One round cipher algorithm for multimedia IoT devices

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

One Round Cipher Algorithm for Multimedia IoT Devices

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

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

Abstract

With the exponential growth in Internet-of-Things (IoT) devices, security and privacy issues have emerged as critical challenges that can potentially compromise their successful deployment in many data-sensitive applications. Hence, there is a pressing need to address these challenges, given that IoT systems suffer from different limitations, and IoT devices are constrained in terms of energy and computational power, which renders them extremely vulnerable to attacks. Traditional cryptographic algorithms use a static structure that requires several rounds of computations, which leads to significant overhead in terms of execution time and computational resources. Moreover, the problem is compounded when dealing with multimedia contents, since the associated algorithms have stringent QoS requirements. In this paper, we propose a lightweight cipher algorithm based on a dynamic structure with a single round that consists of simple operations, and that targets multimedia IoT. In this algorithm, a dynamic key is generated and then used to build two robust substitution tables, a dynamic permutation table, and two pseudo-random matrices. This dynamic cipher structure minimizes the number of rounds to a single one, while maintaining a high level of randomness and security. Moreover, the proposed cipher scheme is flexible as the dimensions of the input matrix can be selected to match the devices' memory capacity. Extensive security tests demonstrated the robustness of the cipher against various kinds of attacks. The speed, simplicity and high-security level, in addition to low error propagation, make of this approach a good encryption candidate for multimedia IoT devices.

Keywords: Internet of Things, Multimedia Internet of Things, Lightweight Image Encryption Algorithm, Substitution, Permutation, Dynamic cryptographic primitives.

1 Introduction

The increasing number of services provided by the Internet has generated a huge increase in the number of connected devices; more than 9 billion network devices are connected and used by billions of users. Services offered can be used for communication, entertainment, sharing knowledge and many other purposes. In addition to traditional devices (laptops, smart phones, etc..), devices

around us will soon be able to communicate with each other [1, 2]. These smart objects, interacting with each other, have transformed the Internet into the "Internet of Things", which is an emerging area in which highly constrained interconnected devices work together to accomplish a specific task and can be used for many purposes such as monitoring and collecting data as well as accessing and processing such data. Indeed, IoT is continuously emerging in many fields such as smart houses/buildings/cities, environment monitoring, traffic monitoring, health monitoring, and even in human bodies for patient monitoring in what is being referred to as mHealth [3, 4].

However, the major problems that hinder the deployment of IoT systems are the security and privacy issues [5, 6, 7], since such systems are more susceptible to diverse kinds of attacks (passive and active) than the traditional systems. The passive attacks can seriously impair the confidentiality of the data by trying to extract the contents of transmitted packets, while active attacks can compromise the data integrity and authentication by inserting, deleting or modifying the packets' contents. One solution to guard against such kinds of attacks is to encrypt the packets transmitted by IoT nodes. Hence, it is necessary to ensure that the transmitted data is secured from any unauthorized access and that data is exchanged only between legitimate parties. In this paper, we address the problem of securing the distributed multimedia systems in IoT, called Multimedia Internet of Things (MIoT) [8], which consists of cameras and microphones as shown in Figure 1.

There are various classes of multimedia IoT devices that are used for different services such as:

- Streaming of stored multimedia content such as audio, video, and so on;
- Live streaming of multimedia content in the cases of video conferencing, online gaming, and so on;
- Real-time interactive multimedia communication such as the case of surveillance.

One of the mostly used MIoT devices is surveillance cameras that are essential for monitoring public and private areas to detect suspicious activities. Typically, the transmissions of these cameras should be secured from eavesdropping and malicious attacks to avoid disclosing any useful information to attackers. MIoT applications have stringent QoS requirements and require security solutions that may entail major resources and latency overhead. This in turn is not practical for MIoT devices that, in some scenarios, might be limited in battery lifetime and computational resources.

As such, multimedia IoT constrained devices may not be able to support the NIST-approved strong cryptographic algorithms since these have a negative impact on the system performance and may degrade the desirable QoS [9], especially since multimedia devices in IoT systems exchange massive amounts of data.

