




Lightweight, dynamic and efficient image encryption scheme

Hassan Noura¹ · Ali Chehab¹ · Mohamad Noura² · Raphaël Couturier²  ·
Mohammad M. Mansour¹

Received: 26 November 2017 / Revised: 26 October 2018 / Accepted: 28 November 2018 /
Published online: 18 December 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

Recently, there has been a growing attention for Chaos-based image encryption algorithms. This class of algorithms relies on embedded chaotic maps to ensure a high security level with minimal performance overhead. One such algorithm, which we refer to as NCIES, was proposed recently, and the authors claimed that the algorithm achieves the required cryptographic properties with just a single round. In this paper, we first assess the performance of the NCIES cipher and we show that a single round is not enough for this cipher to ensure the desired cryptographic properties. In this context, we describe how such a cipher is vulnerable to a chosen plaintext/ciphertext attack. Next, we propose a new lightweight dynamic key-dependent cipher scheme that can address and overcome the issues identified in the NCIES cipher and other recent lightweight image encryption schemes. The proposed cipher is designed in a way to achieve a good balance between the latency, the required resources, and the security level when compared to recent chaotic image cipher schemes.

Keywords Avalanche effect · Chosen plain/cipher text attacks · Lightweight and dynamic cipher scheme · Security/Performance analysis

1 Introduction

The protection of digital images is an essential requirement for safeguarding the confidentiality and privacy of people in general [22, 23], and more so for patients when dealing with medical images. However, traditional cryptographic techniques using symmetric-key encryption algorithms such as DES [6] and AES [11], are not efficient for encrypting image

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

Hassan Noura
hassan.noura@aub.edu.lb

Mohammad M. Mansour
mohamad.noura@univ-comte.fr

¹ Maroun Semaan Faculty of Engineering and Architecture, Electrical and Computer Engineering Department, American University of Beirut, Beirut, Lebanon

² Université Bourgogne Franche-Comté (UBFC), Belfort, France

and video contents due to the intrinsic nature of images and the strong correlation among the adjacent pixels [17]. Therefore, traditional encryption algorithms do not meet the real-time requirements of multimedia streaming [12], which motivates the quest for alternative solutions.

1.1 Objective and contributions

One objective of this work is to analyze the recent lightweight image encryption algorithms presented in [15, 16, 38] and to assess their security and performance levels towards validating their suitability for protecting image contents. First, the NCIES algorithm requires a large memory size, which hinders its applicability to resource-constrained devices. We performed tests that confirmed the fact that the required number of iterations needed to achieve the required avalanche effect (plain-sensitivity property) is six and not one. With a single iteration only, the system is susceptible to different kinds of attacks such as known/chosen plain-text/cipher-text attacks since the cipher layers are based on diffusion primitives [40] that mix together the block bytes with the permuting bits. As such, the actual execution time of the secure NCIES scheme increases by a factor of 6. Therefore, to render the NCIES algorithm secure (6 rounds), the latency becomes inappropriate.

Another drawback of the scheme in [15] is that it is extremely sensitive to error, which precludes any practical implementation. An error of one bit can lead to the loss of the whole image content as we show in Section 3.4. This is caused by the avalanche effect, which employs a diffusion primitive with bit-level permutation.

Recent solutions, such as [16, 38], ensure that the avalanche effect covers the whole image by using the cipher block chaining (CBC) operation mode in forward and backward directions, without the bit permutation operation to reduce the required memory and to limit the effect of errors.

However, applying the CBC mode in forward and backward directions prevents the parallel encryption/decryption implementation and the effect of error propagation is doubled when compared to the original CBC mode, only in the forward direction.

In order to overcome the mentioned issues, in this paper, we propose a new efficient and lightweight cipher scheme that exhibits a better latency and resources' requirement. The proposed solution can be considered as a dynamic ECB mode, which uses key-dependent substitution tables (S-boxes) to perform the substitution operation and to generate the round block keys, which are different for each block and for each iteration. The generation of round keys is lightweight and sequential. To avoid any issues from this sequential generation approach, another block permutation step is introduced at the end of encryption to randomize the order of the blocks. The proposed solution is designed and implemented to achieve 1) a high security level, 2) minimal latency and required resources, and 3) low error propagation effect, especially when compared to [15, 16, 50]. The solution is based on an optimized binary diffusion matrix to benefit from the maximal linear branch number.

Note that the proposed scheme avoids chaining operations, and consequently, it can be realized in parallel, and it limits the effect of error propagation to only the corrupted blocks in contrast to [16, 38]. Moreover, the presented cipher is based on simple integer operations in contrast to [8, 18], which use floating-point arithmetic. The scheme is more efficient than [31], which uses asymmetric encryption.

The rest of this paper is organized as follows. In Section 3, the scheme of [15] is analyzed in terms of the avalanche effect, uniformity, and difference properties. Its security robustness is analyzed against different types of attacks in Section 4, and a chosen plain/cipher text that identifies a security loophole in the NCIES scheme is presented. In Section 5, an

efficient and lightweight dynamic cipher scheme is presented. In Section 6, a cryptanalysis is performed for the proposed cipher, followed by a performance analysis in Section 7 to assess the efficiency of the scheme in terms of latency, error propagation, and memory consumption. Finally, Section 8 concludes the paper.

Table of Notations

Notation	Definition
C	Number of columns of an image
R	Number of rows of an image
P	Number of plane (in gray-scale $P = 1$)
N_B	Number of blocks in one image
B	Number of bytes of a block
len	Number of bits of an image
K_p	Permutation sub-Key uses to construct π_{blocks} .
π	A Dynamic produced permutation table (P-box)
π^{-1}	The inverse corresponding the permutation table (P-box)
I	original image = $\{X_1, X_2, \dots, X_{N_B}\}$
C	Encrypted image $\{c_1, c_2, \dots, c_{N_B}\}$
c_j	The j th encrypted block of x_i
X_j	The j th original plain block
X_{jk}	k th byte of the i th original plain block
DX_j	The j th diffuses block
SK	Secret Key
N_o	Nonce
DK	Dynamic Key
K_{Si}	The i th dynamic substitution sub-key uses to construct the i th substitution table S_i , respectively.
π_{BL}	The produced dynamic permutation table uses for blocks permutations operation.
π_{BL}^{-1}	The inverse produced dynamic permutation table uses for inverse blocks permutation operation.
S_i	i th produced dynamic substitution table.
S_i^{-1}	The i th inverse corresponding substitution table.

2 Related work

One promising paradigm investigated by researchers in the last decade is chaos-based computing. Chaos consists of a non-linear dynamic system that appears to be random. Due to its extreme sensitivity to initial conditions, chaos has been considered for the design of cryptographic algorithms targeting digital images [1, 9, 26, 29, 44, 45, 48, 54].

However, chaos-based encryption algorithms are not always secure; actually, the majority of such algorithms have been successfully cryptanalyzed [32, 33, 42, 53]. This is due to their instability that stems from the periodicity of mapping [25] and the finite computing precision that renders the system vulnerable to various kinds of attacks [3, 4]. Another main disadvantage is that most chaos-based encryption algorithms rely on floating-point arithmetic, which significantly increases the computational complexity of any software or hardware implementation, and hence renders chaos-based cryptographic solutions inefficient when compared to integer-based schemes such as AES and DES. On the other hand, for more than a decade, several AES optimization were presented to make it more suitable

for real-time systems with high data rates, and for limited-power devices. The main objective of these optimization is to reduce the required resources and latency in addition to render its hardware implementation a simple one [5]. In addition, a set of optimized assembly AES instructions were introduced to the instruction set of Intel processors [20, 39].

In fact, AES may not satisfy the future requirements for lightweight cipher schemes according to NIST [35]. Therefore, a new initiative has been launched for the design of new lightweight cryptographic algorithms, especially for the Internet of Things (IoT).

Accordingly, different cipher algorithms have been proposed recently [5] to solve this issue by reducing the number of rounds. Examples of such algorithms include LEA [24], FeW[30], Prince [7], TWINE [47], Lblock [51], Piccolo [46], and LED [21].

However, since these ciphers are also based on static substitution and diffusion primitives, they require iterating a round function for a large number of rounds, r (see Table 1). This 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.

Ensuring data confidentiality within limited systems might introduce an overhead that would hinder their proper operation by impacting the overall system performance [35]. 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.

The minimum required number of iterations for recent lightweight cryptographic algorithms is 4 according to Table 1, which is the case of Hummingbird2 cipher. On the other side, the recent lightweight chaotic image encryption algorithms such as [8, 18, 27, 31, 36] base also on the multi-round structure. The hard challenge that limits the majority of chaotic cipher schemes that they use floating-point calculations and require conversion operations to integer, which introduces an important overhead in terms of latency and required

Table 1 List of recent lightweight encryption algorithms

Algorithm	No. of rounds	Key size	Block size	Structure
TEA	64	128	64	FN
XTEA	64	128	64	FN
LEA [24]		128	64	FN
HEIGHT	32	128	64	GFS
FeW[30]	32	80/128	64	FN-M
SIMON	32/36/42/44/52/ 54/68/69/72	64/72/96/128/ 144/192/256	32/48/64/ 92/128	FNI
PRESENT	31	80/128	64	SPN
RECTANGLE	25	80/120	64	SPN
LEA	24/28/32	128/192/256	128	FN
SPECK	22/23/26/27/28/ 29/32/33/34	64/72/96/128/ 144/192/256	32/48/64/92/128	FN
Prince [7]	11	128	64	SPN
AES	10/12/14	128/192/256	128	SPN
RC5	12	128	32/64/128	FN
Hummingbird2	4	256	16	SPN

resources. In addition, [31] requires asymmetric encryption, which requires more resources and introduces more overhead [19] when compared to symmetric encryption.

Recently, an integer chaotic cipher scheme was proposed in [15], and we shall refer to it as the NCIES cipher; it is based on a static binary diffusion layer, followed by a key-dependent bit-permutation that is based on a periodic 2D cat map, and that iterates only for a single round. The authors claim that the avalanche effect, difference and uniformity properties can be achieved after the first round, as shown in Fig. 7 of [15]. The authors deduced such results based on a small image of size 16×16 , and then, they generalized the avalanche effect result of one round to larger images. However, typically, the number of iterations increases with the number of blocks.

3 Analysis of the NCIES approach

In this section, plain message sensitivity, difference between plain and encrypted messages, and uniformity tests of the NCIES cipher scheme [15] are performed to analyze its performance. The encryption and decryption schemes of this solution are illustrated in Figs. 1 and 2, respectively. From a cryptanalysis viewpoint, the required number of iterations is determined when the required security level is reached, independent of the message type. Such a value cannot be justified based on standard images such as Lenna. The tests conducted should take into account the possibility of chosen plain/cipher messages attack, whereby an attacker may select any message to recover the secret or figure out the cryptographic primitives. These tests should employ special messages that trigger the maximum number of iterations to better quantify the required number of rounds to reach the desired cryptographic strength. Examples of such messages are those having all bytes inactive (zero) except for just one random non-zero byte with the least significant bit (LSB) set to 1.

