Efficient Chaotic Encryption Scheme with OFB mode

Hassan Noura, Christophe Guyeux, Ali Chehab, Mohammad Mansour, and Raphaël Couturier

April 7, 2023

Abstract

Data confidentiality is mandatory during transmission or when storing sensitive information, especially in financial, medical and military applications. In this context, several cipher solutions and techniques have been presented in the literature. However, existing solutions are mainly based on static structures, where the confusion and diffusion primitives are fixed and independent of the secret key. In this article, we propose a new block cipher scheme that is based on the Substitution-Permutation Networks (SPN). The proposed cipher consists of three operations: round-key addition, substitution, and bits' permutation. Moreover, the substitution operation is applied at the byte level and it is based on a dynamically generated S-box, while the diffusion primitives are applied at the bit level using a dynamically generated P-box. Such key-dependent design ensures better cryptographic strength and system performance when compared, for instance, to DES, 3DES, RC5, and PRESENT schemes, among others, due to its key expansion algorithm. Thorough analysis show that the proposed scheme exhibits a high degree of randomness, key and plain-text sensitivity, and it satisfies the avalanche effect. From a theoretical perspective, we have formulated the Output Feedback mode of operation as a discrete dynamical system on a topological space. We prove that the dynamics of this system (in terms of sensitivity to the initial vector, etc.) are directly related to the strong connectivity of a graph. By doing so, we are able to characterize the conditions under which this mode evolves chaotically, as defined in the Devaney's theory. In particular, such theoretical investigations allow us to link the avalanche effect and key sensitivity of the cipher with the sensitivity of the whole process, that is, with the mode of operation.

Keywords: Block ciphers; Dynamic P-boxes and S-boxes; Security analysis; Devaney's chaos; Lyapunov exponent.

1 Introduction

Typically, encryption algorithms are used to ensure the confidentiality of data either during transmission or while being stored. Originally, encryption algorithms were based on symmetric keys such as DES [10], which is based on Feistel Networks, and AES [12], which is based on Substitution Permutation Networks (SPN) (cf. [31]). These traditional techniques have static structures; the confusion and diffusion primitives are fixed and independent of the secret key. As such, a large number of rounds is required to achieve the desired security level. This is associated with an increased level of computational complexity and hence, such techniques may not be suitable for real-time applications with high data rates [28].

An efficient encryption algorithm may satisfy the uniformity property and it may exhibit high sensitivity with respect to plain-text and secret key while encrypting one data block at a time (ECB mode); however, this may not be the case for different modes of operation. For example, the Avalanche effect might be achieved after encrypting one block, however, such property might not be achieved after encrypting the whole message using a specific mode of operation. Hence, it is important to have new, flexible, and practicable block ciphers that are well studied with respect to the different modes of operation. In this paper, we aim to present new contributions in these two directions; we propose a new block cipher scheme that is flexible and secure, and we provide a new theoretical approach to assess whether or not the properties of the block cipher are preserved when applying any mode of operation.

Accordingly, the contributions in this article are two fold. On one hand, a new encryption scheme is proposed and thoroughly studied, and it can be used, for instance, in the Output Feedback (OFB) mode. On the other hand, the dynamics of this mode are studied using tools derived from mathematical topology. More specifically, the proposed cipher scheme satisfies the fundamental security properties such as the avalanche effect, key sensitivity, and randomness, as well as a configurable number of rounds. It includes an efficient technique to construct dynamic substitution S-boxes and permutation P-boxes. The technique is based on a dynamic key that is generated using the secret key and an initial vector. The proposed approach is flexible and the size of input blocks can be varied according to the applications' requirements.

On the theory side, we formulate the OFB mode as a discrete dynamical system based on a relevant metric space. Then, we prove that the dynamics of this system (in terms of sensitivity to the initial vector, chaotic behavior, etc.) are directly related to the strong connectivity property of a well-defined graph. Such theoretical investigation allows us to link the avalanche effect and key sensitivity of the algorithm to the sensitivity of the mode of operation.

The remainder of this article is organized as follows. In the next section, we provide some background about symmetric ciphers in general, and our previously obtained results about the dynamics of the CBC mode of operation. In Section 3, the proposed data encryption algorithm is presented in details. The cryptographic performance is discussed in Section 4, and experimental evaluation is presented in the following section. In particular, we show how to reinforce the links between the avalanche and sensitivity properties of the cipher function and the whole process using the OFB mode of operation. The dynamics of the OFB mode of operation are theoretically investigated in Section 7, in which we provide a characterization that shows sensitive dependence to the initial conditions. This research work ends by a conclusion, in which the contribution is summarized and intended future work is outlined.

2 Background

In this paper, we investigate the conditions under which some modes of operations behave chaotically and then, we propose a symmetric cipher that satisfies them. We start with some background about symmetric ciphers and chaotic behavior.

2.1 Symmetric ciphers

Encryption algorithms are classified into two main classes: symmetric and asymmetric ciphers. In symmetric ciphers, only one key is secretly exchanged between the transmitter and the receiver, and it is used in the encryption and decryption processes. On the other hand, two keys are used in asymmetric ciphers; a sender having a public key can encrypt a message and only one receiver, the one having the corresponding private key, can decrypt such a message. However, asymmetric ciphers are not appropriate in various situations due to their inherent expensive computational complexity and memory usage, when compared to symmetric ciphers.

When using symmetric ciphers, one can select a stream cipher whereby the data is encrypted at the bit or byte level, or a block cipher whereby the data is encrypted one block (a set of bytes) at a time. Block ciphers divide the plain-text message into separate blocks of fixed size, e.g., 64 bits for DES (Data Encryption Standard) and 128 bits for AES (Advanced Encryption Standard), and encrypt the blocks according to some specific mode of operation such as CBC (see Figure 1), OFB, or CTR [15]. Existing block ciphers are mainly based on two different kinds of round functions: Substitution-Permutation Network (SPN) or the Feistel Network (FN). For the same security level, SPN has a better performance and requires a lower number of rounds compared to FN.

A block cipher takes as input the message data blocks and the key, and applies a function for several rounds. According to Shannon, a strong round function should achieve two main properties, confusion and diffusion, when applying substitution and permutation operations [27, 18]. Substitution makes the relationship between the ciphertext and the key obscure, and it ensures the confusion property; it may use one S-box such as the case of AES or several S-boxes such as the case of DES. The diffusion property ensures that any change of a bit in the plaintext is spread over many ciphertext symbols. This permits to hide statistical properties of the plaintext. This property is achieved by us-

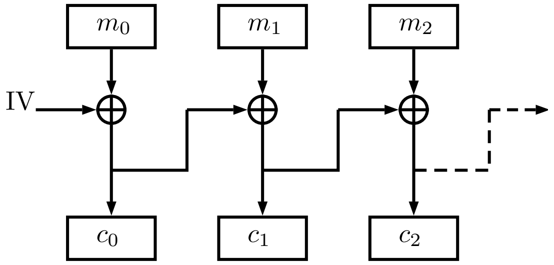


Figure 1: CBC mode of operation

ing permutation P-boxes like the case of DES or by considering a Maximum Distance Separable (MDS) matrix like the case of AES.

Block ciphers can be split into two sub-categories according to the way substitution and permutation are implemented. The first sub-category uses a static structure that minimizes, after a certain number of rounds, the maximum difference propagation probability (against differential attacks) and the maximum input-output correlation probability (against linear attacks). Furthermore, a standard block cipher uses static S-boxes with maximum performance to ensure the confusion property, while P-boxes (as in DES) or static diffusion matrix (MDS matrix as in AES in addition to row rotation) to ensure the diffusion property. Moreover, a key expansion algorithm with a secret key as input is iterated to produce the required round keys (e.g., AES, SAFER [19], 3-WAY [11]). The second sub-category of block ciphers has a dynamic structure reflected in the construction of S-boxes and P-boxes using the secret key. The advantage of the static approach is that its security against differential and linear attacks can be proved. However, it requires several encryption rounds in order to achieve the required security level, i.e., more computational complexity and consequently higher execution time compared to the dynamic approach that can ensure a similar security level with a lower number of rounds. Also, the static block cipher structure gives opportunities to potential attacks since the substitution and diffusion layers are independent from the secret key and are known to the attacker.

2.2 Chaotic properties of the CBC mode of operation

Consider the CBC mode of operation with a keyed encryption function $\varepsilon_\mathsf{k} : \mathbb{B}^\mathsf{N} \to \mathbb{B}^\mathsf{N}$, where N is the size of the block cipher, and $\mathcal{D}_\mathsf{k} : \mathbb{B}^\mathsf{N} \to \mathbb{B}^\mathsf{N}$ is the corresponding decryption function such that $\forall \mathsf{k}, \varepsilon_\mathsf{k} \circ \mathcal{D}_\mathsf{k}$ is the identity function. Using the same canvas as in the case of pseudo-random number generation [6] or hash functions [7, 16], we use the following Cartesian product that was defined in [1]: $\mathcal{X} = \mathbb{B}^\mathsf{N} \times \mathcal{S}_\mathsf{N}$, where $\mathbb{B} = \{0, 1\}$ is the set of Boolean values, while $\mathcal{S}_\mathsf{N} = [\![0, 2^\mathsf{N} - 1]\!]^\mathbb{N}$ stands for the set of sequences of natural integers bounded by $2^\mathsf{N} - 1$ (or in other words, the set of N-bit block messages). As such, the $\mathcal{X}_\mathsf{N}$ product is defined over the internal states of the mode of operation and the sequences of block messages. Let us consider the initialization function

$$\begin{array}{cccc} i: & \mathcal{S}_\mathsf{N} & \longrightarrow & \llbracket 0, 2^\mathsf{N}-1 \rrbracket \\ & (m^i)_{i\in\mathbb{N}} & \longmapsto & m^0 \end{array}$$

that returns the first block of a message, and the shift function

$$\begin{array}{cccc} \sigma: & \mathcal{S}_\mathsf{N} & \longrightarrow & \mathcal{S}_\mathsf{N} \\ & (m^0, m^1, m^2, ...) & \longmapsto & (m^1, m^2, m^3, ...) \end{array}$$

which removes the first block of a message. Let m_j be the j-th bit of a message block $m \in \llbracket 0, 2^\mathsf{N}-1 \rrbracket$ expressed in the binary numeral system, and when counting from the left. We have defined

$$\begin{array}{cccc} F_f: & \mathbb{B}^\mathsf{N} \times \llbracket 0, 2^\mathsf{N}-1 \rrbracket & \longrightarrow & \mathbb{B}^\mathsf{N} \\ & (x, m) & \longmapsto & \left(x_j m_j + f(x)_j \overline{m_j}\right)_{j=1..\mathsf{N}} \end{array}$$

This function returns the input binary vector x, whose m_j-th components x_{m_j} have been replaced by $f(x)_{m_j}$, for all $j = 1..\mathsf{N}$ such that $m_j = 0$. When f is the vector negation, this function will correspond to a simple XOR between the plain-text and the previous encrypted state.

Denote by f_0 the vector negation. So the CBC mode of operation can be rewritten as

$$\left\{ \begin{array}{ll} X^0 = & (IV, m) \\ X^{n+1} = & \left(\mathcal{E}_k \circ F_{f_0}\left(i(X_1^n), X_2^n\right), \sigma(X_1^n)\right) \end{array} \right. \tag{1}$$

For any given $g : \llbracket 0, 2^\mathsf{N}-1 \rrbracket \times \mathbb{B}^\mathsf{N} \longrightarrow \mathbb{B}^\mathsf{N}$, let us define $G_g(X) = (g(i(X_1), X_2); \sigma(X_1))$ (when $g = \mathcal{E}_k \circ F_{f_0}$, we obtain one cipher block of the CBC, as depicted in Figure 1). So the recurrent relation of Eq.(1) can be rewritten in a condensed form,

$$X^{n+1} = G_{\mathcal{E}_k \circ F_{f_0}}\left(X^n\right). \tag{2}$$

With such a notation, one iteration of the above discrete dynamical system corresponds exactly to one cipher block in the CBC mode of operation.

Next, we define a distance $d((x, m); (\check{x}, \check{m})) = d_e(x, \check{x}) + d_m(m, \check{m})$ on $\mathcal{X}_\mathsf{N}$, where [2]

$$\left\{ \begin{array}{ll} d_e(x, \check{x}) & = \sum_{k=1}^{\mathsf{N}} \delta(x_k, \check{x}_k) \\ & \\ d_m(m, \check{m}) & = \dfrac{9}{\mathsf{N}} \displaystyle\sum_{k=1}^{\infty} \dfrac{\sum_{i=1}^{\mathsf{N}} \left|m_i^k - \check{m}_i^k\right|}{10^k}. \end{array} \right.$$

in which $\delta(x, y) = 1$ if $x = y$, else it is 0. Using such a model, we proved the following theorem in [2].

