theorem. IEEE Trans Multimedia 6(3):501–509
22. McLoone M, McCanny JV (2002) Efficient single-chip implementation of sha-384 and sha-512. In: IEEE International conference on field-programmable technology, 2002.(FPT). Proceedings. 2002. IEEE, pp 311–314
23. Norouzi B, Seyedzadeh SM, Mirzakuchaki S, Mosavi MR (2014) A novel image encryption based on hash function with only two-round diffusion process. Multimedia Systems 20(1):45–64
24. Noura H, Courousse D (2015) HLDCA-WSN: homomorphic lightweight data confidentiality algorithm for wireless sensor network. Cryptology ePrint Archive, Report 2015/928. http://eprint.iacr.org/2015/928
25. Noura H, Couroussé D (2015) Lightweight, dynamic, and flexible cipher scheme for wireless and mobile networks. In: International conference on ad hoc networks. Springer, pp 225–236
26. Rhouma R, Belghith S (2008) Cryptanalysis of a new image encryption algorithm based on hyper-chaos. Phys Lett A 372(38):5973–5978
27. Rhouma R, Solak E, Belghith S (2010) Cryptanalysis of a new substitution–diffusion based image cipher. Commun Nonlinear Sci Numer Simul 15(7):1887–1892
28. Seyedzadeh SM, Mirzakuchaki S (2012) A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map. Signal Process 92(5):1202–1215
29. Seyedzade SM, Mirzakuchaki S, Atani RE (2010) A novel image encryption algorithm based on hash function. In: Machine vision and image processing (MVIP), 2010 6th Iranian. IEEE, pp 1–6
30. Shi ZJ (2004) Bit permutation instructions: architecture, implementation and cryptographic properties. Princeton University, Princeton, NJ
31. Tong X, Cui M, Wang Z (2009) A new feedback image encryption scheme based on perturbation with dynamical compound chaotic sequence cipher generator. Opt Commun 282(14):2722–2728
32. Upadhyaya A, Shokeen V, Srivastava G (2015) Image encryption: using aes, feature extraction and random no. generation. In: 4th international conference on reliability, infocom technologies and optimization (ICRITO) (trends and future directions), 2015, pp 1–4

33. Wang Y, Liao X, Xiao D, Wong K-W (2008) One-way hash function construction based on 2d coupled map lattices. Inf Sci 178(5):1391–1406
34. Wang X, Teng L, Qin X (2012) A novel colour image encryption algorithm based on chaos. Signal Process 92(4):1101–1108
35. Zhang G, Liu Q (2011) A novel image encryption method based on total shuffling scheme. Opt Commun 284(12):2775–2780
36. Zhang Y-Q, Wang X-Y (2014) Analysis and improvement of a chaos-based symmetric image encryption scheme using a bit-level permutation. Nonlinear Dyn 77(3):687–698
37. Zhang X, Ren Y, Shen L, Qian Z, Feng G (2014) Compressing encrypted images with auxiliary information. IEEE Trans Multimedia 16(5):1327–1336
38. Zhang X, Sun G, Shen L, Qin C (2014) Compression of encrypted images with multi-layer decomposition. Multimedia Tools and Applications 72(1):489–502
39. Zhou J, Au OC, Zhai G, Tang YY, Liu X (2014) Scalable compression of stream cipher encrypted images through context-adaptive sampling. IEEE Trans Inf Forensics Secur 9(11):1857–1868
40. Zhu C (2012) A novel image encryption scheme based on improved hyperchaotic sequences. Opt Commun 285(1):29–37

**Hassan Noura** is a research associate in the faculty of electrical and computer engineering of the American university of Beirut, Lebanon.

**Lama Sleem** is a PhD student in computer science in University Bourgogne Franche Comté, France.

**Mohamad Noura** is a PhD student in computer science in University Bourgogne Franche Comté, France.

**Mohammad M. Mansour** is a professor in the faculty of electrical and computer engineering of the American University of Beirut.

**Ali Chehab** is a professor in the faculty of electrical and computer engineering of the American University of Beirut.

**Raphaël Couturier** is a professor in computer science in University Bourgogne Franche Comté, France.