3.1 Plain/cipher text sensitivity

An acceptable sensitivity level leads to considerable change in the cipher text (or decrypted plain text) in response to a slight change in the original (or encrypted) message. Let I and I'

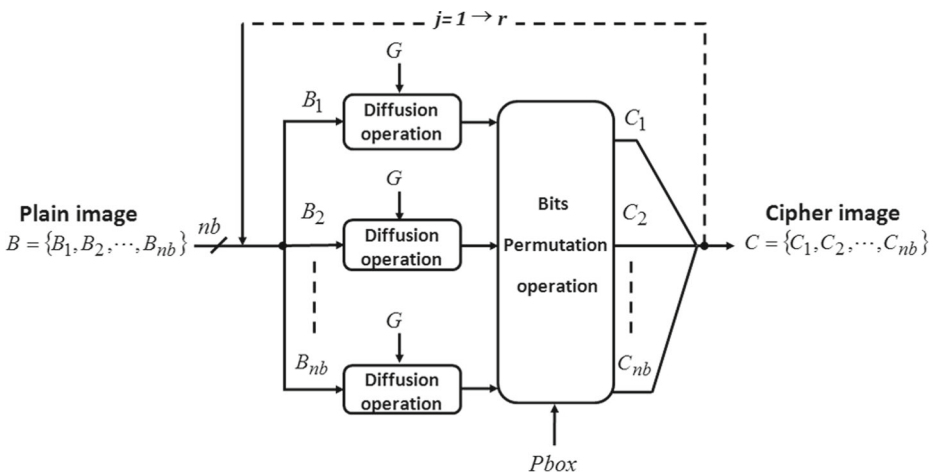


Fig. 1 Architecture of the NCIES encryption algorithm described in [15]

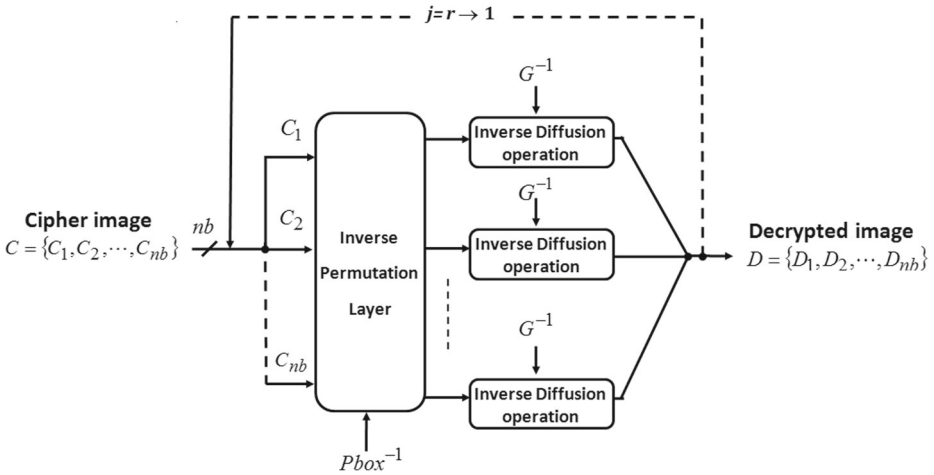


Fig. 2 Architecture of the NCIES decryption algorithm described in [15]

represent two chosen images. A cipher algorithm $E_K()$ with secret key K is considered to be robust against chosen/known plain/cipher text attacks, if it ensures the avalanche effect when applied to I and I' . In other words, the percentage of the Hamming distance (in bits) between the corresponding cipher images $C = E_K(I)$ and $C' = E_K(I')$ should be close to 50%, while I and I' only differ by one bit. The chosen image I has all values equal to zero, while I' differs from I by only one random pixel that has a value of one.

The sensitivity (PS) of the plain image is analyzed for a large number N of random plain images and secret keys, using the following average Hamming distance metric between the cipher images:

$$PS \triangleq \frac{1}{N} \sum_{w=1}^N H(E_{K_w}(I_w), E_{K_w}(I'_w)) \tag{1}$$

$$H(C_w, C'_w) = \frac{1}{T} \sum (\text{Byte2bit}(C_w) \oplus \text{Byte2bit}(C'_w)) \times 100\% \tag{2}$$

where T is the length in bits of the original and cipher images, $C_w = E_{K_w}(I_w)$ and $C'_w = E_{K_w}(I'_w)$ are the corresponding w th cipher images of the original images I_w and I'_w , respectively, and K_w is the w th secret key. The **Byte2bit** function converts its input argument into a binary string, and the XOR operator returns the bit-wise exclusive-or of its string operands. The summation in (2) accumulates the non-zero bits in the resulting binary string. We emphasize that all the elements of I'_w and I_w are zero, except for a randomly chosen byte that has its LSB set to one.

The avalanche effect results are plotted in Fig. 3 for the Lenna image and for the chosen image, respectively, for different sizes M . The results show the average value of PS over $N = 1,000$ random dynamic keys versus the number of rounds r for $M \times M$ sized images. The results clearly indicate that the minimum number of iterations required to reach the avalanche effect for different image sizes is at least six. The results are shown numerically in Table 2 for the Lenna image. These results differ from the ones presented in [15]. On one hand, setting $r = 1$ as recommended in [15] yields a low plain/cipher text sensitivity, which renders the approach weak against chosen/known plain-text attacks since the scheme

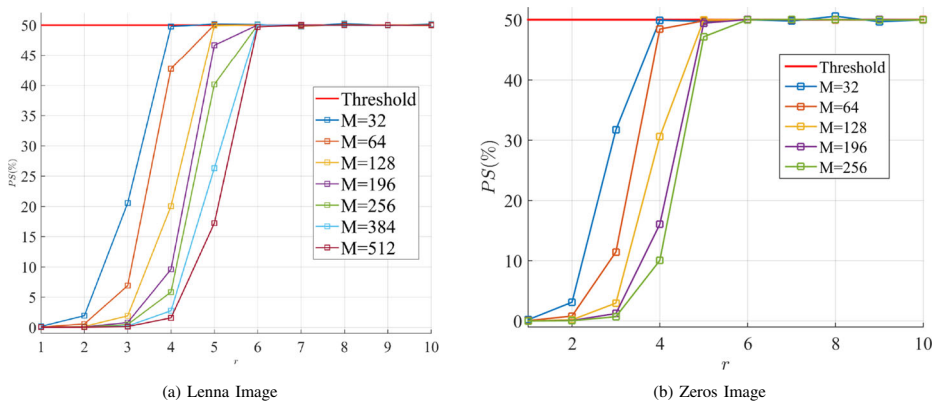


Fig. 3 Variation of the avalanche effect over 1000 random keys versus the number of rounds r for **a** the Lenna image, and **b** the chosen Zeros image using the NCIES algorithm

is based on static diffusion primitives. On the other hand, selecting $r = 6$ leads to a six-fold increase in execution time and power consumption, which defeats the main objective of [15].

3.2 Independence

The independence between the plain and cipher images is measured using the Hamming distance metric between I and its encrypted version C :

$$H(I_w, C_w) = \frac{1}{T} \sum (\text{Byte2bit}(I_w) \oplus \text{Byte2bit}(C_w)) \times 100\% \tag{3}$$

where T, I_w, C_w, K_w are as defined in (1)–(2). Moreover, all the elements of I_w are equal to zero, except for a randomly chosen byte whose LSB is set to one, for $w = 1, \dots, N$.

Figure 4 plots the independence between the plain and cipher images for Lenna and Zeros. Figure 4a shows that the independence between original and encrypted Lenna is obtained after the first round. However, Fig. 4b clearly shows that the original scheme of [15] requires the same number of iterations as the avalanche effect for the zeros image,

Table 2 Variation of avalanche effect versus size of Lenna image and number of rounds r using the NCIES algorithm

$M \times M$	$r = 1$	$r = 2$	$r = 3$	$r = 4$	$r = 5$	$r = 6$	$r = 7$
16×16	0.3092	4.2415	29.052	45.09	49.39	50.03	49.97
32×32	0.077	1.135	11.54	40.26	48.29	49.6	49.86
64×64	0.0193	0.342	5.720	44.70	49.99	50.08	49.64
128×128	0.004	0.082	1.407	15.00	47.34	50.03	49.9
196×196	0.0021	0.035	0.52	4.51	39.80	49.95	49.99
256×256	0.0012	0.0212	0.3366	5.348	42.441	50.01	49.97
300×300	0.0009	0.0152	0.2588	4.290	38.273	50.04	50.12
512×512	0.0003	0.005	0.087	1.429	19.688	49.99	50.02

Bold emphasis indicate the threshold number of iterations is 6

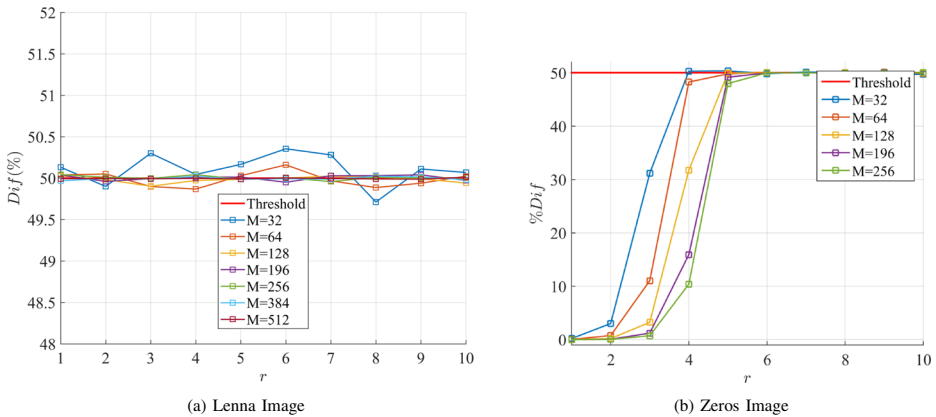


Fig. 4 Variation of the difference between plain and cipher-text over 1000 random keys versus the number of rounds *r* for **a** Lenna, and **b** the chosen Zeros image using the NCIES algorithm

which means that *r* should be at least six to achieve independence. This validates our recommendation for the need to rely on images of mostly zeros to better quantify the required number of rounds necessary to achieve independence and the avalanche effect.

3.3 Uniformity

Resistance against common statistical attacks is assessed using the uniformity property, which measures how uniform the frequency of occurrence of each symbol in the encrypted image is. On average, each symbol should have a probability of occurrence close to $\frac{1}{n}$, where *n* is the number of symbols. To determine the degree of uniformity of an encrypted image, the chi-square test is applied as:

$$\chi^2 = \sum_{i=0}^{Q-1} \frac{(o_i - e)^2}{e}, \tag{4}$$

where *Q* is the number of gray levels (e.g., *Q* = 256 for gray scale images), *o_i* is the frequency of occurrence of the *i*th gray level, and $e = \frac{len}{Q}$ is the desired uniform frequency, with *len* being the image length in bytes.

This statistical test is used to compare the observed data with a specific hypothesis. A null hypothesis is formulated, which is then rejected or accepted with the help of the statistical test. The probability value below which the null hypothesis is rejected is called the alpha level or the “significant level”. A typical level of confidence of 5% is often chosen to decide whether the null hypothesis is false or not [13]. A significance level of 5% is equivalent to a 95% confidence that a given result cannot occur [49]. If the number of gray levels *Q* equals 256, the chi-square test reaches a maximal value that is close to 293 [10].

So, all values lower than 293 are acceptable and satisfy the uniformity property. Accordingly, we employ the chi-square test in this paper to quantify the required number of rounds necessary to reach uniformity.

The cipher image *C_w* defined earlier versus the number of rounds is used as input to the chi-square test, again assuming the image *I_w* has the same structure as defined above.

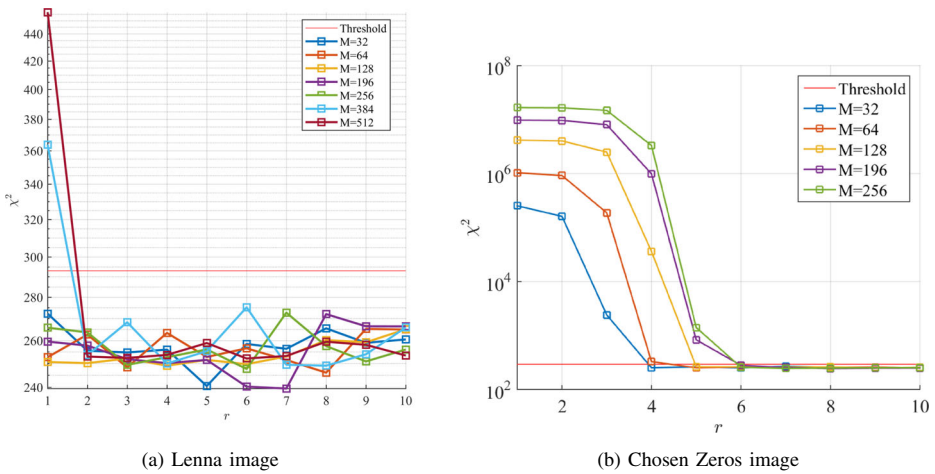


Fig. 5 Variation of the mean chi-square value over 1000 random keys versus the number of rounds r for $Q = 256$ gray levels, using the NCIES algorithm for **a** Lenna, and **b** the chosen Zeros image

Figure 5 shows that the mean chi-square value reaches at most 293 after 2 iterations for certain tested Lenna image sizes, and 6 iterations for all tested chosen Zeros image sizes. This confirms our earlier observation that $r \geq 6$ is needed to achieve the desired cryptographic strength.

3.4 Error propagation

Another important criterion to be satisfied by any cipher is error tolerance as opposed to error propagation. Interference and noise are very typical in transmission channels [34] and storage media [43], which effectively cause bit errors to occur. A single bit error refers to the change of a single bit value from ‘0’ to ‘1’ or vice versa. If an error propagates, it leads to the destruction or degradation of the received data. As such, it is a challenge to design a secure cipher scheme that ensures the global avalanche effect, while simultaneously guaranteeing low error propagation [34].

In case of the cipher scheme presented in [15], a random bit error, in an encrypted block, affects the whole decrypted image. The difference between both decrypted images is calculated by using (2). Figure 6 shows the effect of channel errors on the decrypted image. The Figure clearly shows that any bit error causes about 50% error, which indicates that the scheme is highly prone to error propagation. This can be seen visually in Fig. 7, which shows the degradation in the decrypted image versus channel errors.