Theorem 1 *Consider the directed graph $\mathcal{G}_g$, where:*

- *vertices represent all the possible N-bit words.*

- *there is an edge $m \in [\![0, 2^{\mathsf{N}}-1]\!]$ from x to $\check{x}$ if and only if $g(m,x) = \check{x}$, where $g = \mathcal{E}_k \circ F_{f_0}$.*

So, if $\mathcal{G}_g$ is strongly connected, then the CBC mode of operation $G_{\mathcal{E}_k \circ F_{f_0}}$ is chaotic, as defined by Devaney [14], on the topological space $(\mathcal{X}, d)$. This means that $G_{\mathcal{E}_k \circ F_{f_0}}$ satisfies on $(\mathcal{X}, d)$ the following properties:

- regularity*: the set of periodic points is dense in $\mathcal{X}_{\mathsf{N}}$ (for any point x in $\mathcal{X}_{\mathsf{N}}$, any neighborhood of x contains at least one periodic point).*

- topological transitivity*: for any pair of open sets $U, V \subset \mathcal{X}_{\mathsf{N}}$, there exists an integer $k > 0$ such that $G^k_{\mathcal{E}_k \circ F_{f_0}}(U) \cap V \neq \varnothing$.*

- sensitive dependence on initial conditions*: there exists $\delta > 0$ such that, for any $x \in \mathcal{X}_{\mathsf{N}}$ and any neighborhood V of x, there is $y \in V$ and $n > 0$ such that $d\left(G^n_{\mathcal{E}_k \circ F_{f_0}}(x), G^n_{\mathcal{E}_k \circ F_{f_0}}(y)\right) > \delta$.*

This result has been extended in [3], where both expansivity and sensibility constants of symmetric ciphers have been evaluated in the case of the CBC mode of operation. However, all these results of qualitative and quantitative disorder have been stated on an exotic phase space $\mathcal{X}_{\mathsf{N}}$, equipped with a distance d, very different from the usual Euclidian one. Our objective was then to translate them into a more typical situation, namely the real line equipped with its usual order topology. Using an ad hoc semi-conjugacy, we have then established that the CBC mode $G_{\mathcal{E}_k \circ F_{f_0}}$ on the phase space $\mathcal{X}_{\mathsf{N}}$ consists of simple iterations of a well defined function g over $\mathbb{R}$. Additionally, g has derivatives of all orders over $\left[0, 2^N\right[$, except at the points of the set $\left\{\dfrac{n}{\mathsf{N}} \;/\; n \in [\![0; 2^{\mathsf{N}} \times \mathsf{N}]\!]\right\}$. Furthermore, on each interval of the form $\left[\dfrac{n}{\mathsf{N}}, \dfrac{n+1}{\mathsf{N}}\right[$, with $n \in [\![0; 2^{\mathsf{N}} \times \mathsf{N}]\!]$, g is a linear function, having a slope equal to N: $\forall x \notin I, g'(x) = \mathsf{N}$.

Let us finally recall that, for $f : \mathbb{R} \longrightarrow \mathbb{R}$, the *Lyapunov exponent* of the system defined by $x^0 \in \mathbb{R}$ and $x^{n+1} = f(x^n)$ is

$$\lambda(x_0) = \lim_{n \to +\infty} \frac{1}{n} \sum_{i=1}^{n} \ln \left| \; f'\left(x^{i-1}\right) \right|.$$

This Lyapunov exponent can be computed for the CBC mode of operation.

Theorem 2 *Consider the CBC mode of operation g with block size of N. Then, $\forall x^0 \in \mathcal{L}$, its Lyapunov exponent $\lambda(x^0)$ is equal to $\ln(\mathsf{N})$.*

Proof The proof of this result can be found in [1].

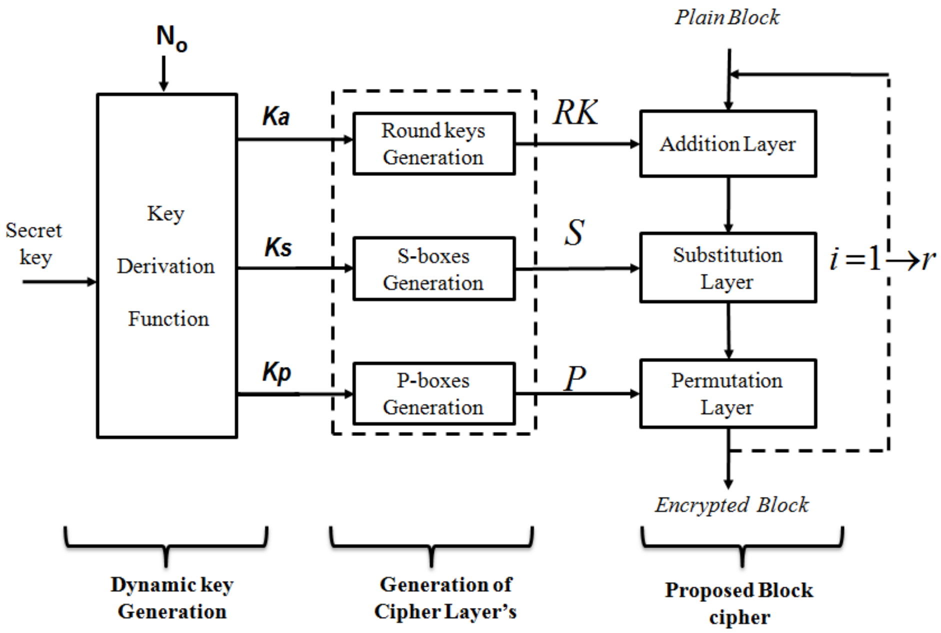


Figure 2: Structures of the proposed cipher scheme (without dynamic IV).

3 The proposed data encryption

3.1 General overview

The general structure of the proposed SPN is shown in Figure 2 for the encryption process. This approach consists of three phases:

1. Dynamic key generation;
2. Cipher layers construction;
3. Encryption/Decryption algorithm.

The encryption algorithm is a block cipher consisting of a round function that iterates r times, where the round function consists of three different layers: addition, substitution, and permutation. The number of rounds, r, is related to the avalanche effect property, which depends on the block size, as will be described later in Subsection 5.2.1 and illustrated in Table 2 and Figure 13. The decryption scheme is similar to the encryption one, but operates in reverse order and replaces the permutation and substitution primitives by their inverse counterparts. Moreover, the proposed encryption/decryption process uses different S-boxes and P-boxes for each round.

In the first phase, the proposed key derivation function generates a dynamic key, which is then divided into three dynamic sub-keys Ka, Kp, and Ks, whereby Ka is used to generate the set of rounds keys, while Ks and Kp are used to build the S-boxes and the P-boxes, respectively. The round keys, in addition to the produced S-boxes and P-boxes, constitute the cryptographic primitives of the proposed encryption scheme.

The first operation in the round function is the addition process, which uses two operations, exclusive-or (XOR) and addition modulo 256, to mix the byte values from the plain block with a round key that is generated based on the chaotic tent map. Then, we perform substitution of the plain block byte values. Next, we perform the permutation operation over the bits of the block; these operations (substitution and permutation) achieve the confusion and diffusion properties. Repeating the same addition-substitution-permutation operations for r rounds using variable dynamic keys and parameters offers a high-security level to the cipher scheme. Note that the S-boxes are sub-blocks of 8×8 bits, which means substitution is applied on elements of 8 bits that will be replaced by another 8 bits (byte) by using the substitution table.

In the following, we describe the generation of the dynamic keys and the construction of the cipher layers.

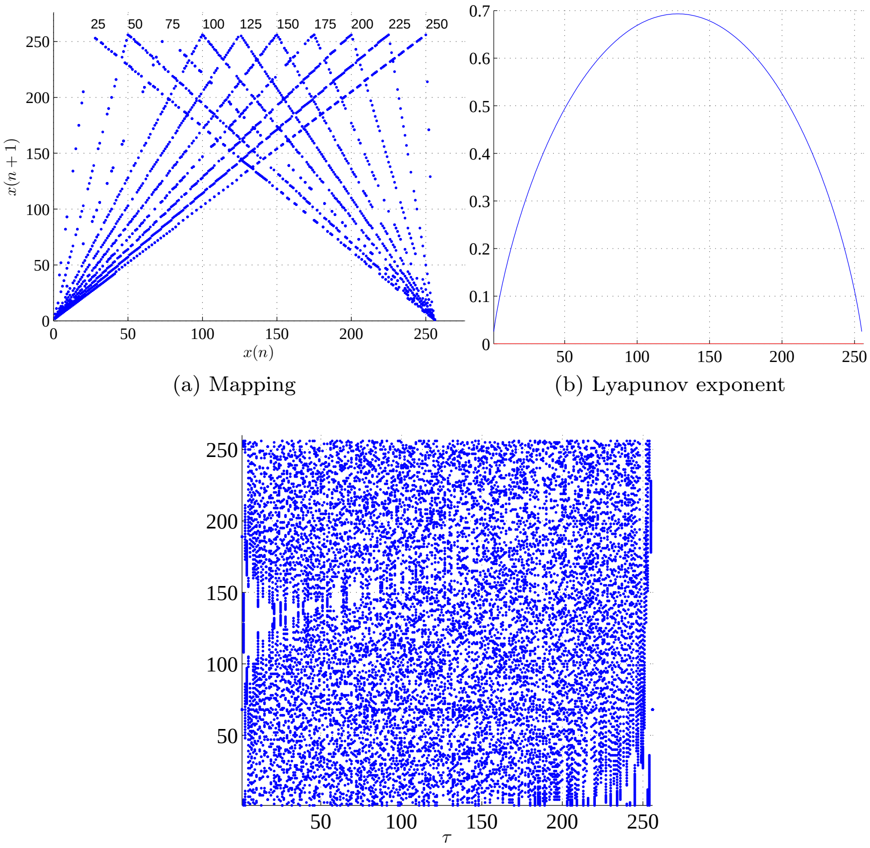
3.2 Integer Non-linear finite skew tent

The integer finite skew tent transformation is reformulated as in [5, 20]. The different cipher primitives such as permutation and substitution tables in addition to round keys are totally dependent on this transformation. Thus, we analyze it in details to show that it provides the required cryptographic strength. This map is defined as

$$y = \begin{cases} \lceil \frac{Q}{\tau_i} \times x \rceil & x \leq \tau \\ \\ \lfloor \frac{Q}{(Q-\tau_i)} \times (Q-x) \rfloor + 1 & x > \tau \end{cases} \tag{3}$$

where $x,\ y,\ \tau_i \in \{1, \ldots, Q\}$, and $Q = 2^8$ when the input message is in byte representation. In addition, τ is the control parameter, while x and y are respectively the input and output of this transformation. This is a piece-wise linear transformation, which is composed of two linear segments as shown in Figure 3-a (for different values of τ). It has good dynamical properties according to its corresponding Lyapunov exponent, which is positive for any given $x(0) \in \{1, ..., 256\}$, as depicted in Figure 3-b. The corresponding bifurcation diagram for $\tau \in \{1, ..., 256\}$ is shown in Figure 3-c.

Additionally, we analyzed the effects of computing precision of this transformation and we obtained the periodicity of all parameters $\tau \in \{1, ..., 256\}$, as represented in Figure 4. This result clearly indicates that the transformation cannot provide stable periodicity: this stability is large for some parameters and it may be very low for other ones. As such, we can detect that employing fixed



(a) Mapping (b) Lyapunov exponent

(c) Bifurcation Diagram

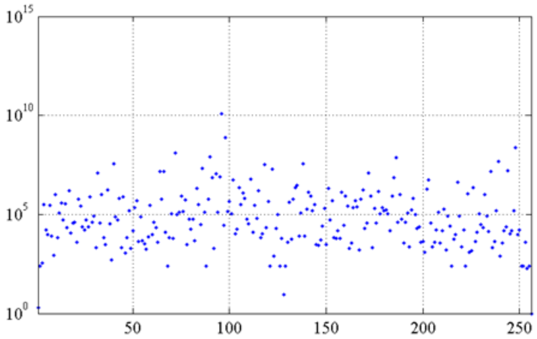
Figure 3: Non Linear Performance of Integer Skew Tent

control parameters is not secure and makes the proposed encryption function not immune against different kinds of attacks such as statistical ones.

3.3 Initialization

This section describes the construction algorithms for the dynamic cipher primitives, S-boxes, P-boxes and round keys. These are generated based on a dynamic secret, which depends itself on a secret key and a nonce. All the notations used in the description are listed in Table 1.

The generation of S-boxes is based on a nonlinear transformation and a bit-rotation operation. These operations ensure the necessary cryptographic performance for a safe implementation in any cryptographic algorithm such as the block cipher proposed in this paper.