1.1 Related Works

Encryption can be realized using symmetric or asymmetric algorithms. In a practical implementation, the symmetric-key scheme is preferred since it requires less computational complexity, memory consumption, and resources when compared to asymmetric ones. Furthermore, symmetric cipher

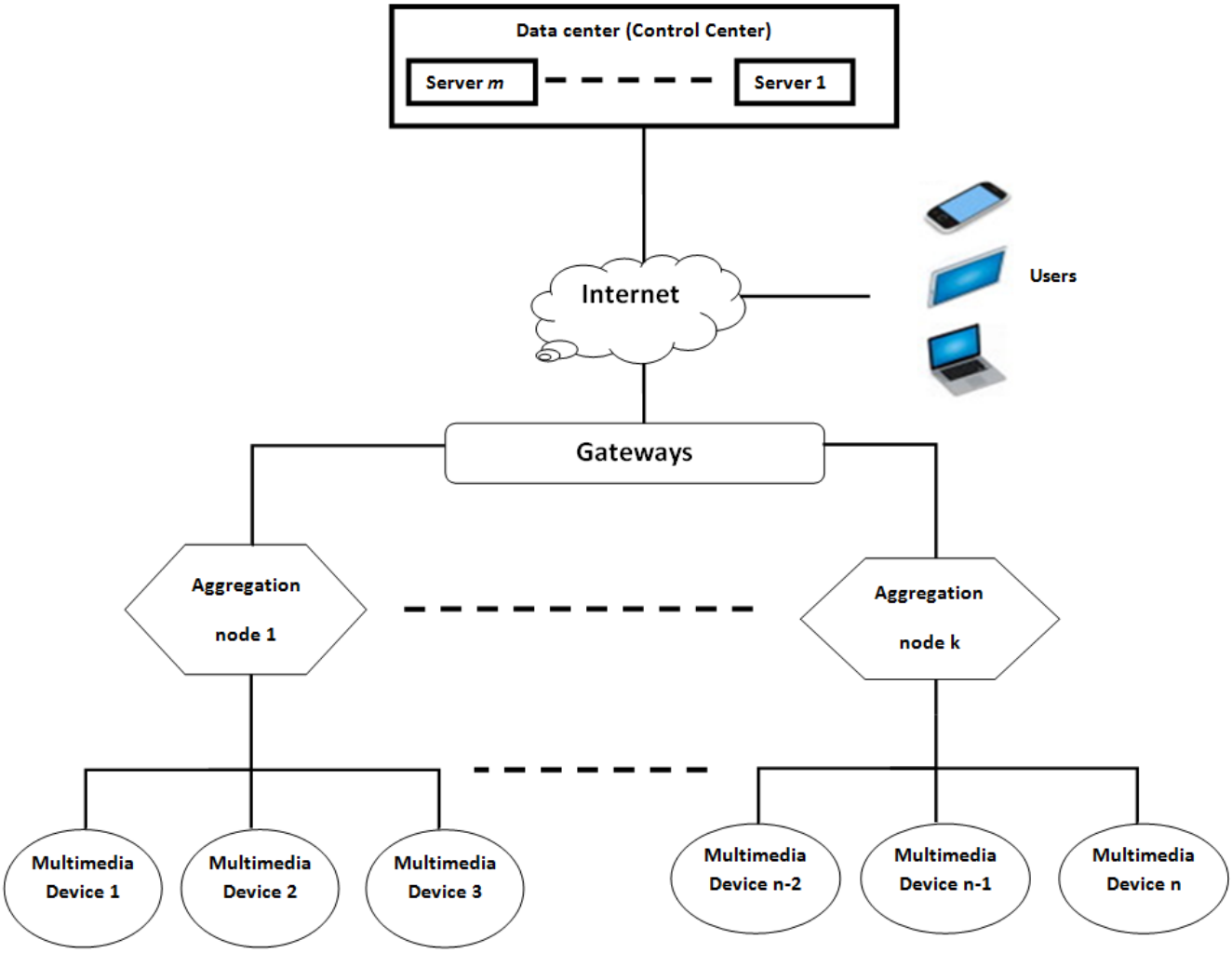


Figure 1: Multimedia IoT approach.

schemes can be divided into two classes: **Stream ciphers and Block ciphers**. For real IoT implementations [6], AES block cipher [10] in Counter (CTR) mode [11, 12] is used, where the ciphering process is independent of plain-text and in this case, block cipher operates as a stream cipher. In general, block cipher such as AES uses a multi-round structure whereby a round function undergoes several iterations r. Round functions can be based on either Feistel Networks (FN) or Substitution-Permutation Networks (SPN). Typically, in each round, the round function applies several operations to ensure the confusion and diffusion properties.