Hence, such a scheme is not suitable for wireless communication channels that are subject to different kinds of noise effects [2]. As a result, the approach of [15] is inefficient compared to existing traditional ciphers (block or stream) from an error propagation perspective [14]. For example, the effect of bit errors in cipher block modes is limited to only one block entirely in the case of ECB, or partially in the cases of OFB and CTR modes. In CBC mode, such an error affects two blocks, where the first block is completely affected and the second is partially damaged.

In addition, the proposed cipher scheme will not affect three blocks as in [16, 38] or the whole image as in [15].

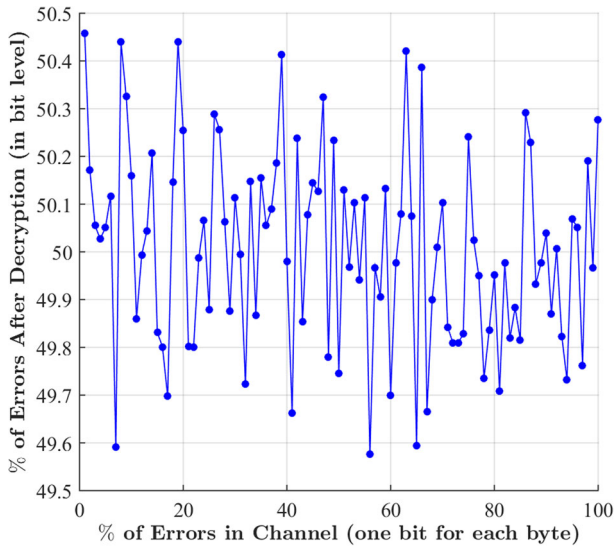


Fig. 6 Error propagation versus percentage of errors using the NCIES algorithm

In summary, the proposed encryption/decryption is done at the block level, and as such, an error that occurs in a byte of an encrypted block affects only its corresponding block and it will not affect its neighbours blocks. Thus, low error propagation is guaranteed and an error correction scheme is presented, which is a great advantage for the proposed cipher scheme.

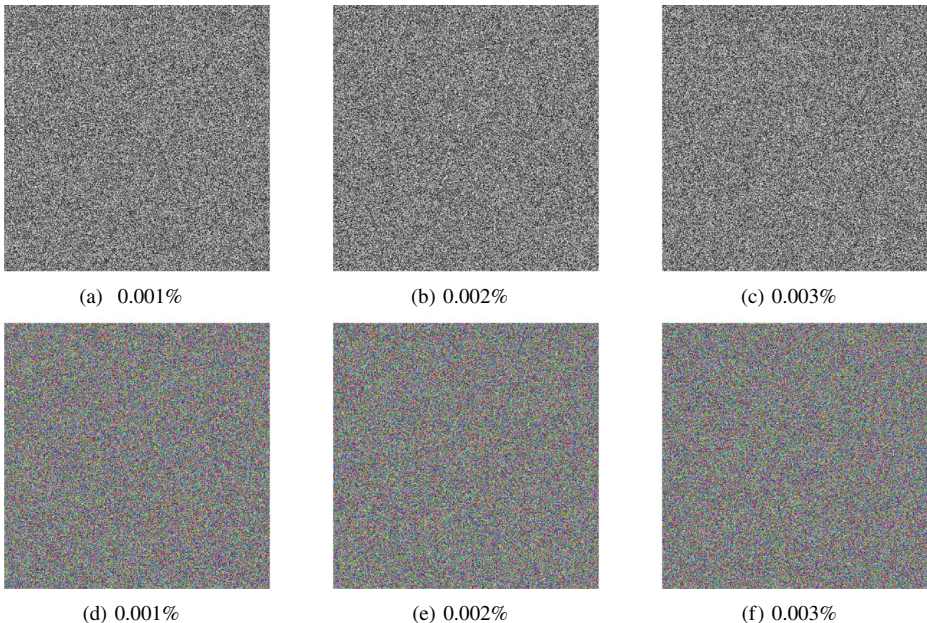


Fig. 7 Decrypted images in relation to the percentage of channel errors of [15]

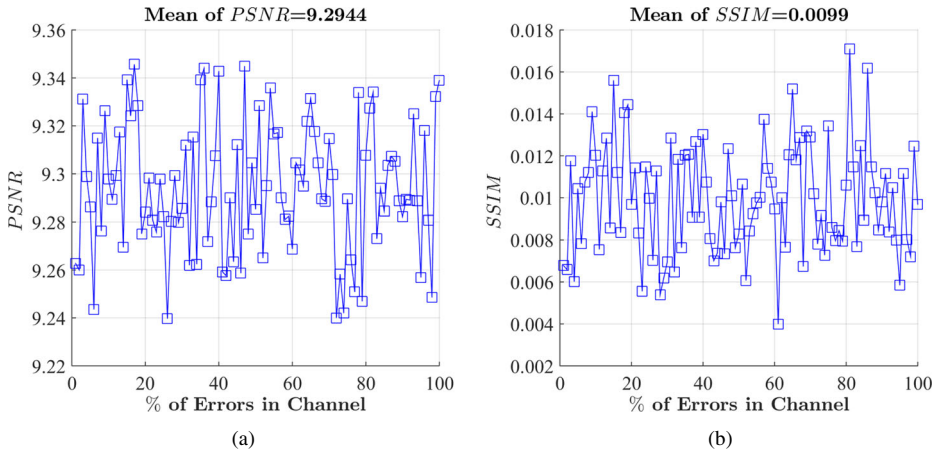


Fig. 8 Average of PSNR and SSIM between both decrypted Lenna images versus the percentage of channel errors, for 1000 random secret keys

3.4.1 Visual degradation effect of error propagation

This test is specific to image and video contents, and it quantifies the visual degradation of a cipher scheme in relation to error propagation. The resulting degradation in the decrypted image after any single bit change creates noticeable distortion. To measure the visual degradation, two well known parameters are studied to measure the encryption visual quality: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM).

PSNR is derived from the mean-squared error (MSE) quantity, which represents the cumulative squared error between an original and an encrypted image. A low PSNR value indicates that there is a large difference between the original and the cipher images. On the other hand, the SSIM index [32] is a metric based on the Human Visual System, and it quantifies the similarity between two images. The SSIM index lies in the interval [0,1]. A value of 0 means that there is no correlation between the original and the cipher images, while a value close to 1 means that both images are approximately the same. In this context, *PSNR* and *SSIM* are measured between two decrypted Lenna images (where the second decrypted image corresponds to the encrypted image with an error percentage). Figure 8a and b plot the variations of PSNR and SSIM as a function of error percentage, respectively. The results again confirm that the NCIES cipher results in a high difference between both decrypted images, even for low error percentages. Consequently, high visual distortion results from a small error percentage, with one-bit error being sufficient to destroy the whole image.

Table 3 Statistical results for the Lenna image over 1,000 random keys

	Integer diffusion matrices			
	Min	Mean	Max	Std
Dif _{Decr}	49.7505	50.0091	50.2399	0.1014
PSNR	9.2397	9.2944	9.3457	0.0275
SSIM	0.0040	0.0099	0.0171	0.0026

Table 3 shows the statistics of PSNR, SSIM, and DIF_{Decr} , where DIF_{Decr} is the sensitivity test that measures the difference between two decrypted images when a percentage of errors is introduced into the encrypted image (we assume that it is the effect of a channel noise in the encrypted image). We can see that it is close to 50%, which is a relatively high value that prevents the recovery of the original image. Accordingly, no useful visual information or structure about the original image can be deduced from the decrypted image if any error is introduced into the channel.

4 Cryptanalysis of the NCIES approach

The NCIES cipher is based on diffusion primitives without any substitution operation, which means that the confusion property is missing. Consequently, the cipher is vulnerable to various kinds of attacks, such as chosen/known plain/cipher text attacks [40] as well as brute force attacks for small image sizes. We propose a chosen plain/cipher text attack that can break the cipher with only two chosen messages. Note that known plain/cipher text attacks can be realized, but they require a large number of plain and cipher text pairs.

4.1 Chosen plain/cipher text attack

Here, one can select any input message. The binary diffusion matrix \mathbf{D} is known and is independent of the secret key. Also, its corresponding inverse matrix \mathbf{D}^{-1} is available for the attacker. Therefore, the goal of this attack is to construct the inverse bit-permutation table, which can be considered as the only secret in the analyzed approach. This can be easily achieved by recovering the secret control parameters (u and v) and the initial conditions (rx and ry). In the following, we show how the chosen plain/cipher text attack can be devised based on specifically chosen blocks.

Note that an essential operation, $(r, c) = \text{index2sub}(ix)$, is required to provide the equivalent row r and column c subscripts of a linear index ix , and can be calculated according to the following equation:

$$\begin{aligned} r &= (ix - 1) \bmod M + 1 \\ c &= \lfloor (ix - 1)/M \rfloor + 1 \end{aligned}$$

where $ix = 1, 2, \dots, len, len = 8 \times N_B \times B = M \times M; N_B$ is the number of byte-blocks in the message, B is the number of bytes per block, $M = \sqrt{len}$ is the dimension of the square matrix in bits, and $\lfloor x \rfloor$ represents the largest integer less than or equal to x . Conversely, the operation $ix = \text{sub2index}(r, c)$ is used to recover a linear index from its row and column subscripts as follows:

$$ix = (c - 1) \times M + r$$

The attacker constructs an image I using N_B blocks X_i each containing B bytes such as $X = \{X_1 || \dots || X_i || \dots || X_{N_B}\}$. Moreover, all blocks X_i are null except for one block, say the j th block X_j , which has a non-zero byte value. Hence,

$$\begin{aligned} X_i^T &= \underbrace{\mathbf{0} \ \mathbf{0} \ \dots \ \mathbf{0} \ \mathbf{0}}_{B \text{ zero bytes}}, \quad i = 1, 2, \dots, N_B; \quad i \neq j \\ \mathbf{0}^T &= \underbrace{0 \ 0 \ \dots \ 0}_{Q=8 \text{ bits}} \end{aligned}$$

where $()^T$ is the transpose operator. The block X_j is chosen according to a specific rule: its corresponding diffused block has only one active byte, say the k th byte X_{jk} , different than zero, and that this byte has only one bit, say the q th bit, different than zero.

$$\begin{aligned}
 X_j^T &= \underbrace{X_{j1} \dots X_{jk} \dots X_{jB}}_{B \text{ bytes}} \\
 &= \mathbf{0} \dots \underbrace{X_{jk}}_{\neq 0} \dots \mathbf{0} \\
 X_{jk}^T &= b_{Q-1} \dots b_q \dots b_1 b_0 \quad (Q\text{-bit representation of byte } X_{jk}) \\
 &= 0 \dots \underbrace{b_q}_{\neq 0} \dots 00 \quad (b_q \in \{0, 1\})
 \end{aligned}$$

This means that the diffusion block of X_j , denoted as DX_j with B bytes, has only one-byte element equals to 2^q , where $q = 0, 1, 2, \dots, 7$ as presented in the following form:

$$DX_j^T = \mathbf{0} \dots \underbrace{2^q}_{\neq 0} \dots \mathbf{0}$$

Let \odot denote the vector “and-xor” dot-product operator between two binary vectors of length B :

$$\mathbf{v}^T \odot \mathbf{w} = (v_1 \cdot w_1) \oplus (v_2 \cdot w_2) \oplus \dots \oplus (v_B \cdot w_B) \tag{5}$$

Then, to obtain the required X_j for the chosen plaintext attack, the inverse diffusion operation is multiplied by the desired diffused block DX_j as shown in the following equation:

$$X_j = \mathbf{D}^{-1} \odot DX_j, \tag{6}$$

Note that in (6), the entries of \mathbf{D}^{-1} are bits while the entries of DX_j are bytes. The “and-xor” operator \odot between a bit-vector and a byte-vector is defined similar to (5), where the presence of a 1 in \mathbf{D}^{-1} indicates that the corresponding byte in DX_j is included in the cumulative-xor, and a 0 indicates it is excluded. For example, assume that the desired DX_j has the last byte as the active one (its LSB is active) as follows:

$$\begin{aligned}
 DX_j^T &= \underbrace{\mathbf{00} \dots \mathbf{0} \dots \mathbf{0}}_{B-1 \text{ zero bytes}} \mathbf{1} \\
 \mathbf{1}^T &= \underbrace{00 \dots 01}_{Q=8 \text{ bits}}
 \end{aligned}$$

The desired X_j is obtained by using (6) that will produce the desired diffused block DX_j . Therefore, for the previous example with $B = 32$, $Q = 8$, and \mathbf{D} as defined in [15] (see Appendix B), the corresponding X_j is

$$X_j^T = \mathbf{1010\ 0000\ 1001\ 1110\ 0000\ 1101\ 0000\ 0100}$$

The second example of the chosen plain-text block should provide diffusion DX_j with only the first byte being the active one (MSB is active) as presented in the following:

$$DX_j^T = \mathbf{1} \underbrace{\mathbf{00} \dots \mathbf{0} \dots \mathbf{0}}_{B-1 \text{ zero bytes}}$$

Its corresponding X_j (according to (6)) should be equal to:

$$X_j^T = \mathbf{1000\ 0000\ 1011\ 0000\ 1101\ 1110\ 1101\ 0101}$$

In fact, the desired diffused block DX_j is first chosen, then the original block X_j can be computed. It is worth mentioning that the diffusion matrix used in [15] (see Appendix B) is not what is claimed in [28].