(a) Periodicity

Figure 4: Variation of the periodicity against $\tau \in \{1, 256\}$

Conversely to the S-box, a degree of non-linearity is not required for the P-box. The same non-linear transformation used to generate the S-boxes is used to generate the P-boxes, but without using the bit permutation operation. It is important to note that these transformations (substitution and permutation) are invertible and ensure the bijectivity property according to [20]. The main goal of employing different dynamic S-boxes and P-boxes is to design a dynamic cipher structure with a high security level compared to static block ciphers.

The simplicity of the proposed scheme is due to the use of integer transformations and avoiding any floating-point operation. On the other hand, the low computational complexity is proven due to the minimum number of required iterations.

3.4 Dynamic Generation of Sub-keys

In contrast to most encryption algorithms that use static S-box and P-box, the ones used in our algorithm are variable and generated in a lightweight manner, and ensure the desired performance of cryptographic primitives.
Our proposal falls within the *secret shared key system*, where the two ends of the communication system share the same secret key, called *Master Secret Key (K)*, which is used to generate a set of *Dynamic keys (DKs)* that are then used to encrypt the transmitted/stored data. the key K is mixed with a nonce N_o to produce an output X, which is then hashed using SHA-512 to produce the dynamic key DK as shown in Figure 5.

Note that the nonce N_o can be produced at the sender and receiver in a synchronized manner using any secure Deterministic pseudo Random Bit Generator (DRBG) [8]. The seed of the selected DRBG can be constructed by hashing the secret key with any public unique parameter. The produced pseudo-random sequence can be divided to form a set of dynamic nonce values. In concept,

Table 1: Notations

K	Secret key
N_o	Nonce
DK	A dynamic key
Ka	the dynamic addition round sub-key
Ks	Substitution Sub-key
Kp	Permutation Sub-key
RK	A set of r rounds keys $\{RK_1,\ RK_2,\ \ldots,\ RK_r\}$
ID_A	Identification of the transmitter
ID_B	Identification of the legal receiver
Qs	Length of the substitution table (2^8=256 for the byte level)
D	a plain data
r	Number of cipher rounds
rp	Number of transient iterations necessary (threshold) to start building r P-boxes
rs	Number of transient iterations necessary to start building r P-boxes
i	Index of round number
j	Index of block number
t	Index of elements in a block
P	The produced permutation vector
P^{-1}	The corresponding inverse permutation vector
S	The substitution table
S^{-1}	The corresponding inverse substitution vectors
$LSB(X,\ s,\ f)$	Returns the least significant bit of X starting from the index s and ending at the index f
MSB(X, s, f)	Returns the most significant bit of X starting from the index s and ending at the index f
PV	Primary vector
len	Length of a plain data (in byte)
$reshape(X, 1, len)$	Returns a vector with length len of matrix X, whose elements are taken column-wise from X.
Tb	Length of plain-block (in bits)
$Padding(X, Tb)$	The padded bits Tb for a matrix X
$\lceil x \rceil$	Denotes the ceil integer part of x
nb	Number of blocks of data after the padding process.

the produced sequence of each of these DRBGs ensures high periodicity (internal update after each maximum ”$requested\_number\_of\_bits$”) and randomness level. Therefore, each nonce can be used only once, and it should be different

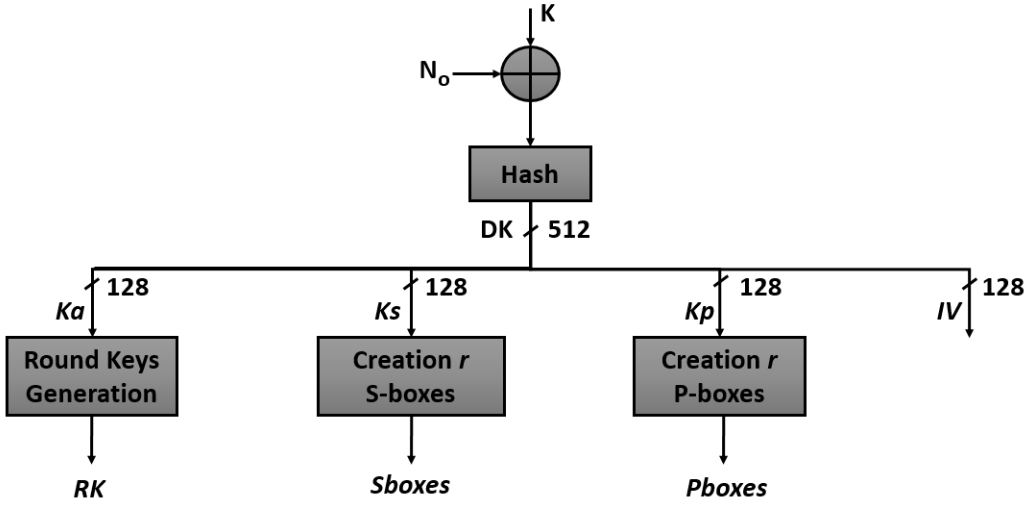


Figure 5: The proposed dynamic key and cipher primitives generation steps.

for every input image. Another possible technique to have a common nonce is to generate it at the sender side and to transmit it in an encrypted form to the legal receiver(s) by using the secret key or the receiver public key.
The master secret key space varies between 128 and 512 bits, while the size of nonce is fixed at 512 bits. Moreover, the size of the dynamic key is also fixed at 512 bits to guard against brute force attacks.

Once DK is generated, it is divided into three sub-keys (Ka, Ks and kp) that contribute to the generation of the three main cipher primitives. The round keys make use of Ka, the substitution tables (S-boxes) are generated as a function of Ks, and the permutation tables (P-boxes) are produced as a function of Kp. Algorithm 1 describes the main steps of the initialization phase: generation of the dynamic key (DK), sub-keys (Ka, Ks, Kp) as well as IV, which will be used for the chaining operation.

The proposed block cipher algorithm uses a variable cipher structure since all cipher primitives are changed after each new input message. For each input plain-block, the round function is iterated for r rounds, depending on the block size, and the value of r is related to the avalanche effect property (see Figure 13), which is variable and depends itself on the block size.

Finally, let us indicate that one of the new innovation idea of the proposed solution is that IV becomes dynamic and secret as the secret key. This leads to increase the security level of the proposed cipher scheme and to make the cryptanalysis more difficult.

Algorithm 1 Initialization step: construction of dynamic r round keys, r S-boxes and r P-boxes

```
procedure INITIALIZATION($K$, $N_o$, $r$, $rs$, $rp$)
    **Input**: $K$ and $IV$
    **Parameters**: $r$,$rs$, and $rp$
▷ Generation of the dynamic key $DK$
    $DK \leftarrow$ **Dynamic_key_derivation**($K$, $IV$ $r$)
    $Ka \leftarrow DK(1:16)$   ▷ Get the first 16 Most Significant Bytes (MSBs).
    $Ks \leftarrow DK(17:32)$   ▷ Get the second 16 MSBs.
    $Kp \leftarrow DK(33:48)$   ▷ Get the third 16 MSBs.
    $IV \leftarrow DK(49:64)$   ▷ It is needed if chaining mode is used.
    $RK = \{RK_1, \ldots, RK_r\} \leftarrow$ **Round_Keys_Generation** ($Ka$, $r$)   ▷ Generation of $r$ round keys
    $Sboxes \leftarrow$ **Creation_r_Sboxes**($Ks$, $rs$, $r$, $Qs$)   ▷ Generation of $r$ substitution tables
    $Pboxes \leftarrow$ **Creation_r_Pboxes**($Kp$ $rp$, , $r$, $Tb$)
    return $\{RK, Pbox, Sbox\}$   ▷ Generation of the $r$ permutation tables
    **Output:** $r$ rounds addition keys, $r$ dynamic key dependent S-boxes and P-boxes.
end procedure
```

3.4.1 Generation of Round keys

The set of round keys, RK, is generated as follows: the current dynamic subkey Ka (16 bytes) is divided into 2 equal parts, each with a size of 8 bytes. Then, each part is converted to an unsigned integer of 64 bits; these parts are used as an initial condition and a control parameter for the integer Skew tent, respectively, to generate the key-stream of the r round keys. To do that, the map is iterated $nt = \lceil \frac{r \times Tbyte}{64} \rceil$ times. Each addition round key has a size of $Tbyte$ and it can be obtained after each of the $\lceil \frac{Tbyte}{64} \rceil$ iterations.

Note that this chaotic map is selected for its simplicity of implementation in hardware and software. Next, the output of the integer skew tent map is converted from integer precision ($N = 64$) back to 8 bytes. As such, each sample is represented using 8 bytes and each group of 2 bytes is used to form 4 sub-samples$\{o_1, o_2, o_3, o_4\}$ that are treated separately.

These variables are mixed together using a non-invertible binary matrix expressed by:

$$
\begin{aligned}
\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \oplus \begin{bmatrix} o_1 \\ o_2 \\ o_3 \\ o_4 \end{bmatrix} \\
\Rightarrow q_1 &= o_1 \oplus o_2 \oplus o_3 \oplus o_4 \\
q_2 &= o_2 \oplus o_3 \oplus o_4 \\
q_3 &= o_2 \\
q_4 &= o_1 \oplus o_2
\end{aligned}
\tag{4}
$$

Then, each q_i is rotated for 3 times to increase the non-linearity degree of the produced round keys, where $i = (1,\ 2,\ \ldots,\ r) \times Tbyte$. Note that the integer skew tent is used with a dynamic sub-key for each new input message or a new session according to the selected configuration.

3.4.2 Generation of S-boxes

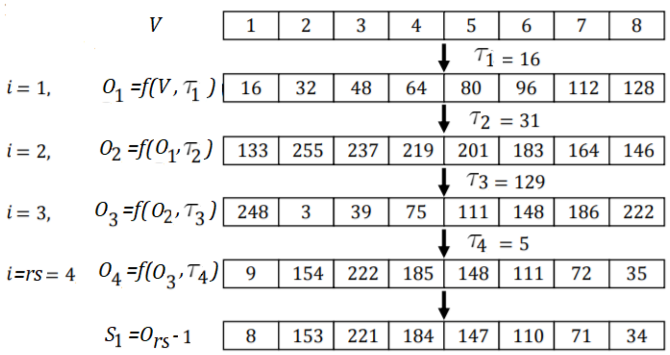
The confusion property requires a substitution operation using a substitution table, $S-box$. The substitution process is a very important operation in cryptographic algorithms, since it provides non-linearity and ensures resistance against specific attacks like differential and linear ones.

The substitution operation is applied at the byte level, and the generation of the S-Box starts with an initial vector V of length 2^8 with values ranging from 1 to 256. Then, we iterate the integer skew tent on this vector for rs times to generate the first useful S-Box. The first $rs-1$ iterations represent the transient phase to ensure the required cryptographic performance whereby the different performance criteria(LPF, DPF, SAC, and BIC [23]) are considered stable and close to the desired one; this is described in section 4.3 and shown in Figure 11, where we show that the minimum value for rs is 4 for the proposed technique. Note that, the output of the chaotic map after each iteration is decremented by 1 to have a value ranging from 0 to 255. Next, a Bitwise right shift (>>) is operated 3 times on each element of the vector; this rotation increases again the non-linearity degree of the whole process [24].

Figure 6 shows a numerical example of S-Box generation. Since rs=4, we perform 4 iterations to get the first useful S-Box, and the Figure shows the first 8 values of the initial vector, and the outputs of the first 4 iterations. The values of the fourth iteration, decremented each by 1, constitute the values of the first S-Box to be used in the first round. Afterwards, we perform a new iteration for every new round to generate the corresponding S-Box. For example, if we need to perform r rounds, we iterate the map first 3 times $(rs-1)$, and then, we iterate the map for an additional r times for a total of $(rs+r-1)$ times.

When performing r rounds, and while iterating the map for $i = 1, 2, \ldots,\ rs+r-1$, the output vector, after each iteration, becomes the input vector to the next one. More details about the proposed generation technique of the S-boxes are listed in Algorithm 2. In addition, The cryptographic performance of the proposed substitution technique is quantified for each iteration.

A numerical example of a full dynamic S-box and its corresponding inverse are displayed in Figure 7.