On the other hand, to launch chosen ciphertext attacks, we need to select diffused block DX_j that can provide X_j with only one active byte and with only one active bit. To obtain the required DX_j for the chosen ciphertext attack, the diffusion operation is performed on the desired original (inverse diffusion) block X_j as expressed in the following equation:

$$DX_j = \mathbf{D} \odot X_j \tag{7}$$

For example, if the desired X_j has the last byte as the active one and with only the LSB active as presented in the following:

$$X_j^T = \underbrace{00 \dots 0 \dots 0}_{B-1 \text{ zero bytes}} \mathbf{1}$$

then the corresponding diffused block DX_j according to (7) is

$$DX_j^T = \mathbf{1010\ 0001\ 0000\ 0111\ 0001\ 1101\ 0000\ 0111}.$$

In addition, a second example of chosen cipher block DX_j that can provide X_j with only the first byte having a value equal to 1 as presented in the following:

$$X_j^T = \mathbf{1} \underbrace{00 \dots 0 \dots 0}_{B-1 \text{ zero bytes}}$$

Its corresponding DX_j (according to (7)) should be equal to:

$$DX_j^T = \mathbf{1110\ 1001\ 1011\ 1001\ 0111\ 0000\ 1001\ 0101}.$$

Now, for chosen plain-text attacks, an attacker builds a set of original images $\{I_1, I_2, \dots\}$, where for each original one, a specific unique non-null original block X_j is introduced that can ensure that the output diffused block has only one active bit. However, for chosen ciphertext attack, an attacker builds a set of encrypted images $\{C_1, C_2, \dots\}$, where for each encrypted one, a specific unique non-null encrypted DX_j block is introduced that can ensure that the output inverse diffusion block has only one active bit.

In fact, the diffusion of zero blocks will provide zero blocks and only the diffused X_j block, (DX_j), will provide a non-zero block with only one non-zero byte that has only one active bit. In addition, the permutation/inverse permutation operation will not change the byte values since all are zero, except for just one byte. Therefore, the indices of this bit before and after permutation (ix and ixp , respectively) are known by the attacker now for each chosen plain image or cipher image. The index ixp can be obtained easily from the encrypted/decrypted message since it is the only non-null bit. Then, the row and column subscripts of ix and ixp can be obtained as $(r, c) = \text{index2sub}(ix)$ and $(rp, cp) = \text{index2sub}(ixp)$.

Note that two matrices (\mathbf{A} and \mathbf{R}_{xy}) are generated from the permutation key and they are used to build the permutation table and its corresponding inverse permutation table.

For a successful chosen plain-text attack or vice-versa for the chosen cipher-text attack, these matrices should be known. Indeed, \mathbf{A} is based on both control parameters (u, v) and \mathbf{R}_{xy} is based on two initial conditions rx and ry .

$$\mathbf{A} = \begin{bmatrix} 1 & u \\ v & 1 + u \times v \end{bmatrix}, \text{ and } \mathbf{R}_{xy} = \begin{bmatrix} rx + ry \\ ry \end{bmatrix} \tag{8}$$

Then, from the first chosen plain-text or cipher-text attack, the following relation can be established:

$$\begin{bmatrix} rp \\ cp \end{bmatrix} = \mathbf{A} \cdot \begin{bmatrix} r \\ c \end{bmatrix} + \mathbf{R}_{xy} = \begin{bmatrix} 1 & u \\ v & 1 + u \times v \end{bmatrix} \cdot \begin{bmatrix} r \\ c \end{bmatrix} + \begin{bmatrix} rx + ry \\ ry \end{bmatrix} \tag{9}$$

Additionally, one chosen plain/cipher text is not sufficient to solve this transformation and to recover these unknown parameters and initial conditions. Therefore, we require iterating the chosen plain/cipher attack again while choosing a new different chosen block to obtain a new system of equations. So, the same steps are repeated and new indices of this bit before (*ix2*) and after (*ixp2*) permutation are computed. Similarly, *ixp2* can be obtained easily from the encrypted/decrypted message since it is the only non-null bit. Therefore, their corresponding rows and columns (*r2*, *c2*) for *ix2* and (*rp2*, *cp2*) for *ixp2* can be calculated using also the *index2sub* operation.

This means a new system of equations can be built, which is given by:

$$\begin{bmatrix} rp2 \\ cp2 \end{bmatrix} = \mathbf{A} \cdot \begin{bmatrix} r2 \\ c2 \end{bmatrix} + \mathbf{R}_{xy} \text{ mod } M = \begin{bmatrix} 1 & u \\ v & 1 + u \times v \end{bmatrix} \cdot \begin{bmatrix} r2 \\ c2 \end{bmatrix} + \begin{bmatrix} rx + ry \\ ry \end{bmatrix} \text{ mod } M \tag{10}$$

Now, subtracting (10) from (8) permits to remove \mathbf{R}_{xy} and the output becomes only related to the unknown matrix \mathbf{A} :

$$\begin{bmatrix} rp2 - rp1 \\ cp2 - cp1 \end{bmatrix} = \begin{bmatrix} 1 & u \\ v & 1 + u \times v \end{bmatrix} \cdot \begin{bmatrix} r2 - r1 \\ c2 - c1 \end{bmatrix} \text{ mod } M \tag{11}$$

Then, *u* can be calculated by using the following equation:

$$u = \frac{rp2 - rp1 - r2 + r1}{c2 - c1} \text{ mod } M \tag{12}$$

After the calculation of *u*, *v* can be computed by using the following equation:

$$v = \frac{(cp2 - cp1 - c2 + c1)}{r2 - r1 + u \times (c2 - c1)} \text{ mod } M \tag{13}$$

Now, the attacker can build the matrix \mathbf{A} since *u* and *v* are known. The next step is to determine the matrix \mathbf{R}_{xy} . This is possible by using (8) or (10). In the following, (8) is used to recover first *ry* and then *rx*.

ry and *rx* can be computed using the following equations:

$$ry = cp1 - v \times r1 - (1 + u \times v) \times c1 \text{ mod } M \tag{14}$$

$$rx = rp1 - r1 - u \times c1 - ry \text{ mod } M \tag{15}$$

Therefore, the proposed cryptanalysis requires only two chosen plain/cipher blocks to recover the permutation key (*u*, *v*, *rx*, and *ry*). Accordingly, the employed permutation table P-box and its corresponding inverse can be generated by using the recovered permutation key. Indeed, the inverse permutation table π^{-1} can be obtained by using the produced P-box π according to the following equation:

$$\pi^{-1}[\pi(t)] = t, t = 1, 2, \dots, M^2 \tag{16}$$

where $1 \leq \pi(t) \leq M^2$. Then, by calculating the permutation or inverse permutation table, chosen plain-text or cipher-text attacks allow the decryption of the next encrypted images easily. Consequently, NCIES is broken and it cannot ensure the data confidentiality.

As lessons learned from NCIES cipher, several security vulnerabilities should be fixed and these are as follows:

1. with only one round, the avalanche effect is not reached.
2. the diffusion process is broken since no addition round key operation exists in the analyzed scheme. The diffusion of zero blocks will preserve their zero values after diffusion.
3. no substitution operation to ensure the non-linearity property and to make the solution of this system of equations a hard task.

Algorithm 1 presents the steps of chosen plain-text attacks, which can be summarized by forming two plain images, followed by calculating the indices of the active bit before and after encryption, and then, using these indices to obtain the permutation key, which in turn is used to form the permutation table and consequently its inverse. In addition, to perform a chosen cipher-text attack, minor modifications are required, which are:

1. Using chosen cipher images instead of the original ones, and
2. Iterating the decryption algorithm instead of the encryption one (lines 5 and 10).

In the pseudo-code, the notation $A(i : j)$ refers to the elements of \mathbf{A} at indices i to j .

Algorithm 1 Proposed chosen plaintext algorithm

Input: Two Chosen plain-blocks length $(X1_j, X2_j)$; and active byte indices $(ix1, ix2)$.

Output: The permutation table $Pbox$ and its corresponding inverse one Inv_Pbox .

- 1: **procedure** $(Pbox, Inv_Pbox) = \text{ChosenPlainTextAttack}(X1_j, X2_j, ix1, ix2)$
- 2: $I(1 : len) \leftarrow 0$
- 3: $I1 \leftarrow I$ ▷ Building of the first chosen image
- 4: $I1_{(j-1) \times 32 + 1 : j \times 32} \leftarrow X1_j$ ▷ j represents the chosen blocks
- 5: $C1 \leftarrow \text{encryption_NCIES}(I1)$
- 6: $C1b \leftarrow \text{byte2bit}(C)$
- 7: $ixp1 \leftarrow \text{find}(C1b == 1)$
- 8: $I2 \leftarrow I$ ▷ Building of the second chosen image
- 9: $I2_{((j-1) \times 32 + 1 : j \times 32)} \leftarrow X2_j$
- 10: $C2 \leftarrow \text{encryption_NCIES}(I2)$
- 11: $C2b \leftarrow \text{byte2bit}(C2)$
- 12: $ixp2 \leftarrow \text{find}(C2b == 1)$ ▷ Calculate the permutation key $(u, v, rx$ and $ry)$
- 13: $(r1, c1) \leftarrow \text{ind2sub}(ix1)$
- 14: $(rp1, cp1) \leftarrow \text{ind2sub}(ixp1)$
- 15: $(r2, c2) \leftarrow \text{ind2sub}(ix2)$
- 16: $(rp2, cp2) \leftarrow \text{ind2sub}(ixp2)$
- 17: $(u, v) \leftarrow \text{CalculateSecretParameters}(r1, c1, rp1, cp1, r2, c2, rp2, cp2)$
- 18: $(ry, rx) \leftarrow \text{CalculateInitialConditions}(r1, c1, rp1, cp1, u, v)$ ▷ Iterate cat map using the obtained permutation key then the sub2ind operation
- 19: $Pbox \leftarrow \text{ConstructPermutationTable}(u, v, rx, ry)$ ▷ use (16)
- 20: $Inv_Pbox \leftarrow \text{CalculateInversePermutationTable}(Pbox)$
- 21: **return** $Pbox, Inv_Pbox$
- 22: **end procedure**

In the following section, we propose a cipher scheme that introduces substitution and round block key operations, in addition to ensuring the avalanche effect at the block level.

Moreover, round keys are different for each block and for each iteration to complicate the cryptanalysis process and to make it a hard task.

4.2 Brute force attacks

In fact, NCIES [15] uses the secret key to produce the permutation key. Moreover, the permutation key is divided into two parts: control parameters (u and v) and two initial conditions (rx and ry), where each one requires qp bits. Therefore, the size of the permutation key is $4 \times qp \times rp$, where $qp = \lceil \log_2(M) \rceil$ and $r = rp = 1$ according to their configuration. Therefore, the permutation key depends on the size of the square input image ($M \times M$). Furthermore, independent of the large size of the secret key, which is used as a seed for this chaotic generator, the length of the produced permutation key is not sufficient to resist brute force attacks, specifically for small-size images.

In fact, for $M \leq 256$, their chaotic generator is iterated once to produce a permutation key with 32-bit length. Hence, a brute force attack is possible [40] for this small size of the permutation key. On the other hand, their generator may require to iterate twice for $M \geq 256$. This means that the length of the permutation key is 64 and it is also not sufficient to resist brute force attacks [40].

The analyzed scheme [15] can be broken by employing other types of attacks because of $r = 1$. For example, the scheme cannot resist statistical attacks (uniformity is not reached) in addition to chosen/known plain-text attacks with the miss-configuration. Moreover, the problem of single image failure and accidental key disclosure is not taken into consideration by this scheme. Finally, the analyzed cipher cannot resist different kinds of attacks in addition to different performance limitations.

4.3 Discussion and recommendations

Based on the results of the NCIES cipher analysis, a new efficient cipher solution should be designed to overcome the discussed weaknesses. Several recommendations will be presented and discussed to justify the proposed cipher structure and how such a solution achieves better efficiency without degrading the security level.