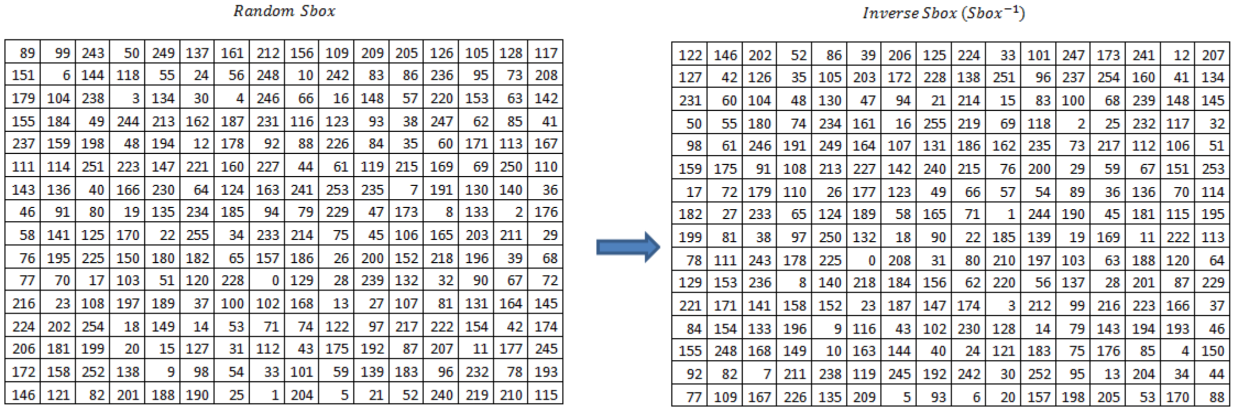
		1	2	3	4	5	6	7	8
	V	1	2	3	4	5	6	7	8
	$\tau_1 = 16$								
$i = 1,$	$O_1 = f(V, \tau_1)$	16	32	48	64	80	96	112	128
	$\tau_2 = 31$								
$i = 2,$	$O_2 = f(O_1, \tau_2)$	133	255	237	219	201	183	164	146
	$\tau_3 = 129$								
$i = 3,$	$O_3 = f(O_2, \tau_3)$	248	3	39	75	111	148	186	222
	$\tau_4 = 5$								
$i = rs = 4$	$O_4 = f(O_3, \tau_4)$	9	154	222	185	148	111	72	35
	$S_1 = O_{rs} - 1$	8	153	221	184	147	110	71	34

Figure 6: A numerical example of how to construct the first substitution table (after rs iterations) for only the first 8 elements of V with τ={16, 31, 129, 5} and Qs =256.

Algorithm 2 Construction of dynamic r S-boxes.

```
procedure CREATION_r_SBOXES($Ks$, $rs$, $r$, $Qs$)
    Input: $Ks$
    Parameters: $r$,$rs$, and $Qs$
    Output: $r$ dynamic key dependent S-boxes.

    for $t \leftarrow 1$ to $Qs$ do
        $V_t \leftarrow t$
    end for
    ▷ Iterate the Integer Non-Linear Finite Skew Tent Map (INLFSTM) for $rs - 1$ iterations
    for $i \leftarrow 1$ to $rs - 1$ do
        $V \leftarrow INLFSTM(V, Ks_i)$
    end for
    ▷ Continue to iterate the Integer Non-Linear Finite Skew Tent Map (INLFSTM) for $r$ iterations, and a substitution table is produced after each iteration.
    for $i \leftarrow rs$ to $r + rs - 1$ do
        $V \leftarrow INLFSTM(V, Ks_i)$
        $Sbox_i \leftarrow (V - 1) >> 3$
    end for
    return Sboxes $\{Sbox_1, Sbox_2, \ldots, Sbox_r\}$
end procedure
```



Random Sbox

89	99	243	50	249	137	161	212	156	109	209	205	126	105	128	117
151	6	144	118	55	24	56	248	10	242	83	86	236	95	73	208
179	104	238	3	134	30	4	246	66	16	148	57	220	153	63	142
155	184	49	244	213	162	187	231	116	123	93	38	247	62	85	41
237	159	198	48	194	12	178	92	88	226	84	35	60	171	113	167
111	114	251	223	147	221	160	227	44	61	119	215	169	69	250	110
143	136	40	166	230	64	124	163	241	253	235	7	191	130	140	36
46	91	80	19	135	234	185	94	79	229	47	173	8	133	2	176
58	141	125	170	22	255	34	233	214	75	45	106	165	203	211	29
76	195	225	150	180	182	65	157	186	26	200	152	218	196	39	68
77	70	17	103	51	120	228	0	129	28	239	132	32	90	67	72
216	23	108	197	189	37	100	102	168	13	27	107	81	131	164	145
224	202	254	18	149	14	53	71	74	122	97	217	222	154	42	174
206	181	199	20	15	127	31	112	43	175	192	87	207	11	177	245
172	158	252	138	9	98	54	33	101	59	139	183	96	232	78	193
146	121	82	201	188	190	25	1	204	5	21	52	240	219	210	115

Inverse Sbox ($Sbox^{-1}$)

122	146	202	52	86	39	206	125	224	33	101	247	173	241	12	207
127	42	126	35	105	203	172	228	138	251	96	237	254	160	41	134
231	60	104	48	130	47	94	21	214	15	83	100	68	239	148	145
50	55	180	74	234	161	16	255	219	69	118	2	25	232	117	32
98	61	246	191	249	164	107	131	186	162	235	73	217	112	106	51
159	175	91	108	213	227	142	240	215	76	200	29	59	67	151	253
17	72	179	110	26	177	123	49	66	57	54	89	36	136	70	114
182	27	233	65	124	189	58	165	71	1	244	190	45	181	115	195
199	81	38	97	250	132	18	90	22	185	139	19	169	11	222	113
78	111	243	178	225	0	208	31	80	210	197	103	63	188	120	64
129	153	236	8	140	218	184	156	62	220	56	137	28	201	87	229
221	171	141	158	152	23	187	147	174	3	212	99	216	223	166	37
84	154	133	196	9	116	43	102	230	128	14	79	143	194	193	46
155	248	168	149	10	163	144	40	24	121	183	75	176	85	4	150
92	82	7	211	238	119	245	192	242	30	252	95	13	204	34	44
77	109	167	226	135	209	5	93	6	20	157	198	205	53	170	88

Figure 7: A specific example of creation of dynamic S-box with numerical values and its corresponding inverse one (see Eq. 8) by using the proposed technique.

3.4.3 Generation of P-boxes

In the proposed scheme, the permutation process is performed at the bit level. A specific permutation table $P-box$ is used for each encryption round, while its inverse $(P-box)^{-1}$ is used for decryption. Initially, a Primary Vector of Tb elements is initialized as follows: $PV_t = t$, with $t = 1, \ldots, Tb$. Then, the integer skew tent map is applied on this vector several times; first, we iterate for rp times to generate the first useful P-Box. The first $(rp-1)$ iterations represent the transient phase to ensure the required cryptographic performance; this is described in Section 4.2 and shown in Figure 10, where we show that the minimum value for rp is 4 for the proposed technique. Note that for each iteration w, the w^{th} byte of Kp is used as a control parameter and the output vector of each iteration becomes the input vector to the next one. This will randomize the PV vector to become the first P-box after rp iterations.
Next, we iterate for $(r-1)$ times for a total of $(rp+r-1)$ to generate the required r P-Boxes. The implementation details are listed in Algorithm 3.

Finally, and since the integer skew tent map is invertible (bijective) [20], then each produced $P-box$ is invertible and its inverse $(P-box)^{-1}$ always exists and can be produced by using Eq. 11.

Figure 8 shows a numerical example of P-Box generation. Since $rp = 4$, we perform 4 iterations to get the first useful P-Box, and the Figure shows the values of the initial vector for a block size Tb of 8, and the outputs of the first 4 iterations. The vector values of the fourth iteration constitute the values of the first P-Box to be used in the first round. Afterwards, we perform a new iteration for every new round to generate the corresponding P-Box. For example, if we need to perform r rounds, we iterate the map first 3 times $(rp-1)$, and then, we iterate the map for an additional r times for a total of $(rp+r-1)$ times.

Algorithm 3 Construction of dynamic r P-boxes.

```
procedure CREATION_r_PBOXES($Kp$, $rp$, $r$, $Tb$)
    Input: $Kp$
    Parameters: $r$,$rp$, and $Tb$
    Output: $r$ dynamic key dependent P-boxes.

    for $t \leftarrow 1$ to $Tb$ do
        $PV_t \leftarrow t$
    end for
    ▷ Iterate the Integer Non-Linear Finite Skew Tent Map (INLFSTM) for rp-1 iterations.
    for $i \leftarrow 1$ to $rp - 1$ do
        $PV \leftarrow INLFSTM(PV, Kp_i)$
    end for
    ▷ Continue to iterate the Integer Non-Linear Finite Skew Tent Map (INLFSTM) for r iterations, and a permutation table is produced after each iteration.
    for $i \leftarrow rp$ to $r + rp - 1$ do
        $PV \leftarrow INLFSTM(PV, Kp_i)$
        $Pbox_i \leftarrow PV$
    end for
    return Pboxes $\{Pbox_1,\ Pbox_2,\ \ldots,\ Pbox_r\}$
end procedure
```

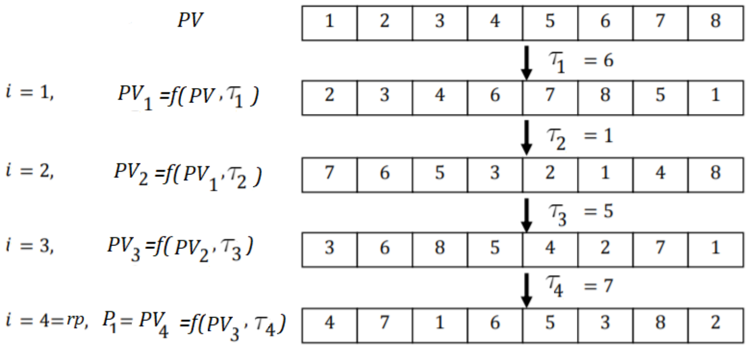


Figure 8: A numerical example of how to construct the first permutation table (after rp iterations) with length Tb equals to 8. PV is iterated with Kp = τ={6, 1, 5, 7}.

3.5 Proposed Block Cipher: CBC mode

In this section, the proposed cipher/decipher algorithm is described. The plain data D is divided into nb blocks, each of length Tb in bits (32, 64, 128, 256, and so on): $\{B_1, B_2, \ldots, B_{nb}\}$. The proposed encryption algorithm deals with data of *len*-byte length. If *len* is not a multiple of $Tbyte = \frac{Tb}{8}$, a padding operation is required for the last block of data when considering ECB, CBC, or CFB operation modes, while this operation is not needed for OFB and CTR.

3.5.1 Encryption Algorithm

Based on the dynamic key DK, the three sub-keys are produced and used for the generation of r round keys, S-boxes and P-boxes (or their inverse counterparts at the receiver side). The proposed encryption algorithm performs r rounds of SP-network on each block. Note that the block length Tb depends on the application and its corresponding memory constraints; a small size of Tb is preferable for low memory capacity, which is less than or equal to 64 bits. Note that our algorithm can run with any value of $Tb = 2^i$, where $i = 3,\ 4,\ 5,\ 6,\ \ldots$.

Algorithm 4 Proposed SPN Algorithm

```
procedure SPN_Ciphering($B$, $Ka$, $Pbox$, $Sbox$, $r$)
    Input:$B$, $Ka$, $Pbox$, $Sbox$
    Parameters: $r$
    Output: Encrypted block $C$
    $C \leftarrow B$
    for $i \leftarrow 1$ to $r$ do
        $X \leftarrow Addition\_Round\_key(C,\ Ka_i,\ i)$
        $Y \leftarrow Byte\_Substitution\_layer(X,\ Sbox_i)$
        $Z \leftarrow Bits\_Permutation\_layer(Y,\ Pbox_i)$
        $C \leftarrow Z$
    end for
    return $C$
end procedure
```

Algorithm 4 describes the proposed SPN block cipher, which is implemented in the following modes: Cipher Block Chaining (CBC), Output Feedback (OFB), and Counter (CTR) mode. The decryption scheme is similar to the encryption one, but the decryption process reverses both the order of rounds and the round layers, and it employs the inverse of P-boxes and S-boxes. The basic steps of the proposed encryption scheme are presented in Figure 2; they consist of applying a round function for r rounds to achieve a high level of randomness and to ensure the avalanche effect. This function consists of three main layers: addition, byte substitution operation based on an S-box, and the diffusion operation is based on bit permutation – in order to achieve the confusion and diffusion proprieties.

- **Addition layer**

This is the first operation of the encryption round function (and so the last one for the decryption round function). The addition operation is carried out on bytes by using the logical exclusive-OR operation:

$$X^j = B^j \oplus RK_i, \tag{5}$$

where $i = 1,\ 2,\ \ldots,\ r$ and j=1, $2,\ \ldots,\ nb$, B^j is the j^{th} plain block, and RK_i is the i^{th} addition round key.

- **Substitution layer: S-box**

Each resultant mixed block X^j follows a substitution process that employs the i^{th} S-box (S_i) to perform the corresponding substituted block Y^j

$$Y^j = Substitution(X^j, S_i) \tag{6}$$

The inverse operation of substitution layer that is used in the decryption process is given by

$$X^j = Substitution(Y^j, S_i^{-1}) \tag{7}$$

The S-box inverse, S^{-1}, can be obtained by the operation:

$$S^{-1}[S(t)] = t, \text{ where } t \text{ and } S(t) \in \{0, 255\}. \tag{8}$$

- **Permutation layer, r P-boxes:**

Each substituted mixed block, Y^j, follows a bit permutation process that employs the i^{th} P-box, P_i, which leads to the output Z^j, and the latter becomes the input block for the $(i+1)^{th}$ round:

$$Z^j = \pi(Y^j, P_i) \tag{9}$$

The inverse operation of the permutation layer is given by:

$$Y^j = \pi(Z^j, P_i^{-1}) \tag{10}$$

Similarly, the inverse P-box, P^{-1}, is also bijective and can be obtained by:

$$P^{-1}[P(t)] = t, \text{ where } t \text{ and } P(t) \in \{1, ..., Tb\}. \tag{11}$$

3.5.2 Decryption Algorithm

Similarly to the encryption scheme, decryption consists of applying the inverse round function, thus applying the layers in reverse order. Algorithm 5 summarizes the decryption scheme, which reverses the encryption order and replaces the permutation and substitution boxes by their inverse counterparts, $(S - box)^{-1}$ and $(P - box)^{-1}$.

Algorithm 5 The corresponding decipher SPN Algorithm

```
procedure SPN_DECIPHERING($C$, $RK$, $Pbox^{-1}$, $Sbox^{-1}$, $r$)
    Input:$C$, $RK^{-1}$, $Pbox^{-1}$, $Sbox^{-1}$
    Parameters: $r$
    $B = C$
    for $i \leftarrow r$ down to 1 do
        $B \leftarrow Inverse\_Bits\_Permutation\_layer(B, Pbox_i^{-1})$
        $B \leftarrow Inverse\_Byte\_Substitution\_layer(B, Sbox_i^{-1})$
        $B \leftarrow Inverse\_Addition\_Round\_key(B, RK_i, i)$
    end for
    return Decrypted block $B$.
end procedure
```

4 Cryptographic Performance of the Proposed Cipher Layers

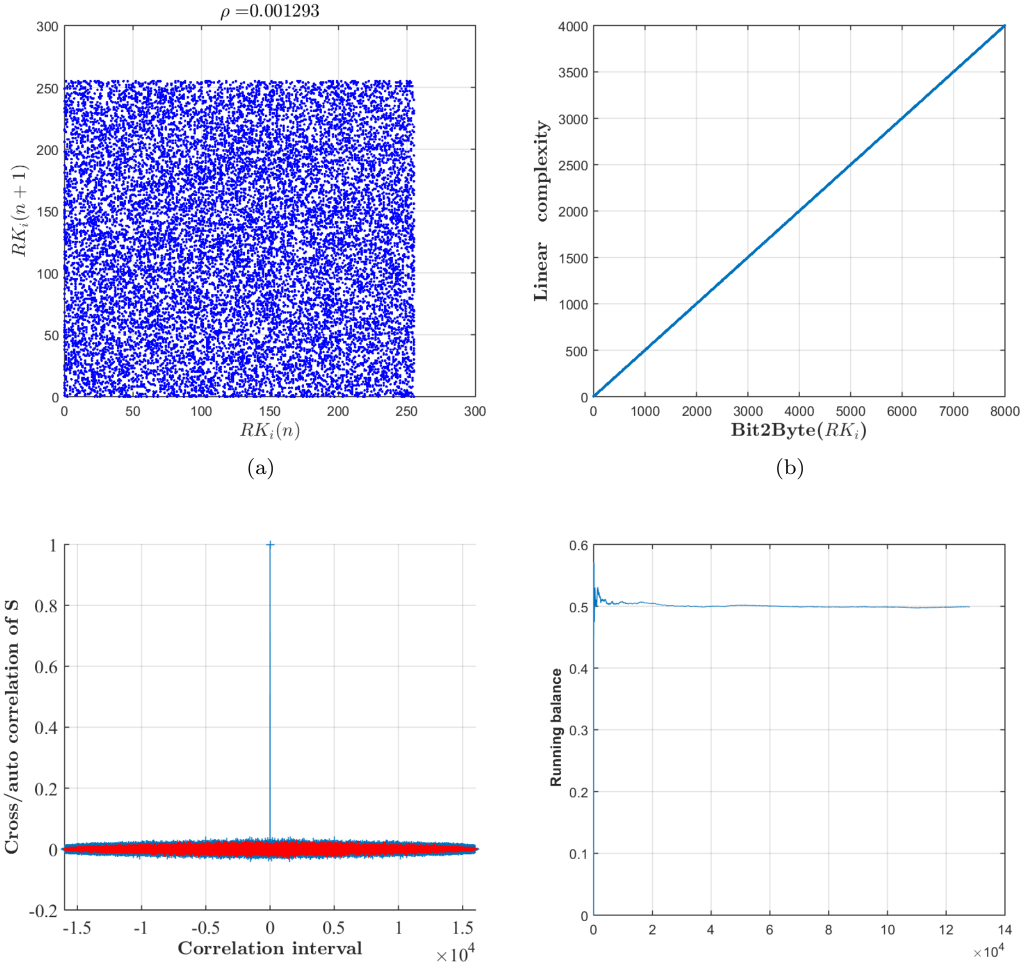
In this section, the cryptographic properties of the proposed round key expansion, substitution, and diffusion primitives are discussed and analyzed in details, to prove their robustness and to validate them as appropriate primitives for symmetric cryptography.

4.1 Performance of key derivation function

A good key derivation function should produce pseudo-random bit sequences with good randomness degree. This latter can be checked by the following properties: uniform distribution, large linear complexity (approximately equal to half of the sequence period), long period, delta-like auto-correlation, nearly zero cross-correlation, and it should finally be able to pass empirical statistical tests. As can be seen in Figure 9: a) the produced sequence has a random trajectory; b) the linear complexity of the bit sequence is as required; c) the auto-correlation looks like a delta-function, and the cross-correlation curves are always close to zero, although there is only a slight difference between the parameters; and d) there is an equal mean of bits. In order to quantify the randomness degree of the proposed key derivation function, the NIST test has been applied as in [26] and the proportion value from these tests show that the proposed Key Derivation Function (KDF) passes all the tests, and the produced binary sequence achieves all necessary conditions and has a good randomness degree and as such, it can be safely used to secure communication systems.

4.2 Cryptographic performance of the dynamic permutation layer

The performance of the proposed dynamic permutation algorithm should be quantified in order to demonstrate its safe implementation. Two criteria are



(a) (b)

(c) (d)

Figure 9: Cryptographic properties of sequence obtained using our proposed generator, with N=32 and sequence length 8,000 bits (1,000 bytes). (a) Mapping, (b) linear complexity, (c) auto/cross-correlation function, (d) running balance and (e) NIST tests.

employed here; the first one is the coefficient of correlation (described in [25]) to check the recurrence of permuted vectors $((P_{(t)},\ P_{(t+1)}),\ t = 1,\ 2,\ \ldots,\ Tb-1)$, and to quantify the minimum iteration-number of permutations, rp. These tests were applied for $nk = 2^{15}$ random dynamic keys and for different sizes of the block (Tb=2^q, q=4, $\ldots$, 10). Figure 10 shows the average coefficient of correlation between the recurrence of permuted index versus rp for various Tb. The second criterion, the probability of the number of unique P-boxes versus

rp for different Tb, is shown in Figure 10.

Based on these results, it is clear that $rp \geq 4$ is the iteration threshold, since the coefficient correlation becomes close to zero (ideal value) and the probability of unique P-boxes is close to the maximum possible value ($0.8 \times Tb!$),where ! represents the factorial function. Consequently, rp is set to 4 for the different sizes of Tb, and the choice of this value is justified.

4.3 Cryptographic performance of the dynamic substitution layer

According to information theory, a strong $n \times n$ substitution layer must have the following properties: bijectivity, both a low linear probability function (LPF) and differential probability function (DPF), strict avalanche criterion (SAC), output bits independence criterion (BIC), and equiprobable input/output XOR distribution [9, 21, 30]. The proposed nonlinear transformation is applied for multi-iterations $i_{rs} = 1, 2, \ldots, rs$ with rs being the number of iterations of the substitution layer. Starting with initial vector V, where $V_j = j$ and $j = 0, 1, \ldots,\ 255$, and using 1,000 random dynamic keys of substitution layer ($s_i,\ t_i$), the output vector after each iteration becomes the input vector for the next iteration. And, at each iteration, a random control parameter (τ_i) is generated, while the aforementioned cryptographic properties of each key are calculated after each iteration.

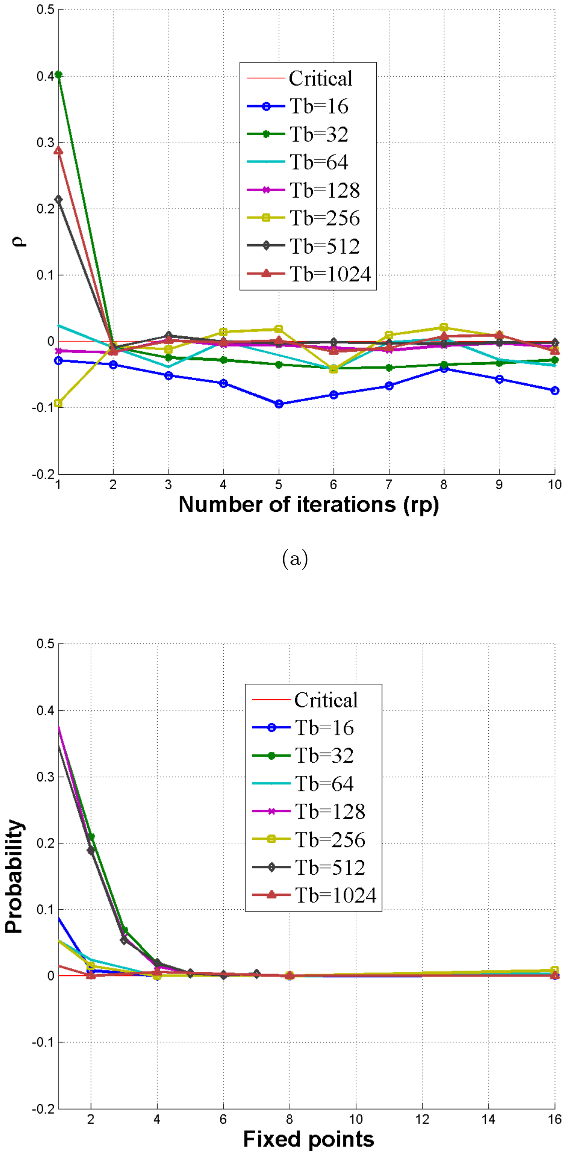
The results for LPF, DPF, SAC, and BIC show that the optimal number of iterations to construct good substitution layers with acceptable cryptographic performance is 4. Therefore, the substitution layer is constructed by applying the proposed transformation for more than four iterations, where each iteration uses random control parameters. In summary, the proposed confusion layer possesses suitable properties to be used in a block cipher with a chaining mode of operation.

5 Experimental results

In this section, an experimental evaluation of the proposed block cipher algorithm is performed, which encompasses the measurement of uniformity, randomness, and key sensitivity.

5.1 Statistical Analysis

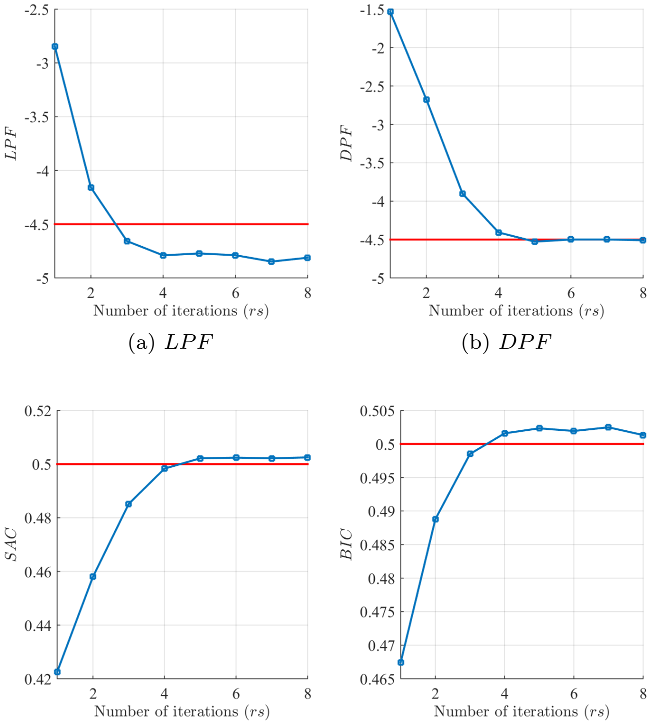
To be able to resist statistical attacks, the cipher should exhibit various random properties [32]. To this end, we conducted an analysis involving the following statistical tests: (a) Histogram analysis, (b) Entropy analysis, and (c) Correlation between plain and encrypted data.