1. Employing key dependent Substitution-Diffusion Primitives: The proposed cipher is based on key-dependent substitution and diffusion primitives similar to the techniques presented in [37, 38]. The key-dependent approach for the confusion and diffusion primitives is preferable since different entities use different primitives that are unknown by the attackers and hence, introduce a challenge to break.
2. Dynamic-key Approach: Our main recommendation is to use a dynamic key-dependent approach [16, 37, 38, 41] to achieve high resistance against attacks. This makes the cipher primitives change with time and constitute a hard problem for the attackers who will not be able to gather useful information from different sets of encrypted/decrypted images resulting from different dynamic keys. Moreover, the binary diffusion matrix is independent of the secret key. Recent cipher solutions that employ a dynamic key approach are presented in [16, 37, 38]. Note that the dynamicity should be preserved such that the confusion and diffusion primitives are able to reach the required level of cryptographic strength.

3. Towards reducing the high error propagation: To limit the effect of error propagation, the encryption should be performed independently of the block level, which can be considered as a good balance between the avalanche effect and error propagation.
4. Towards reducing the memory consumption: Substitution and diffusion primitives should be performed at the byte or word levels instead of the bit level, and the round function should be limited to the block level to reduce the required memory size, latency and error propagation. Note that the bit permutation operation introduces an important overhead compared to the byte permutation since it requires conversion and re-conversion operations. This was the case of DES and to avoid such an overhead, the AES logic was recommended to avoid all cipher operations at the bit level.
5. Towards reducing the latency: Designing a simple lightweight round function is essential to reduce the latency and the required resources. More importantly, reaching the avalanche effect with the minimum possible number of iterations is mandatory. In fact, the dynamic key-dependent approach allows for the reduction of the number of rounds since the cipher primitives would not be known to the attacker and they are built to ensure a high possible level of confusion, diffusion and avalanche effect at the block level. Additionally, the cipher scheme should avoid chaining operations between blocks to allow for parallel implementations.

5 Proposed image cipher scheme

The proposed approach is based on a dynamic key, DK , which is obtained by applying a secure cryptographic hash function such as SHA-512 on the mixing of a secret key SK and Nonce N_o . Then, the key is divided into 5 sub-keys as illustrated in Fig. 9, which are required to form three substitutions tables (S_1, S_2, S_3 , one permutation table, π_{BL} , and a selection index table, SI). For more details about the construction of the key-dependent substitution and permutation tables, readers are referred to our previous work presented in [38].

The proposed cipher scheme deals with flexible image sizes. If the number of bytes of an image is not a multiple of 32, a padding operation is required. Then, the input image is divided into N_B blocks $\{X_1, X_2, \dots, X_{N_B}\}$, where each has a size of $B = 32$ bytes. Furthermore, the cipher process is illustrated in Fig. 10, while its pseudo-code is described in Algorithm 2.

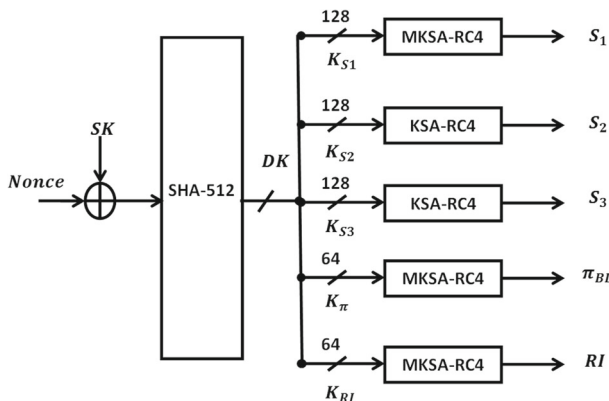


Fig. 9 Architecture of the proposed dynamic key derivation function

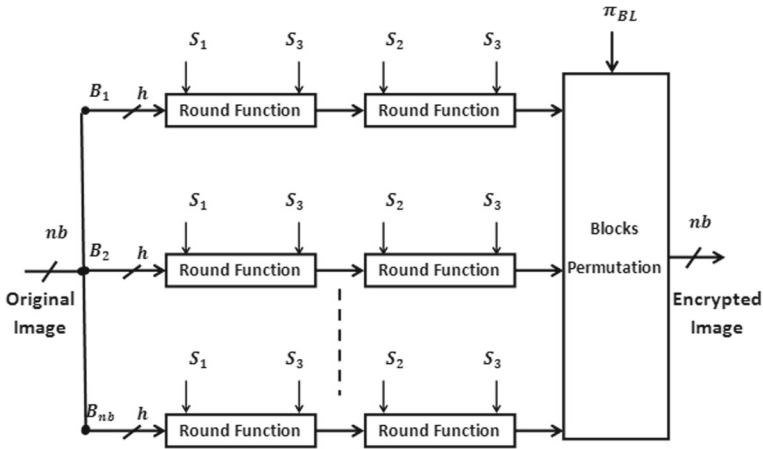


Fig. 10 Architecture of the proposed image encryption algorithm

Algorithm 2 Proposed encryption algorithm

Input: Three Substitution tables (S_1, S_2, S_3), pseudo-random selection index table SI , permutation table π_{blocks} ; the size of input block B and the plain image I .

Output: An encrypted image C .

```

1: procedure PROPOSED_ENCRYPTION( $I, S_1, S_2, S_3, SI, \pi_{blocks}, B$ )
2:    $Ml \leftarrow \text{reshape}(I, 1, 1 : L \times C \times P)$ 
3:    $len \leftarrow \text{length}(I)$ 
4:    $N_B \leftarrow \lceil \frac{len}{B} \rceil$ 
5:    $Ml \leftarrow \text{pad}(Ml, N_B \times B - len)$ 
6:    $X \leftarrow \text{reshape}(Ml, N_B, B)$ 
7:   for  $it \leftarrow 1$  to  $N_B$  do
8:      $ix1 \leftarrow (SI(it) : SI(it + h - 1))$  ▷ First Round Function
9:      $X \leftarrow X(it, 1 : B) \oplus S_1(ix1)$  ▷ Substitution using  $S_3$ 
10:     $Y \leftarrow S_3(X)$ 
11:     $Z \leftarrow \text{BinaryDiffusion}(Y)$ 
12:     $ix2 \leftarrow (SI(it - h + 1) : SI(it))$  ▷ Second Round Function
13:     $X \leftarrow Z \oplus S_2(ix2)$ 
14:     $Y \leftarrow S_3(X)$ 
15:     $Z \leftarrow \text{BinaryDiffusion}(Y)$ 
16:     $Tmp(it, 1 : B) \leftarrow Z$ 
17:  end for ▷ Order the encrypted blocks according to  $\pi_{blocks}$ .
18:   $Cp \leftarrow Tmp(\pi_{blocks}, 1 : h)$ 
19:   $C \leftarrow \text{reshape}(Cp, 1, nb \times h)$  ▷ Reshape the encrypted blocks to the original image size.
20:  return  $C$ 
21: end procedure

```

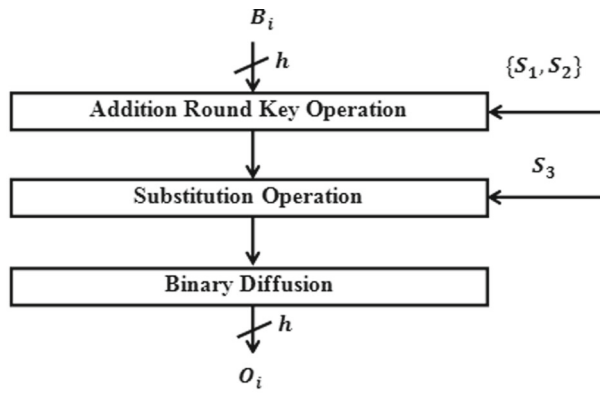
This scheme consists of two main layers as illustrated in Fig. 10. First, the proposed round function is iterated twice, which is sufficient to ensure the avalanche effect at the block level. Therefore, a single bit change of any input block will lead to a different

cipher block with a high percentage change (close to 50% at the bit level). Finally, a block permutation process is introduced to randomize the sequential order of the blocks.

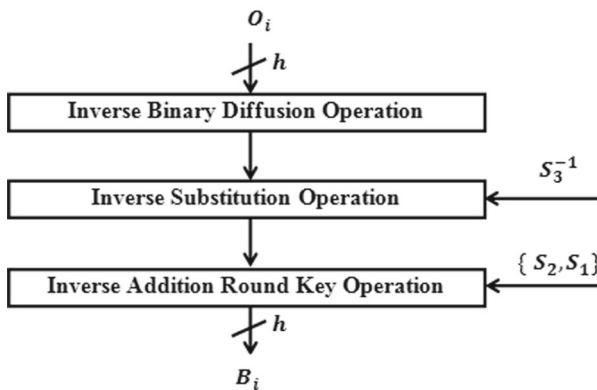
The round function and its corresponding inverse are shown in Fig. 11. The proposed round function consists of three operations: the addition round key (uses S_1 for the first iteration and S_2 for the second one, in addition to a selection table SI to control the generation of round block keys), substitution using S_3 , and a binary diffusion. The inverse round function has the same operations in a reverse order, and it uses the inverse substitution table, S_3^{-1} , and the inverse diffusion matrix.

The diffusion operation is based on the inverse binary diffusion matrix of [28] and it has a linear branch number equals to 10. Indeed, the proposed diffusion operation is based on a static binary mixing matrix G . While the addition round key, substitution and permutation operations are based on the generated S-boxes and P-box. For further details about the permutation and substitution construction and realization, readers are referred to our previous work [38].

S_1 and S_2 are used to generate the round block keys for the first and second iterations, respectively, while S_3 is used in the substitution process.



(a) Proposed Round Function



(b) Inverse Round Function

Fig. 11 Architecture of the proposed round function (a) and its corresponding inverse one (b)

Similarly, the decryption scheme consists of applying first an inverse block permutation process using the inverse of π_{BL} , π_{BL}^{-1} . In addition, the corresponding two rounds are applied in reverse order and they use the corresponding round block keys, inverse dynamic substitution table, S_3^{-1} , and the inverse binary diffusion matrix, G^{-1} .

The round block keys generation is designed to be lightweight, and the produced keys ensure a high level of randomness and uniformity as shown in the statistical tests (See Section 6.1).

6 Cryptanalysis of the proposed cipher scheme

In this section, all the required cryptographic metrics of the proposed cipher scheme are verified: high level of randomness, avalanche effect, independence, and high key sensitivity. The experiments were performed on the standard image “Lenna” and Zeros Image. The results, shown below, validate the robustness of the proposed approach. For more details about the description of the security and performance tests, readers are referred to our previous work [38]. Table 4 presents the statistical results of these tests.

6.1 Randomness

In order to be immune against statistical attacks, the proposed cipher scheme should exhibit specific random properties such as uniformity and independence [52].

The uniformity can be verified visually by plotting the Probability Density Function (PDF) of the encrypted message and it should be uniform. Original and encrypted “Lenna” with their corresponding PDFs are shown in Fig. 12. While original and encrypted zeros image (all elements are equal to 0) with their corresponding PDF are shown in Fig. 13. Figure 14 shows a 8×8 sub-matrix of the original Lenna image of Fig. 12a and their corresponding encrypted sub-matrix of Fig. 12c. As shown, all numerical values are changed, this ensures the randomness property. In addition, the PDF of the encrypted Lenna image is close to a uniform distribution with a value close to 0.039 ($\frac{1}{256}$), where 256 represents the number of possible symbols. Moreover, the entropy of the encrypted message should be close to its maximum value (In our case 8, since the data precision is “UINT-8”). We measure the entropy at the sub-matrix level to better quantify its level, where each sub-matrix has 256-byte elements. According to Fig. 15, the entropy of the encrypted sub-matrices of

Table 4 Statistical results of the proposed cipher scheme using Lenna image using and for 1000 random keys

Proposed Scheme	Min	Mean	Max	Std
<i>Dif</i>	49.8967	50.0021	50.1143	0.0349
<i>PS</i> (block level)	38.2813	50.0074	62.8906	3.1293
<i>KS</i>	49.8504	50.0021	50.1028	0.0356
<i>H – E</i>	7.1690	7.1749	7.1813	0.0016
ρ_h	–0.0516	0.0001	0.0482	0.0157
ρ_v	–0.0617	–0.0005	0.0529	.0157
ρ_d	–0.0503	–0.0001	0.0455	0.0158
PSNR	9.2026	9.2305	9.2592	0.0096
SSIM	0.0304	0.0359	0.0412	0.0017

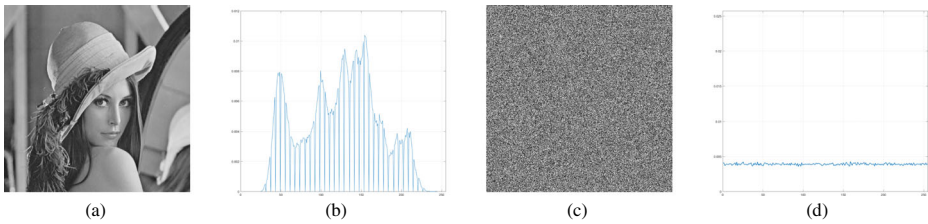


Fig. 12 **a** Original Lenna, **b** PDF of original Lenna with size $512 \times 512 \times 3$, **c** Encrypted Lenna, **d** PDF of encrypted Lenna using the proposed scheme

the gray and colored Lenna has a value close to 7.3, which is greater than that of the original sub-matrices and closer to the desired value of 8 as compared to the original ones.