(a)

(b)

Figure 10: Variation of the average of ρ of the recurrence of producing P-boxes versus rp for 1,000 random dynamic keys (a) and its corresponding probability of fixed points for $rp = 3$.



(a) LPF

(b) DPF

(c) DPF

(d) DPF

Figure 11: Average LPF (a), DPF (b), SAC (c) and BIC (d) versus the number of iterations.

5.1.1 Uniformity Analysis

The histogram of the data to encrypt, converted into an image, is used to show how pixels are distributed by plotting the number of pixels at each intensity level. An encryption algorithm can resist statistical attacks when the plain data does not have any similarity to the corresponding cipher data. To verify this characteristic, we analyzed the histograms of eight original standard images: Lena, Pepper, Baboon, Cameraman, Tiffany, Lake, F16, and Elaine, each of size 256×256, as well as the histogram of their corresponding cipher data, as shown in Figure 14.

We can see that the histograms of the original data are quite different from those of the corresponding cipher data. Indeed, the cipher data follows a uniform distribution, which is significantly different when compared to the distribution of the plain data. Hence, the histograms of the cipher data do not reveal any useful information that can be used to launch a statistical attack on the proposed encryption approach. Moreover, in order to compute the uniformity level of the encrypted data, the entropy test is applied as described next.

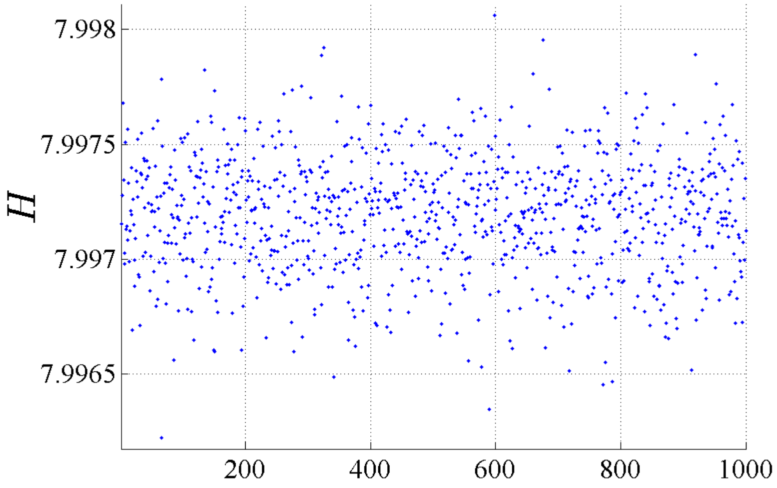


Figure 12: Entropy analysis for the encrypted Lena data with 1,000 random dynamic keys

5.1.2 Information Entropy Analysis

The information entropy of a source message m is a metric that measures the level of uncertainty in a random variable [33], and is defined using the equation:

$$H(m) = \sum_{i=0}^{2^M-1} p(m_i) \log_2 \frac{1}{p(m_i)} \tag{12}$$

where $p(m_i)$ represents the probability of occurrence of the symbol m_i and 2^M is the total number of states of the information source. Note that this entropy is expressed in bits. The entropy of a truly random source is equal to 8.

Figure 12 illustrates the variation of entropy of the encrypted Lena data through 1,000 iterations. As we can see, the mean value of entropy is 7.997 which is very close to the theoretical value of 8. This value can be compared with the entropy of recent schemes mentioned in [33]. The Baptista's algorithm, for instance, reports $H = 7.926$, and the Wong's one has $H = 7.969$. Xiang's algorithm, for its part, reports $H = 7.995$, while the Sun's method presents an entropy equals to $H = 7.996$. This comparison allows us to conclude that the proposed encryption scheme achieves a better experimental security than the existing algorithms, and a similar security compared to Kalka ($H = 0.997$): the proposed cipher is sufficiently secure against entropy-related attacks.

5.2 Sensitivity Test

There are several attacks based on studying the relationship between two cipher blocks resulting from a slight change (usually one-bit difference) of an original block or secret key. The sensitivity test indicates how much a slight change in the plain-block or in the key will affect the resulted encrypted/decrypted block. The higher the change, the better the sensitivity of the encryption algorithm. Such types of sensitivity are analyzed below.

5.2.1 Plain-text Sensitivity: choice of the rounds' number value r

To demonstrate the level of sensitivity of the proposed cipher against a little change on the plain-text, we consider the following scenario: Two plain-text blocks P_1 and P_2 that have only one bit difference (their Least Significant Bit LSB) are encrypted separately, to produce two cipher-texts C_1 and C_2. Then, the Hamming distance (in bits) between these two cipher-texts is calculated using Equation 13:

$$\begin{aligned} PS =& \frac{\sum_{k=1}^{T} C_1 \oplus C_2}{Tb} \times 100\% \\ =& \frac{\sum_{k=1}^{T} E_{DK_v}(P_1) \oplus E_{DK_v}(P_2)}{Tb} \times 100\% \end{aligned} \tag{13}$$

where Tb is the length in bits of the plain-text and cipher-text blocks.

To confirm the result, this process is iterated on 100 random plain-texts. The obtained mean value is close to 50%, which means that with a little change in the plain-text, more than 50 % of the corresponding cipher-text is changed. As such, we can reasonably assume that the cipher successfully satisfies the avalanche effect. Figure 13 (a) shows that the majority of plain-text sensitivity PS values are close to the optimal value of 50%. Therefore, the proposed approach has enough sensitivity against any change on the plain-text. Based on these results, the number required to attain a good avalanche effect is $r_{min} = 4$ for $Tb = 64$ and 5 when $Tb = 128$.

Table 2: Minimum required number of rounds (r_{min}) to reach the avalanche effect versus the block size (Tb)

Tb	16	32	64	128	256	512	1024
r_{min}	3	4	4	5	5	6	6

5.2.2 Key Sensitivity

To study the key sensitivity, two dynamic keys are used: DK_v and DK'_v, which differ by only one bit (the LSB). The key sensitivity analysis KS is performed following the same procedure of the plain-text sensitivity. The Hamming distance of the corresponding encrypted cipher-texts C and C' is computed and the obtained results are depicted in Figure 13-(b).

Again, the majority of values are close to the optimal value (50 %), which indicates that the proposed encryption approach has enough strength against any little change in the dynamic key and achieves the avalanche criterion. Furthermore, the results are close to those reported in [4, 17, 29, 22] which are equal to 49.98, 49.97, 49.99 and 49.999, respectively.

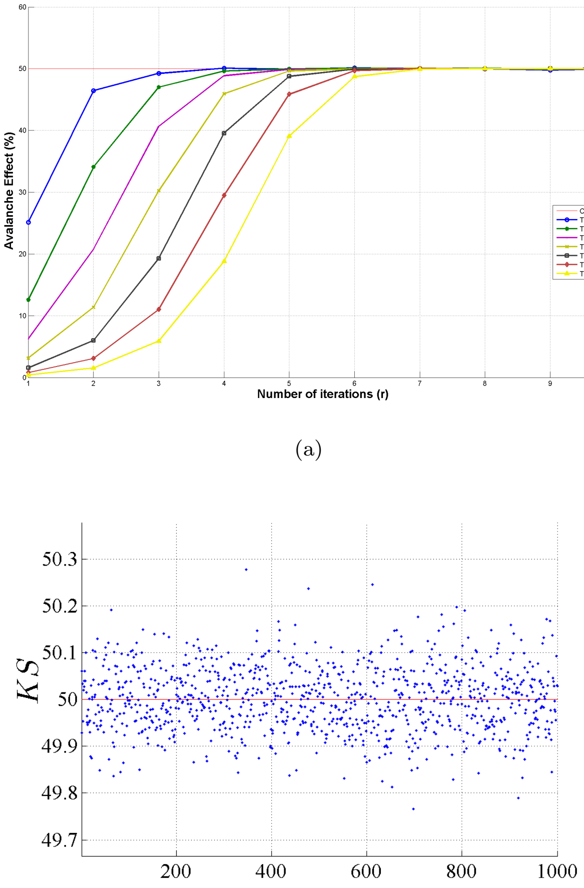
5.3 Key space

To resist brute force attacks, the cipher scheme should have a large key space with a key size ($\geq$ 128 bits). In our case, the master key space can be 2^{128}, 2^{256}, or 2^{512}, which is sufficiently large to make the brute-force attack unfeasible in practice. Additionally, the key space of the dynamic key is 2^{512}, which can also be considered large enough to make the brute-force attack unfeasible.

Additionally, a large master secret key and also a large dynamic key are used in our proposition. As the difficulty of cipher-text-only attack is equal to one of the brute force attacks, it becomes impossible for a cipher-text-only attack to retrieve useful information from the cipher-data. Therefore, the method resists cipher-text attacks.

5.4 On the usefulness of chaos properties

In the previous sections, we have discussed and verified experimentally various properties related to the security of the encryption algorithm. Such properties include the uniformity analysis, the information entropy, and both plain-text and key sensitivities. However, the encryption function is not applied alone, as it is embedded within a specific mode of operation. It is important to verify that the avalanche effect is likely to be preserved, as well as the aforementioned sensitivities.



(a)

(b)

Figure 13: Average of (a) plain-text sensitivity (Avalanche effect) and (b) key sensitivity for $1,000$ different dynamic keys.

At this point, we have computationally constructed the graph $\mathcal{G}_g$, as defined in Section 2.2 and with $g = \varepsilon_k \circ F_{f_0}$, in which ε_k is our encryption scheme proposed in Section 3. For that, experiments have been conducted for two kinds of key size. First, we consider all the possible keys of size 2^8, leading each time to a directed graph with 2^8 edges, each having 2^8 outgoing edges. We used Matlab and verified that every time, the associated directed graph is strongly connected. For keys and messages of size 2^{16}, such an approach becomes intractable, and so only some keys have been randomly picked and tested. With more details, using a 8 cores Xeon machine, it takes 4 hours to check this property for only one key. So we tested 100 keys, and every time the obtained graph was verified to be a strongly connected one.

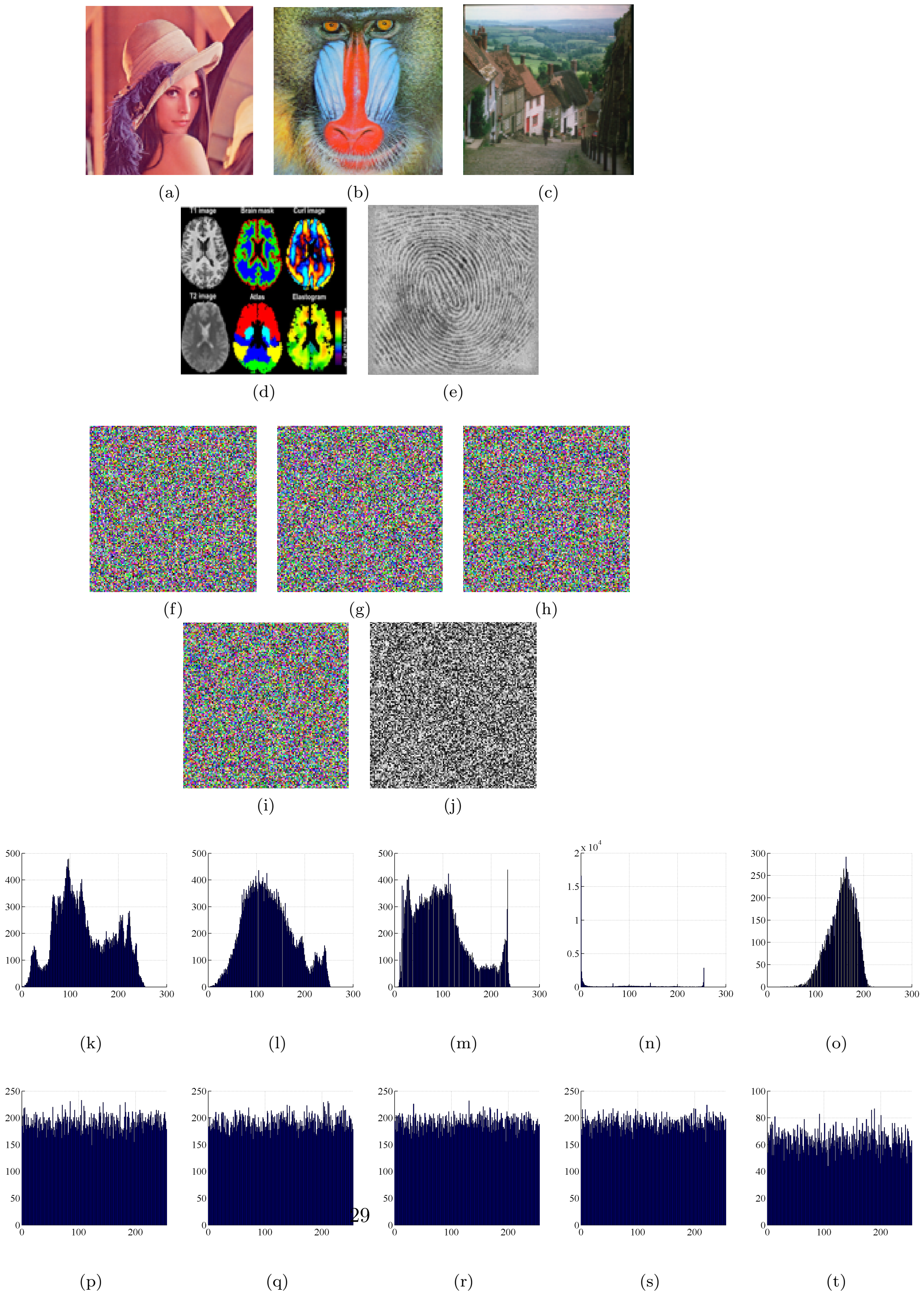


Figure 14: (a)- (e) Original Lena, Baboon, Goldhill, Murphy of brain MRE and finger images, (k)- (o) their corresponding histograms, (f)- (j) the corresponding cipher images, (p)-(t) the Histograms of the corresponding cipher images.