On the other hand, the independence can be quantified by measuring the correlation between plain and encrypted messages, as well as the difference between the original and encrypted images; the independence value should be very close to zero.

Figure 16 shows the Empirical Cumulative Distribution Function (ECDF) of the correlation coefficient of the adjacent pixels of the encrypted “Lenna” image in horizontal, vertical and diagonal directions, with 1,000 random keys. We can see that the coefficient correlation of the adjacent pixels in encrypted Lenna images is very low and close to 0, which demonstrates that the proposed scheme eliminates the spatial redundancy. Similarly, the ECDF of the percent difference between original and encrypted Lenna images (at the bit level) is shown in Fig. 17. The results show a large difference between the original and encrypted images.

Consequently, the proposed cipher scheme ensures independence and uniformity, and hence it exhibits the required cryptographic randomness degree.

6.2 Visual degradation

Figure 18 shows the histogram of the obtained results of PSNR and SSIM between the original and the encrypted Lenna image for 1,000 dynamic keys. As shown, the PSNR has low values between [9.2, 9.26] with a mean equals to 9.23 dB. In addition, the SSIM values vary within [0.031, 0.041], which is very low and close to zero. Accordingly, a hard visual distortion is achieved and no useful information about the original image could be revealed from the encrypted image.

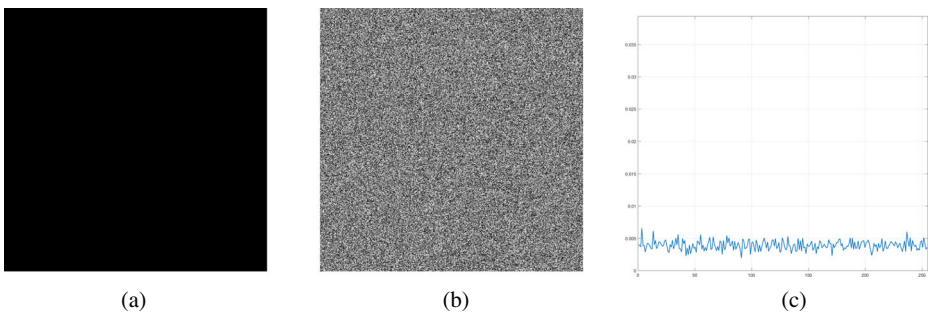


Fig. 13 **a** Zeros image with size $512 \times 512 \times 3$, **b** Encrypted Zeros image, **d** PDF of encrypted image using the proposed scheme

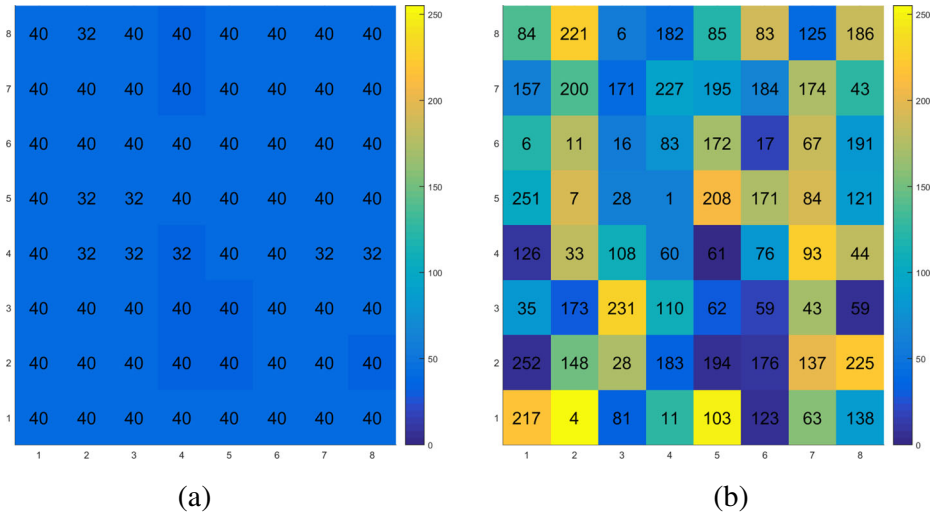


Fig. 14 **a** 8×8 cropped plain-image Lenna with its corresponding gray-scale matrix, **b** encrypted version of this matrix using the proposed cipher algorithm with its gray-scale value

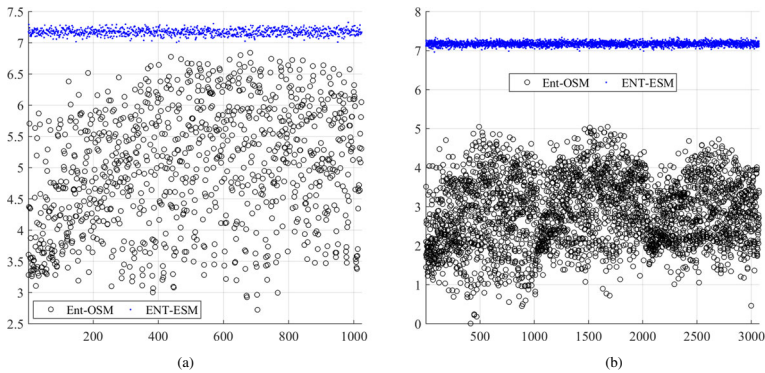


Fig. 15 **a** The variation of the entropy test on the original and encrypted sub-matrices of the gray **(a)** and color **(b)** Lenna images with size $512 \times 512 \times 3$ using the proposed scheme

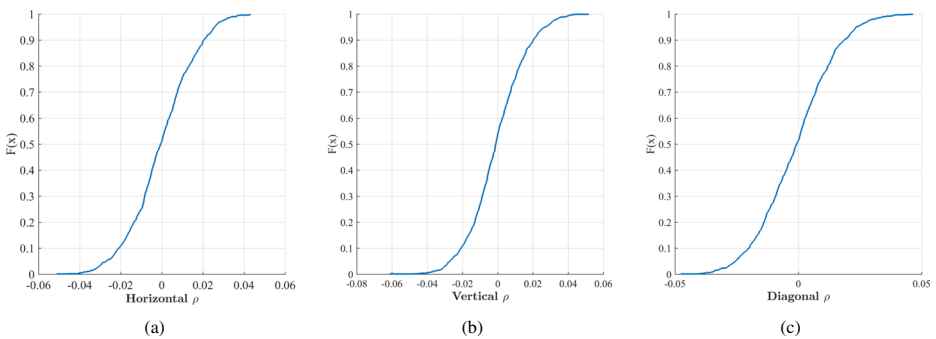


Fig. 16 ECDF of the correlation coefficient of adjacent pixels in ciphered Lenna images: **a** horizontally, **b** vertically and **c** diagonally, respectively using the proposed scheme

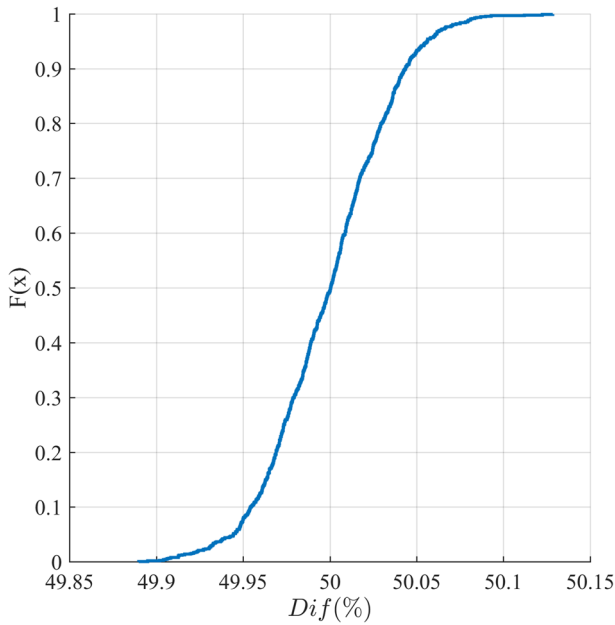


Fig. 17 ECDF of the difference test result (between original and encrypted Lenna images) using the proposed cipher scheme

6.3 Plain block sensitivity: avalanche effect

Figure 19 shows the ECDF results of the avalanche effect at the block level by using zero blocks with only one active bit. The results show that the low probabilities of the avalanche

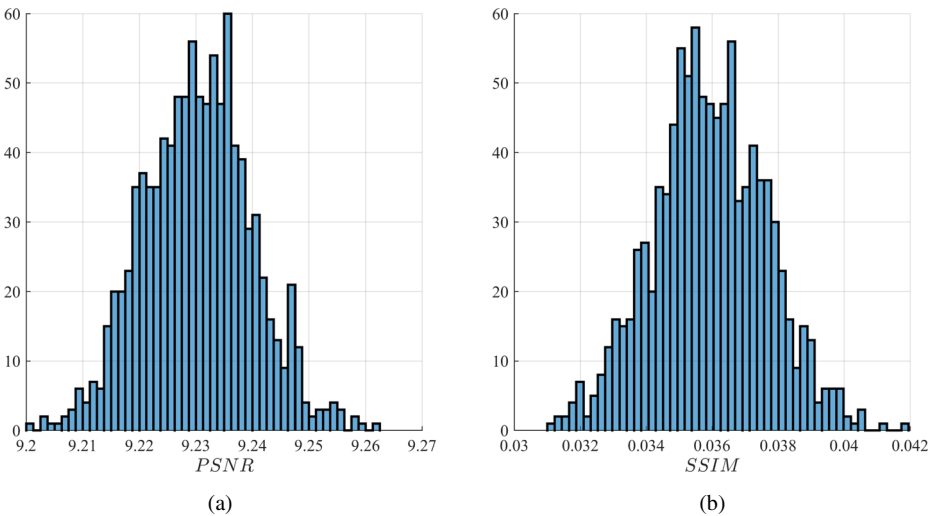


Fig. 18 Histogram of the obtained *PSNR* and *SSIM* (between the original and the encrypted Lenna images) for 1000 dynamic keys using the proposed scheme

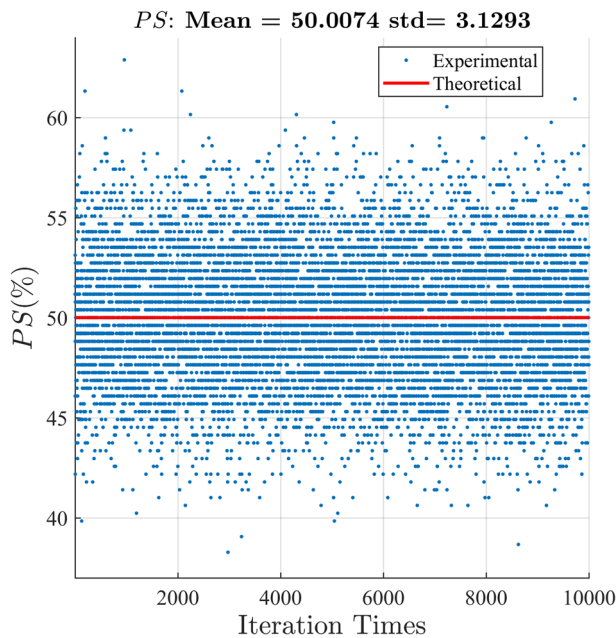


Fig. 19 The variation of the percent of the avalanche effect results using the proposed cipher scheme for 10000 random keys at the block level. The both chosen blocks have all elements equal to zeros, except the second block has only one byte that has only one active bit

effect are less than 45% and the minimum is close to 40%; while the high percentages are in the interval [47–52]. This clearly satisfies the avalanche effect at the block level.

6.4 Key sensitivity test

To avoid different kinds of key-related attacks, a sufficient large key size is required in addition to a high key sensitivity. Any slight difference in the key (usually one bit difference) must affect significantly the resultant encrypted image.

The ECDF of the key sensitivity test is illustrated in Fig. 20 against 1,000 random dynamic keys. The majority of values can be seen to be close to the optimal value of 50%. This proves the high key sensitivity and consequently, the ability of the proposed cipher to resist all key-related attacks. Moreover, a visual result is shown in Fig. 21 for a decrypted Lenna image with one-bit difference in the dynamic key. We can see that the decrypted image carries no useful information.