According to Theorem 1, such a property leads to a chaotic behavior of the mode of operation. In particular, properties of sensitivity to the initial conditions, of expansiveness, and of large Lyapunov exponent, guarantee that avalanche effect and both plain-text and key sensitivities will be preserved. Of course, it would be interesting to check this property for bigger size of messages and keys, but it is impossible to check this practically, and theoretical investigations are preferred instead.

6 Cryptanalysis Discussion

In principle, an efficient cipher scheme should resist to most known types of attacks such as statistical, differential, brute-force, and chosen/known plain-text/cipher-text attacks. This section discusses the proposed cipher scheme in the context of these attacks.

6.1 Statistical Attacks

To resist statistical attacks, encrypted messages should meet the randomness requirements. The obtained results of statistical tests presented in Section 5 prove the random nonlinear recurrence, uniformity of encrypted message, and that no correlation exists between the encrypted and original messages. Consequently, no useful information can be detected from transmitted messages (a dynamic key for each input message), which validates the robustness of proposed cipher scheme. Finally, the proposed cipher scheme exhibits a high resistance degree against statistical attacks.

6.2 Linear and Differential Attacks

The proposed scheme is based on a dynamic key-dependent approach. It achieves the avalanche effect, and the encrypted messages have a high nonlinear degree based on the use of r different S-boxes and the use of a dynamic key. Moreover, it is very difficult for an attacker to determine which dynamic key is used for each input message (one-way function). Therefore, the proposed cipher approach is immune against linear attacks.

In a differential attack, the relationship between two encrypted messages is exploited. However, in the proposed scheme, a different dynamic key and consequently different cipher primitives are generated for every message, which makes the relationship between two consecutive encrypted messages highly uncorrelated. This has also been demonstrated in the key sensitivity tests in which two encrypted messages originally derived from the same original message using slightly different keys, are significantly different (by at least 50%). Thus, the proposed technique is robust and secure against linear and differential attacks.

6.3 Resistance Against Chosen/Known Plain-text/Cipher-text Attacks

Note that the chosen/known plain-text/cipher-text attack is a subset of the linear and differential attacks, which have proven to be unfeasible and unsuccessful when using the proposed dynamic scheme. Hence, this validates the security, efficiency and robustness of the proposed scheme against chosen/known plain-text/cipher-text attacks. In addition, the issues related to single data failure and accidental key disclosure are avoided in this case.

6.4 Resistance Against Key-Related Attacks

As illustrated in Figure 13-(b), the proposed technique exhibits a key sensitivity close to the desired one of 50%. Consequently, a high level of resistance against related-key attacks is achieved. This outcome is justified since the proposed scheme has a dynamic structure that changes with every input message, in contrast to existing symmetric cipher schemes, which follow a static structure.

6.5 Weak Keys

The proposed key generation and update cipher primitive technique produces a set of dynamic sub-keys with a high degree of randomness. Moreover, all cipher operations are directly related to a dynamic key and ensure the desirable cryptographic strength. In the worst case, if any weakness exists in any of the dynamic keys, it will not affect the previous or subsequent message. Hence, the proposed approach is highly resistant to weak keys; this complicates the task of attackers since it will be very hard to recover the next dynamic key that is based on a different nonce.

Any weakness in any cryptographic operation is avoided in the proposed scheme since different elements (round key, S-box or P-box) are produced for each round. Moreover, the variation of the dynamic key for each new input message and consequently the cipher primitives, results in different cipher-texts and consequently, guards against any key disclosure accidents.

6.6 Brute-Force and Key-related Attacks

The secret key space is sufficiently large (2^{128}, 2^{196}, 2^{256} or 2^{512}), and the nonce and dynamic key space is 2^{512}, which makes brute-force attacks unfeasible. In addition, the obtained results of the sensitivity tests prove that any bit change in the secret key or the nonce causes a significant difference in the encrypted messages, as seen in Figure 13-(b). This demonstrates the efficiency of the proposed dynamic key scheme against key-related attacks due to the dynamic structure.

Finally, the proposed dynamic key-dependent approach provides better immunity against powerful future attacks since dynamic cipher primitives and IV can be produced for each new input message. Note that the existing attacks target traditional static cipher structures, which is not the case in the proposed approach.

7 Theoretical Considerations: Chaos Properties of OFB Mode

In this section, we refer to the definitions of Section 2.2 to show that the study of the dynamics and the disorder generated by the CBC mode can be extended to other modes of operation. We consider the Output Feedback (OFB [13]) mode, which is defined below.

Let $IV \in \mathbb{B}^{\mathsf{N}}$ be the input vector, $(m_i)_{i\in\mathbb{N}} \in \left(\mathbb{B}^{\mathsf{N}}\right)^{\mathbb{N}}$ the sequence of block messages to encrypt, and $\mathcal{E}_{\mathsf{k}} : \mathbb{B}^{\mathsf{N}} \longrightarrow \mathbb{B}^{\mathsf{N}}$ the encryption function, where k is the encryption key. The sequence $(o_i)_{i\in\mathbb{N}} \in \left(\mathbb{B}^{\mathsf{N}}\right)^{\mathbb{N}}$ of encrypted output block messages is computed as described below.

$$\begin{cases} i_0 = IV \\ o_0 = \mathcal{E}_{\mathsf{k}}(i_0) \\ c_0 = m_0 \oplus o_0 \end{cases}$$

and $\forall n \in \mathbb{N}t$

$$\begin{cases} i_{n+1} = o_n \\ o_{n+1} = \mathcal{E}_{\mathsf{k}}(i_{n+1}) \\ c_{n+1} = m_{n+1} \oplus o_{n+1}. \end{cases}$$

We are then left to rewrite these recurrent sequences as a discrete dynamical system, in order to study its chaotic behavior. Let us consider the maps

$$\begin{array}{rcl} i: & \left(\mathbb{B}^{\mathsf{N}}\right)^{\mathbb{N}} & \longrightarrow \left(\mathbb{B}^{\mathsf{N}}\right)^{2} \\ & (m_0, m_1, m_2, \ldots) & \longmapsto (m_0, m_1) \end{array}$$

that outputs the two first terms of a sequence, and

$$\begin{array}{rcl} \sigma: & \left(\mathbb{B}^{\mathsf{N}}\right)^{\mathbb{N}} & \longrightarrow \left(\mathbb{B}^{\mathsf{N}}\right)^{\mathbb{N}} \\ & (m_0, m_1, m_2, m_3, \ldots) & \longmapsto (m_1, m_2, m_3, \ldots) \end{array}$$

that performs a shift on it. Considering the set

$$\mathcal{X} = \left(\mathbb{B}^{\mathsf{N}}\right)^{\mathbb{N}} \times \mathbb{B}^{\mathsf{N}},$$

then, the OFB mode of operation can be rewritten as follows:

$$\begin{cases} X^0 &= ((m_0, m_1, m_2, \ldots); m_0 \oplus \mathcal{E}_{\mathsf{k}}(IV)) \\ X^{k+1} &= \left(\sigma(X_0^k); i(X_0^k)_1 \oplus \mathcal{E}_{\mathsf{k}}\left(i(X_0^k)_0 \oplus X_1^k\right)\right) \\ &= f_{\mathcal{E},\mathsf{k}}(X^k), \end{cases}$$

in which $\forall k \in \mathbb{N}, X^k \in \mathcal{X}$ is the k-th term of the sequence X, which has two components: $X_0^k \in \left(\mathbb{B}^{\mathsf{N}}\right)^{\mathbb{N}}$ and $X_1^k \in \mathbb{B}^{\mathsf{N}}$ as designed by the Cartesian product defining $\mathcal{X}$. More precisely, X_0^k is the sequence of block messages to encrypt after k shifts, while X_1^k is the encrypted block message o_k to output. The OFB mode of operation is thus the discrete dynamical system $X^{k+1} = f_{\mathcal{E},\mathsf{k}}(X^k)$ defined on $\mathcal{X}$, where

$$\begin{array}{rcl} f_{\mathcal{E},\mathsf{k}}: \quad \mathcal{X} & \longrightarrow & \mathcal{X} \\ (m, E) & \longmapsto & (\sigma(m); i(m)_1 \oplus \mathcal{E}_{\mathsf{k}}\left(i(m)_0 \oplus E\right)). \end{array}$$

There are two steps before we can study the dynamics of the OFB: first, an ad-hoc distance d must be defined on $\mathcal{X}$, and then, the continuity of $f_{\mathcal{E},\mathsf{k}}$ must be stated on the metrical space $(\mathcal{X}, d)$. The former is defined by:

$$d(X,Y) = d_M(X_0, Y_0) + d_E(X_1, Y_1),$$

where

$$\begin{cases} d_M\left((m_0, m_1, m_2, \ldots); (\check{m}_0, \check{m}_1, \check{m}_2, \ldots)\right) = \dfrac{9}{\mathsf{N}} \displaystyle\sum_{k=0}^{\infty} \dfrac{|m_k - \check{m}_k|}{10^{k+1}} \\ d_E\left((e_1, \ldots, e_{\mathsf{N}}); (\check{e}_1, \ldots, \check{e}_{\mathsf{N}})\right) = \displaystyle\sum_{k=1}^{\mathsf{N}} |e_k - \check{e}_k|. \end{cases}$$

Note that d_E is simply the Hamming distance on $\mathbb{B}^{\mathsf{N}}$, while the value $\dfrac{9}{\mathsf{N}}$ in d_M is a simple normalization factor, which has been introduced for the following reason: the k-th digits of $d_M(m, \check{m})$ is 0 if and only if $m_k = \check{m}_k$. Note finally that the integral part of $d(X,Y)$ is $d_E(X_1, Y_1)$, while its fractional part is $d_M(X_0, Y_0)$. The proof that d_M is a distance is immediate, while d_E is known to be a distance (the Hamming one). So d is a distance, since it is defined as the sum of two distances.

Let us now prove that:

Proposition 1 $f_{\mathcal{E},\mathsf{k}}$ *is continuous on the metrical space* $(\mathcal{X}, d)$.

PROOF This property is established by using the sequential characterization of the continuity. Let us consider a sequence $X^n = (m^n, e^n)$ that converges to $X = (m, e)$. If we can establish that $f_{\mathcal{E},\mathsf{k}}(X^n)$ converges to $f_{\mathcal{E},\mathsf{k}}(X)$, then we have proven the continuity of the map. Note that, $\forall n, m^n$ is a sequence of block messages, and so $(m^n)_{n \in \mathbb{N}}$ is a sequence of sequences.