6.5 Security analysis of the proposed cipher scheme

In the following, well-known cryptanalytic analysis is performed to assess the immunity of the proposed cipher against such attacks. The proposed cipher scheme follows the Kerchoff's concept, where the structure is considered to be public and the cryptanalyst has a complete knowledge of the construction technique of the dynamic substitution and permutation tables and the employed static diffusion, but no knowledge about the secret key or the nonce, and hence no knowledge about the dynamic key.

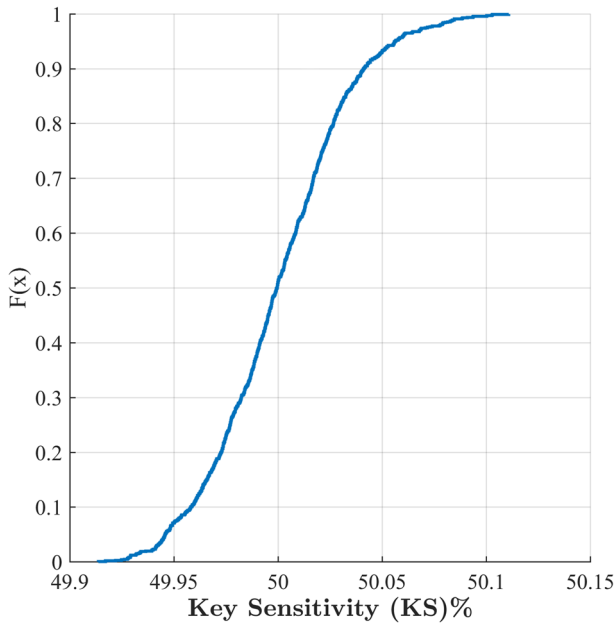


Fig. 20 ECDF of the Key sensitivity results using the proposed scheme for 1,000 random dynamic keys

6.5.1 Statistical attacks

The proposed scheme ensures a high degree of randomness since the uniformity and the independence were proved in Section 6.1 (PDF, entropy analysis; correlation test in addition to difference test). Therefore, the scheme can ensure a high resistance degree against statistical attacks.



Fig. 21 Decrypted Lenna image using the proposed scheme with its corresponding correct dynamic key **(a)** and with one bit error in the dynamic key used **(b)**

6.5.2 Weak keys attacks

In fact, the scheme is based on a dynamic approach (see Fig. 9) in contrast to the majority of the existing schemes that are based on a static approach. Furthermore, the dynamic key generation process leads to a set of dynamic keys with a high degree of randomness since a secure cryptographic hash function with maximum collision resistance is used, SHA-512. In addition, all cipher operations such as the generated substitution and permutation tables are related to this dynamic key. If any weakness in any dynamic key exists, it will not affect the previous and the next encrypted images. Therefore, the proposed cipher scheme exhibits a high resistance degree against weak keys and limits its impact. Moreover, the frequent variation of the secret key results in different sets of dynamic keys and consequently, prevents the accidental key disclosure.

6.5.3 Related key attacks: key sensitivity

As previously shown, the proposed cipher ensures a key and nonce sensitivity, (see Section 6.4). Also, the dynamic key has high sensitivity since all cipher primitives are related to it except the diffusion operation. As such, a change to these primitives leads to a different cipher image on the sender side or a different decrypted image at the receiver side with a 50% change. Thus, the proposed cipher exhibits a high resistance degree against related-key attacks.

6.5.4 Brute force attacks

The size of the secret key can be 128, 192, or 256 bits in a similar manner to AES, and the size of the dynamic key is 512 bits, which are sufficient enough to protect the proposed cipher against brute force attacks.

6.5.5 Chosen/known plaintext/ciphertext attacks

The proposed scheme, with its dynamic and static cipher primitives, exhibits excellent cryptographic performance. The cipher uses three dynamic substitution tables that ensure low linear and differential approximation probabilities, and it is strictly compliant to the avalanche criterion according to our previous work [38]. Equally important, the employed diffusion matrix of Koo et al. [28] ensures a linear branch equals to 10. Furthermore, the addition round keys along with the substitution operation are mandatory to avoid chosen/known plain/cipher text attacks as indicated previously.

The generation of addition round block keys is based on both substitution tables (S_1 and S_2). In fact, the blocks' permutation operation is introduced after the encryption of all blocks since the required round block keys are sequentially generated to reduce the computational cost of the generation process and to allow for a parallel implementation. However, an attacker cannot detect any sequential relation of the round block keys based on the proposed lightweight complex schemes of substitution and diffusion. It is very important to note that these primitives change for each new dynamic key generation, which prevents these kinds of attacks to collect the required set of plaintexts/ciphertexts.

6.5.6 Resistance against modern future powerful attacks

This paper benefits from the dynamic key approach to prevent different kinds of modern powerful attacks [41]. This can be justified since different cipher primitives are used for each dynamic key and consequently attackers cannot build their required information to break the proposed scheme. Let us indicate also that the maximum level of security can be reached by using a dynamic key for each input message.

7 Performance of the proposed cipher scheme

In this section, we assess the performance of the proposed cipher scheme in terms of execution time, memory consumption and error propagation. The experiments were conducted using the standard image of “Lenna”. The obtained results (below) validate the efficiency of the proposed approach.

7.1 Effect of error propagation

Note that since the proposed cipher can be considered as an ECB mode, thus, the effect of any bit error will only propagate to its corresponding block. Moreover, the limitation of ECB with identical blocks is solved since different keys are used for different blocks.

In Fig. 22, we show the effect of errors at the bit level (uniform random distribution within the encrypted image) on the proposed cipher scheme. The impact of errors on the proposed cipher scheme is low compared to NCIES (see Fig. 6), which suffers from high error propagation. In addition, visual results of the decrypted noisy images are shown in Fig. 23. This also validates the previous remarks by comparing to Fig. 7 of NCIES. Moreover, the effect of errors on the visual degradation of the proposed scheme is shown in Fig. 24. Indeed, by comparing it to Fig. 8 of NCIES, we can see a lower error propagation and consequently, lower visual degradation.

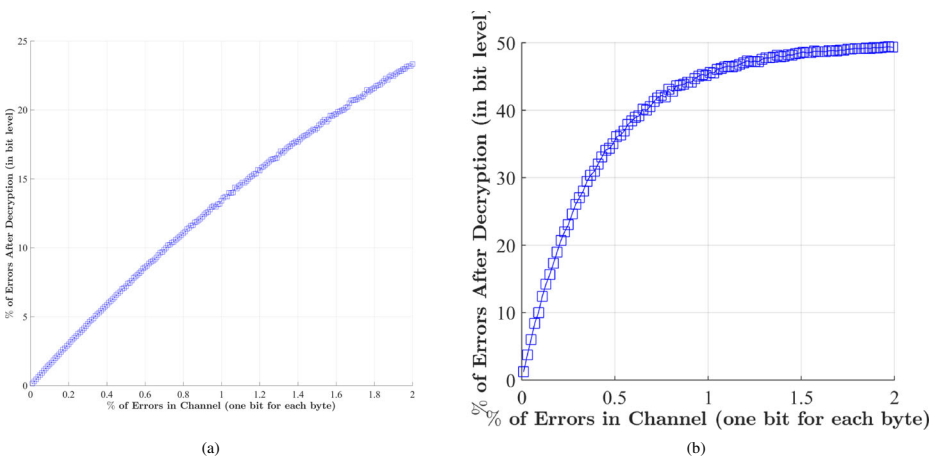


Fig. 22 Variation of the impact of the error propagation on the proposed cipher scheme (a) and of [16, 38] according to the percentage of errors. Introduced errors follows the uniform distribution, which is the worst case

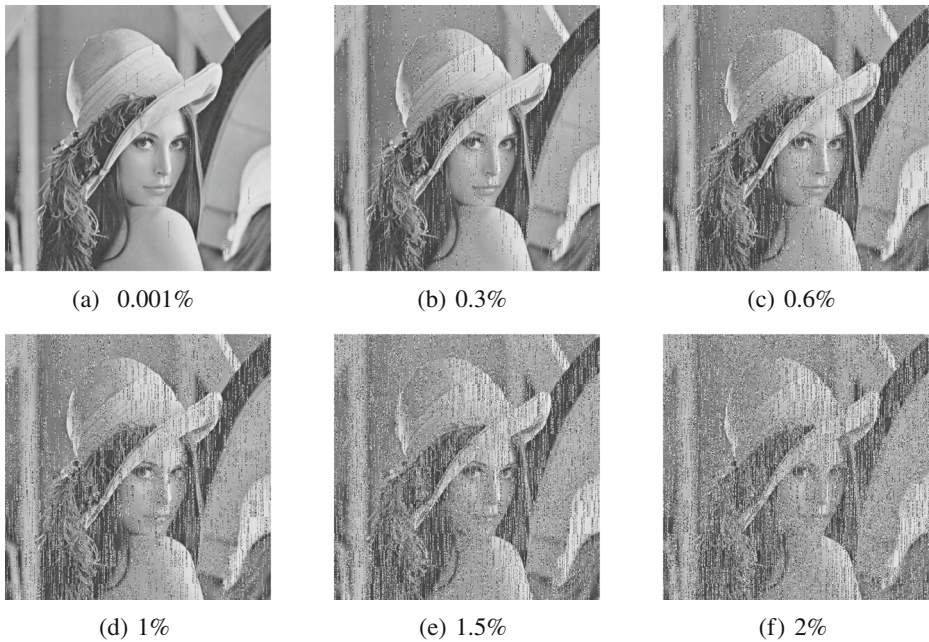


Fig. 23 Decrypted images in function of different errors percentage using the proposed scheme

In addition, for [16, 38], a bit error in any byte of the encrypted block C_i , will affect three blocks $\{X_{i-1}, X_i, X_{i+1}\}$ in the decrypted image. Two of them $\{X_i, X_{i+1}\}$ have random bit errors that occur independently in any bit position with an expected probability of $\frac{1}{2}$ and the third block X_{i-1} has only one specific bit error in the same bit-error position. While, for the proposed scheme, the effect of bit error introduces only random bit errors at the same block position with an expected probability of $\frac{1}{2}$ in the decrypted image.

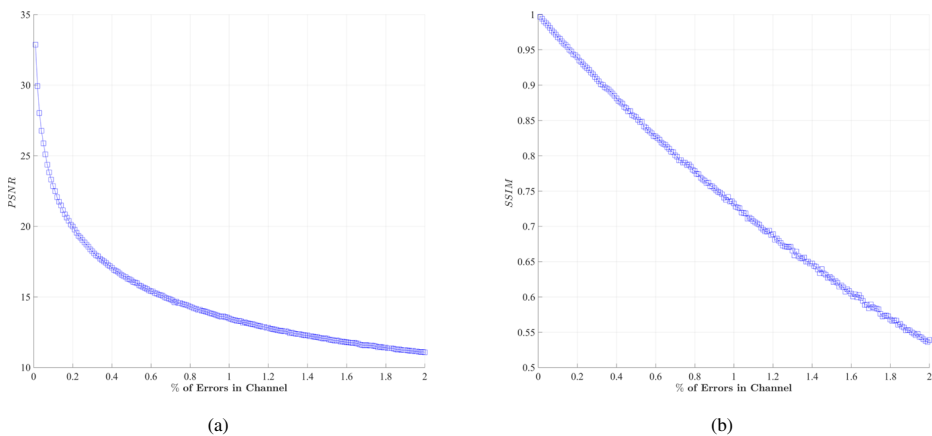


Fig. 24 The average variation of the *PSNR* and *SSIM* of the proposed cipher scheme versus the percentage of errors

Figure 22 shows the difference between both decrypted images versus the percentage of channel error (random with a uniform distribution) for both approaches as well as the proposed one. The results indicate that the error in the decrypted image for [16, 38] is close to 50% for a channel error $\geq 2\%$, while it requires 4% for the proposed approach. This indicates also the suitability of the proposed cipher in terms of error propagation as compared to [16, 38].

7.2 Space complexity

To encrypt one block, we only need the current block and not the whole image as in [15]. Hence, the total space complexity is reduced from $O(M \times N \times P \times 8)$ to $O(h)$, the space complexity for a block of size h .

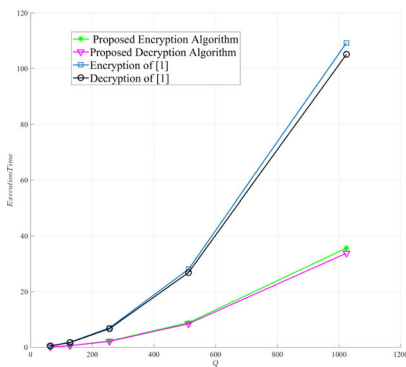
7.3 Execution time

The average calculation time to encrypt the standard Lenna image as a function of its size ($Q \times Q$) length is performed by using the proposed approach and of that of NCIES.