On the one hand, $X^n \longrightarrow X$ implies that $e^n \longrightarrow e$. Due to the Hamming distance, $\exists n_0 \in \mathbb{N}, \forall n \geqslant n_0, e^n = e$. On the other hand, $m^n \longrightarrow m$ so, based on the definition of d_M, $\exists n_1 \in \mathbb{N}, n \geqslant n_1 \Rightarrow m_0^n = m_0$ and $m_1^n = m_1$. As a consequence, $\forall n \geqslant n_1$, $i(X_0^n)_0 = m_0$ $i(X_0^n)_1 = m_1$, which implies that $f(X^n)_1 = m_1 \oplus \mathcal{E}_{\mathsf{k}}\left(m_0 \oplus e\right) = f(X)_1$: the convergence of the second component is guaranteed. Finally,

$$\begin{aligned}
d_M(\sigma(m^n),\sigma(m)) &= \frac{9}{\mathsf{N}}\sum_{k=0}^{\infty}\frac{|m^n_{k+1}-m_{k+1}|}{10^{k+1}} \\
&= \frac{9\times 10}{\mathsf{N}}\sum_{k=1}^{\infty}\frac{|m^n_{k}-m_{k}|}{10^{k+1}} \\
&< \frac{9\times 10}{\mathsf{N}}\sum_{k=0}^{\infty}\frac{|m^n_{k}-m_{k}|}{10^{k+1}} = 10\times d_M(m^n,m) \longrightarrow 0,
\end{aligned}$$

which establishes the convergence of the first component, and so $f_{\mathcal{E},\mathsf{k}}(X^n) \longrightarrow f_{\mathcal{E},\mathsf{k}}(X)$.

Let us now introduce the graph $\mathcal{G}\left(f_{\mathcal{E},\mathsf{k}}\right)$:

- its vertices are the block messages of $\mathbb{B}^{\mathsf{N}}$,
- its edges are couples of $\mathbb{B}^{\mathsf{N}} \times \mathbb{B}^{\mathsf{N}}$ such that there is an edge from e_1 to e_2 labeled by (m_1, m_2) if, and only if $e_2 = m_2 \oplus \mathcal{E}_{\mathsf{k}}\left(m_1 \oplus e_1\right)$.

With such a graph, we can prove the following result.

Proposition 2 *If $\mathcal{G}\left(f_{\mathcal{E},\mathsf{k}}\right)$ is strongly connected, then $f_{\mathcal{E},\mathsf{k}}$ is strongly transitive on $(\mathcal{X}, d)$.*

Proof Let (m^1, e_1) and (m^2, e_2) be two points in $\mathcal{X}$, and $\varepsilon > 0$. We are looking for (m', e') in the open ball $\mathcal{B}((m^1, e_1), \varepsilon) = \left\{x \in \mathcal{X} \mid d(x, (m^1, e_1)) < \varepsilon\right\}$ and $n \in \mathbb{N}$ such that $f^n_{\mathcal{E},\mathsf{k}}(m', e') = (m^2, e_2)$.

As (m', e') has to be at a distance of less than ε of (m^1, e_1), based on the definition of d:

- ε can be < 1, it is needed that $e' = e_1$.
- let $k_0 \in \mathbb{N}$ such that $10^{-k_0} < \varepsilon - \lfloor \varepsilon \rfloor \leqslant 10^{-k_0-1}$. So, having regard to the construction of d_M, it is necessary that $\forall k \leqslant k_0, m'_k = m_k$, in order to have $(m', e') \in \mathcal{B}\left((m^1, e_1), \varepsilon\right)$.

Let us consider $\check{e} = f^{k_0}_{\mathcal{E},\mathsf{k}}\left((m^1, e_1)\right)$. $\mathcal{G}\left(f_{\mathcal{E},\mathsf{k}}\right)$ being strongly connected, there is a path

$$((\check{m}_1, \check{m}_2); (\check{m}_3, \check{m}_4); \ldots; (\check{m}_{k_1}, \check{m}_{k_1+1}))$$

starting from $\check{e}$ and arriving to e_2. And so

$$f^{k_0+2k_1}_{\mathcal{E},\mathsf{k}}\left((m_0, \ldots, m_{k_0}, \check{m}_1, \ldots, \check{m}_{k_1+1}, m^2_0, m^2_1, \ldots), e_1)\right)$$

has e_2 as Boolean vector and m^2 as sequence, which establishes the strong transitivity of $f_{\mathcal{E},\mathsf{k}}$.

We are now able to prove that:

Proposition 3 *If $f_{\mathcal{E},\mathsf{k}}$ is strongly transitive, then it is regular on $(\mathcal{X}, d)$.*

Proof Let us consider $(m^1, e_1) \in \mathcal{X}$ and $\varepsilon > 0$. We are looking for a point $(m^2, e_2) \in \mathcal{X}$ that is both periodic and within $\mathcal{B}\left((m^1, e_1), \varepsilon\right)$. We will now proceed as in the previous proof, by constructing the desired point.

- As ε can be < 1, we need $e_2 = e_1$.
- Let $k_0 \in \mathbb{N}$ such that $10^{-k_0} < \varepsilon - \lfloor \varepsilon \rfloor \leqslant 10^{-k_0-1}$. Again, due to the definition of d_M, we must have: $\forall k \leqslant k_0, m_k^2 = m_k^1$.

Let us consider the point $(m^3, e_3) = f_{\mathcal{E},\mathsf{k}}^{k_0}\left((m^1, e_1)\right)$. $f_{\mathcal{E},\mathsf{k}}$ being strongly transitive, there is $(m^4, e_4 = \text{in}\, \mathcal{B}\left((m^3, e_3), \dfrac{1}{10}\right)$ and $k_1 \in \mathbb{N}$ such that $f_{\mathcal{E},\mathsf{k}}^{k_1}\left((m^4, e_4)\right) = (m^1, e_1)$. But then, the point

$$\left((m_0^1, \ldots, m_{k_0}^1, m_0^3, \ldots, m_{k_1}^3, m_0^1, \ldots, m_{k_0}^1, m_0^3, \ldots, m_{k_1}^3, \ldots); e_1\right)$$

is such that:

- it is at ε from (m^1, e_1),
- it is periodic with period $k_0 + k_1 + 2$, by construction.

We have now all the ingredients to prove that:

Theorem 3 *If $\mathcal{G}\left(f_{\mathcal{E},\mathsf{k}}\right)$ is strongly connected, then the OFB mode $f_{\mathcal{E},\mathsf{k}}$ is chaotic according to Devaney.*

Proof $\mathcal{G}\left(f_{\mathcal{E},\mathsf{k}}\right)$ being strongly connected, we can conclude that $f_{\mathcal{E},\mathsf{k}}$ is strongly transitive, and so transitive. Then, due to Proposition 3, it is regular too. Thanks to the Banks theorem, $f_{\mathcal{E},\mathsf{k}}$ is sensible to the initial conditions. And so it is chaotic, as defined by Devaney.

Based on Theorem 3, it is now possible to test if the OFB mode applied on our proposed encryption scheme will exhibit the properties of sensitivity to the initial conditions, expansiveness, and so on. Such properties are useful when the avalanche effect or key sensitivity are needed. This can be verified by following an approach similar to what has been done in Section 5.4.

8 Conclusion and perspectives

In this paper, a new lightweight and flexible cipher candidate has been proposed. The robustness and speed of the proposed cipher are achieved by reducing the number of rounds and employing a variable structure of SP-boxes. The r S-boxes and P-boxes are designed in a lightweight manner, and the required number of iterations needed to construct the boxes is quantified in addition to the number of rounds needed to reach the avalanche effect. Moreover, a linear transformation is employed to form the P-boxes, while a nonlinear one is employed to form the S-boxes. The proposed cipher can effectively resist the differential/linear, statistical, and brute-force attacks: theoretical analysis and experimental results prove that the proposed cipher offers a high-security level and low computational complexity when compared with DES, 3DES, AES, ECKBA, Kamur, and Yang algorithms. From a theoretical aspect, one contribution is the rewriting of the OFB mode as a discrete dynamical system on a relevant metric space. Its dynamics were studied using the mathematical topology, and the ways to have a chaotic dependence between the original message and the cipher one have been emphasized through a well-defined graph.

In future work, we intend to further analyze the OFB mode of operation using the proposed reformulation. It will be compared to other existing modes by measuring, for instance, their ergodicity, metrical entropy, etc. A more formal relationship will be developed, to properly relate the key sensitivity and the avalanche effect to definitions taken from measure theory. Finally, our cipher will be refined and more thoroughly compared against the state-of-the-art.

Acknowledgement

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 Labex ACTION program (contract ANR-11-LABX-01-01).

References

[1] Abdessalem Abidi, Christophe Guyeux, Jacques Demerjian, Belagacem Bouallègue, and Mohsen Machhout. Lyapunov exponent evaluation of the cbc mode of operation. In *CHAOS 2017, 10th CHAOS International Conference*, pages 185–196, Barcelona, Spain, May 2017.

[2] Abdessalem Abidi, Qianxue Wang, Belgacem Bouallegue, Mohsen Machhout, and Christophe Guyeux. Proving chaotic behavior of cbc mode of operation. *International Journal of Bifurcation and Chaos*, 26(07):1650113, 2016.

[3] Abdessalem Abidi, Qianxue Wang, Belgacem Bouallegue, Mohsen Machhout, and Christophe Guyeux. Quantitative evaluation of chaotic cbc mode of operation. In *Advanced Technologies for Signal and Image Processing (ATSIP), 2016 2nd International Conference on*, pages 88–92. IEEE, 2016.

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

[5] JM Amigo, Ljupco Kocarev, and Janus Szczepanski. Theory and practice of chaotic cryptography. *Physics Letters A*, 366(3):211–216, 2007.

[6] Jacques Bahi, Xiaole Fang, Christophe Guyeux, and Qianxue Wang. Evaluating quality of chaotic pseudo-random generators. application to information hiding. *IJAS, International Journal On Advances in Security*, 4(1-2):118–130, 2011.

[7] Jacques Bahi and Christophe Guyeux. Hash functions using chaotic iterations. *Journal of Algorithms and Computational Technology*, 4(2):167–181, 2010.

[8] Elaine Barker and John Kelsey. Recommendation for random number generation using deterministic random bit generators. *NIST Special Publication*, 800:90A, 2012.

[9] Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. *Journal of CRYPTOLOGY*, 4(1):3–72, 1991.

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

[11] Joan Daemen, René Govaerts, and Joos Vandewalle. A new approach to block cipher design. In *Fast Software Encryption*, pages 18–32. Springer, 1994.

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

[13] Donald W Davies and Graeme IP Parkin. The average cycle size of the key stream in output feedback encipherment. In *Workshop on Cryptography*, pages 263–279. Springer, 1982.

[14] R. L. Devaney. *An Introduction to Chaotic Dynamical Systems.* Addison-Wesley, Redwood City, CA, 2nd edition, 1989.

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

[16] Christophe Guyeux and Jacques Bahi. A topological study of chaotic iterations. application to hash functions. In *CIPS, Computational Intelligence for Privacy and Security*, volume 394 of *Studies in Computational Intelligence*, pages 51–73. Springer, 2012. Revised and extended journal version of an IJCNN best paper.

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

[18] C. Li, D. Lin, and J. Lü. Cryptanalyzing an image-scrambling encryption algorithm of pixel bits. *IEEE Transactions on Multimedia*, 24(3):64–71, 2017.

[19] James L Massey. Safer k-64: A byte-oriented block-ciphering algorithm. In *Fast Software Encryption*, pages 1–17. Springer, 1994.

[20] Naoki Masuda, Goce Jakimoski, Kazuyuki Aihara, and Ljupco Kocarev. Chaotic block ciphers: from theory to practical algorithms. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 53(6):1341–1352, 2006.

[21] Mitsuru Matsui. Linear cryptanalysis method for des cipher. In *Advances in Cryptology–EUROCRYPT'93*, pages 386–397. Springer, 1994.

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

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

[24] Ronald L Rivest, MJB Robshaw, Ray Sidney, and Yiqun Lisa Yin. The rc6tm block cipher. In *First Advanced Encryption Standard (AES) Conference*, page 16, 1998.

[25] Joseph L. Rodgers and Alan W. Nicewander. Thirteen Ways to Look at the Correlation Coefficient. *The American Statistician*, 42(1):59–66, 1988.

[26] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, and Elaine Barker. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical report, Booz-Allen and Hamilton Inc Mclean Va, 2001.

[27] Claude Elwood Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[28] Daniel Socek, Shujun Li, Spyros S Magliveras, and Borko Furht. Short paper: Enhanced 1-d chaotic key-based algorithm for image encryption. In *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*, pages 406–407. IEEE, 2005.

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

[30] AF Webster and Stafford E Tavares. On the design of s-boxes. In *Advances in Cryptology—CRYPTO'85 Proceedings*, pages 523–534. Springer, 1986.

[31] Eric Yong Xie, Chengqing Li, Simin Yu, and Jinhu Lü. On the cryptanalysis of fridrich's chaotic image encryption scheme. *Signal Processing*, 132:150 – 154, 2017.

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

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