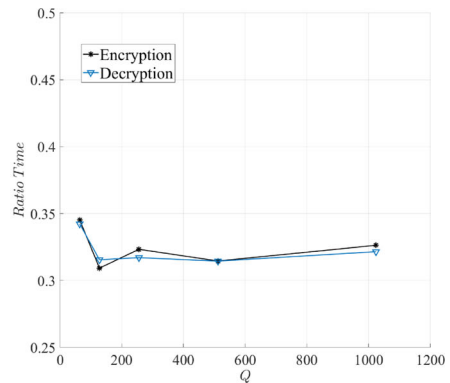
The time analysis is performed using the same machine with the same environment conditions. In addition, the employed software and hardware environments are: MATLAB R2015a simulator, 2 Intel(R) Xeon(R) CPU E5-2623 v4 @ 2.60GHz running Debian Linux. The required execution time is shown in Fig. 25a for both approaches and the ratio of the proposed one and the compared one is presented in Fig. 25b. First of all, the required execution time of the proposed scheme is approximately three times less when compared to the execution time of [15].

Accordingly, the proposed scheme reduces the required execution time while ensuring a high level of security (see Section 6.5). In conclusion, the proposed scheme is more efficient and can be suitable for real-time applications and limited devices.

In addition, we computed the ratio between the scheme of [16] and the proposed one with parallel computations for a different number of threads, n . First of all, the required execution time of the proposed scheme without parallelism is very close to the execution time of [16].



(a) Execution times between our approach and the approach of [7]



(b) Speed up of the new proposed key dependent dynamic block cipher

Fig. 25 Average of execution times between our approach and the NCIES approach in function of the size of the input image ($Q \times Q$) for 10000 iterations (a) and the corresponding speedup between the proposed solution and the NCIES approach

However, when applying parallelism, the results show that the proposed approach is faster compared to [16], which cannot be parallelized. Moreover, the results clearly indicate that increasing the number of threads (n) permits further reduction of the execution time. This makes the proposed scheme more suitable for real-time applications when compared to [16].

8 Conclusion

In this paper, the NCIES cipher, in addition to two lightweight image encryption schemes that require only two rounds, are analyzed from different aspects such as desirable cryptographic properties (avalanche effect, difference between original and cipher text, and uniformity), execution time and error propagation. The results showed that the NCIES scheme does not exhibit the claimed results concerning the avalanche effect, difference and uniformity. Consequently, the proposed round number leads to several cryptographic weaknesses. We showed that a chosen plain/cipher text attack can recover the permutation key with only two chosen messages in addition to the possibility of brute force attacks for a small size image. Furthermore, the effect of error propagation was studied and the results showed a high level of error propagation on the decryption side since no useful data could be obtained even for a single bit error. Therefore, the NCIES cipher is not suitable for wireless applications due to the high risk of data loss. Also, the analyzed cipher requires a substantial memory size and a high latency for the proper configuration of $r \geq 6$. These results are presented in order to verify the credibility and the safe employment of the NCIES cipher. In parallel, other related lightweight image encryption schemes, which use CBC in forward and backward directions, cannot be realized in parallel and they double the effect of error propagation compared to the original CBC in the forward direction only.

Moreover, we proposed an efficient cipher solution that overcomes all the stated challenges (weaknesses) of the analyzed ciphers. Finally, the work in this paper provides an overview for the design of modern, efficient, and secure lightweight image encryption schemes with a structure that combines static and dynamic keys.

Acknowledgements This paper is partially supported with funds from the Maroun Semaan Faculty of Engineering and Architecture at the American University of Beirut and also from the EIPHI Graduate School (contract “ANR-17-EURE-0002”).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

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. Alajel KM, Xiang W, Leis J (2010) Error resilience performance evaluation of h. 264 i-frame and jpw1 for wireless image transmission. In: 2010 4th international conference on signal processing and communication systems (ICSPCS). IEEE, pp 1–7
3. Alvarez G, Li S (2009) Cryptanalyzing a nonlinear chaotic algorithm (NCA) for image encryption. *Commun Nonlinear Sci Numer Simul* 14(11):3743–3749
4. 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
5. Beaulieu R, Shors D, Smith J, Treatman-Clark S, Weeks B, Wingers L (2015) Simon and speck: block ciphers for the internet of things. *IACR Cryptology ePrint Archive* 585

6. Biham E, Shamir A (1993) Differential cryptanalysis of the data encryption standard, vol 28. Springer, New York
7. Borghoff J, Canteaut A, Güneysu T, Kavun EB, Knezevic M, Knudsen LR, Leander G, Nikov V, Paar C, Rechberger C et al (2012) Prince—a low-latency block cipher for pervasive computing applications. In: *Advances in cryptology—ASIACRYPT 2012*. Springer, pp 208–225
8. Boriga R, Dăscălescu AC, Priescu I (2014) A new hyperchaotic map and its application in an image encryption scheme. *Signal Process Image Commun* 29(8):887–901
9. Borujeni SE, Eshghi M (2013) Chaotic image encryption system using phase-magnitude transformation and pixel substitution. *Telecommun Syst* 52(2):525–537
10. Chen J-X, Zhu Z-L, Chong F, Zhang L-B, Zhang Y (2015) An efficient image encryption scheme using lookup table-based confusion and diffusion. *Nonlinear Dyn* 81(3):1151–1166
11. Daemen J, Rijmen V (2002) The design of Rijndael: AES—the advanced encryption standard. Springer Science & Business Media
12. Dan T, Wang X (2008) Image encryption based on bivariate polynomials. In: *2008 international conference on computer science and software engineering*, vol 6. IEEE, pp 193–196
13. du Prel J-B, Hommel G, Röhrig B, Blettner M (2009) Confidence interval or p-value?: part 4 of a series on evaluation of scientific publications. *Deutsches Ärzteblatt Int* 106(19):335–339
14. Dworkin M (2016) Recommendation for block cipher modes of operation. NIST Spec Publ 800:38G
15. El Assad S, Farajallah M (2016) A new chaos-based image encryption system. *Signal Process Image Commun* 41:144–157
16. Fawaz Z, Noura H, Mostefaoui A (2016) An efficient and secure cipher scheme for images confidentiality preservation. *Signal Process Image Commun* 42:90–108
17. Flayh NA, Parveen R, Ahson SI (2009) Wavelet based partial image encryption. In: *International multimedia, signal processing and communication technologies, 2009. IMPACT'09*. IEEE, pp 32–35
18. Ghebleh M, Kalso A, Noura H (2014) An image encryption scheme based on irregularly decimated chaotic maps. *Signal Process Image Commun* 29(5):618–627
19. Granjal J, Monteiro E, Silva JS (2015) Security for the internet of things: a survey of existing protocols and open research issues. *IEEE Commun Surv Tutor* 17(3):1294–1312
20. Gueron S (2009) Intel's new aes instructions for enhanced performance and security. In: *FSE*, vol 5665. Springer, pp 51–66
21. Guo J, Peyrin T, Poschmann A, Robshaw M (2011) The LED block cipher. In: *Cryptographic hardware and embedded systems—CHES 2011*. Springer, pp 326–341
22. Gutub AA-A, Khan FA-A (2012) Hybrid crypto hardware utilizing symmetric-key and public-key cryptosystems. In: *2012 international conference on advanced computer science applications and technologies (ACSAT)*. IEEE, pp 116–121
23. Gutub A, Al-Juaid N, Khan E (2017) Counting-based secret sharing technique for multimedia applications. *MTAP*, pp 1–29, ISSN 1573-7721
24. Hong D, Lee J-K, Kim D-C, Kwon D, Ryu KH, Lee D-G (2014) LEA: a 128-bit block cipher for fast encryption on common processors. In: *Information security applications*. Springer, pp 3–27
25. 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
26. 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
27. Janakiraman S, Thenmozhi K, Rayappan JBB, Amirtharajan R (2018) Lightweight chaotic image encryption algorithm for real-time embedded system: implementation and analysis on 32-bit microcontroller. *Microprocess Microsyst* 56(Supplement C):1–12
28. Koo BW, Jang HS, Song JH (2006) On constructing of a 32×32 binary matrix as a diffusion layer for a 256-bit block cipher. Springer, Berlin, pp 51–64
29. Kumar A, Ghose MK (2011) Extended substitution-diffusion based image cipher using chaotic standard map. *Commun Nonlinear Sci Numer Simul* 16(1):372–382
30. Kumar M, Pal SK, Panigrahi A (2014) FeW: a lightweight block cipher. *IACR Cryptology ePrint Archive* 326
31. Laiphrakpam DS, Khumanthem MS (2017) A robust image encryption scheme based on chaotic system and elliptic curve over finite field. *Multimed Tools Appl* 77(7):8629–8652
32. 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
33. Li C, Chen MZQ, Lo K-T (2011) Breaking an image encryption algorithm based on chaos. *Int J Bifurcation Chaos* 21(7):2067–2076
34. Massoudi A, Lefebvre F, De Vleeschouwer C, Macq B, Quisquater J-J (2008) Overview on selective encryption of image and video: challenges and perspectives. *EURASIP J Inf Secur* 2008:5

35. McKay KA, Bassham LE, Turan MS, Mouha NW (2017) Report on lightweight cryptography. NIST Interagency/Internal Report (NISTIR)-8114
36. Mondal B, Mandal T (2017) A light weight secure image encryption scheme based on chaos & dna computing. *J King Saud University - Comput Inf Sci* 29(4):499–504
37. Noura H, Martin S, Al Agha K, Erss-rinc KC (2014) Efficient and robust secure scheme for random linear network coding. *Comput Netw* 75:99–112
38. Noura H, Sleem L, Noura M, Mansour MM, Chehab A, Couturier R (2017) A new efficient lightweight and secure image cipher scheme. *Multimedia Tools and Applications*
39. O'Melia S, Elbirt AJ (2010) Enhancing the performance of symmetric-key cryptography via instruction set extensions. *IEEE Trans Very Large Scale Integr VLSI Syst* 18(11):1505–1518
40. Paar C, Pelzl J (2009) *Understanding cryptography: a textbook for students and practitioners*. Springer Science & Business Media
41. Pradeep LN, Bhattacharjya A (2013) Random key and key dependent s-box generation for aes cipher to overcome known attacks. In: *International symposium on security in computing and communication*. Springer, pp 63–69
42. 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
43. Rubio-González C, Gunawi HS, Liblit B, Arpacı-Dusseau RH, Arpacı-Dusseau AC (2009) Error propagation analysis for file systems. In: *ACM sigplan notices*, vol 44. ACM, pp 270–280
44. Seyedzade SM, Mirzakuchaki S, Atani RE (2010) A novel image encryption algorithm based on hash function. In: *2010 6th Iranian machine vision and image processing (MVIP)*. IEEE, pp 1–6
45. Seyedzadeh S, Mirzakuchaki S (2012) A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map. *Signal Process* 92(5):1202–1215
46. Shibutani K, Isoke T, Hiwatari H, Mitsuda A, Akishita T, Shirai T (2011) Piccolo: an ultra-lightweight blockcipher. In: *Cryptographic hardware and embedded systems—CHES 2011*. Springer, pp 342–357
47. Suzaki T, Minematsu K, Morioka S, Kobayashi E (2013) TWINE: a lightweight block cipher for multiple platforms. In: Knudsen LR, Wu H (eds) *Selected areas in cryptography*, volume 7707 of *lecture notes in computer science*. Springer, Berlin, pp 339–354
48. 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
49. VanVoorhis CRW, Morgan BL (2007) Understanding power and rules of thumb for determining sample sizes. *Tutor Quant Methods Psychol* 3(2):43–50
50. Wadi SM, Zainal N (2014) High definition image encryption algorithm based on aes modification. *Wirel Pers Commun* 79(2):811–829
51. Wu W, Zhang L (2011) LBlock: a lightweight block cipher. In: *Applied cryptography and network security*. Springer, pp 327–344
52. Xu S, Wang Y, Wang J, Tian M (2008) Cryptanalysis of two chaotic image encryption schemes based on permutation and xor operations. In: *International conference on computational intelligence and security*, 2008. CIS'08, vol 2. IEEE, pp 433–437
53. 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
54. Zhu Z-L, Zhang W, Wong K-Wo, Hai Y (2011) A chaos-based symmetric image encryption scheme using a bit-level permutation. *Inf Sci* 181(6):1171–1186



Hassan Noura is a researcher in the department of Electrical and Computer Engineering of the American University of Beyrouth (AUB), Lebanon.



Ali Chehab is a professor in the department of Electrical and Computer Engineering of the American University of Beyrouth (AUB), Lebanon.



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



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



Mohammad M. Mansour is a professor in the department of Electrical and Computer Engineering (AUB), Lebanon.