For more than a decade, many efforts have been spent to make AES act as a lightweight block cipher and thus, to make it practical for tiny-limited devices. Indeed, several improvements have been realized to reach this goal in both hardware and software implementations. Examples of these variations include AES-128 hardware ASCI implementation with 2400 Gate Equivalents (GE) is

presented in [13], while efficient software AES implementations for 8-bit in [14], 16-bit in [15] and 32-bit in [16]. Moreover, AES optimized instructions were added to the instruction set of Intel's [17, 18]. At the time of this writing, there are no further optimization techniques to AES, and there might be no more possible optimization [19].

However, even with the current optimization to AES and the attempts to make it applicable to constrained devices and adaptive to the increasing data rates, the enhancements still suffer from several limitations that cannot be easily overcome. According to NIST [9], AES might not meet the future requirements and consequently, a new project has been launched to design new lightweight cryptographic algorithms with lower number of GE (preferably, less than 2,000 [20]).

Even though the AES hardware implementation is fast, however, the implementation itself is complex and is not suitable for constrained devices [19]. On the other hand, a simple AES hardware implementation decreases its efficiency. Consequently, AES hardware implementation suffers from a trade-off between efficiency and implementation complexity [19]. Moreover, the presented implementation in [21] shows that AES rapidly decreases the lifetime of battery nodes and networks such as in ZigBee [22], and WirlessHART [23]. As such, it is safe to conclude that AES is not really suitable for constrained devices, such as IoT ones, as stated in [9, 19, 24].

Consequently, a different cipher methodology has been presented recently [19] to solve this issue by reducing the size of the secret key and/or the block size to meet the constrained requirements of tiny devices. Examples of such techniques include LEA [25], FeW[26], Prince [27], TWINE [28], Lblock [29], Piccolo [30], LED [31] and other ciphers, a list of which is shown in Table 1.

However, all of these ciphers are based on static substitution and diffusion primitives, which require iterating the round function for a large number of r (see Table 1) to ensure the required security level. Unfortunately, the multi-round structure that is used in the these ciphers provides a high level of security but at the expense of high computational complexity. Therefore, to use these cipher algorithms in MIoT devices, there is a trade-off between security and performance: the required execution time of these ciphers is r times the round function's execution time, and the required resources are also multiplied by r.

The incorporation of encryption into MIoT devices introduces an overhead that might prevent such devices from ensuring their main functionality and consequently impacting the overall system performance [9]. Accordingly, the limitations of IoT devices mandate the design of a new cipher methodology that can ensure secure data transmission among IoT nodes with low computational complexity and resources. This paper presents a new methodology to reduce r to 1 and to form a dynamic key-dependent lightweight round function to fulfill the security aspects efficiently. Consequently, this paper proposes a new cipher design methodology that may help in the design of future lightweight cryptographic algorithms.

1.2 Motivation & Contribution

In this paper, we present a new efficient, lightweight cipher algorithm for MIoT applications. Contrary to other cryptography algorithms (multiple rounds), the proposed cipher only employs a

Table 1: List of recent lightweight cryptographic algorithms

Algorithm	No. of rounds	Key size	Block size	Structure
TEA	64	128	64	FN
XTEA	64	128	64	FN
LEA [25]		128	64	FN
HEIGHT	32	128	64	GFS
FeW[26]	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	FNl
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 [27]	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

single dynamic key-dependent round function. The proposed round function is based on simple operations and achieves the required cryptographic performance. To accomplish this objective, we relied on the dynamic key approach whereby a dynamic key is generated for each input audio, image or video. This dynamic key depends on a secret key and a Nonce similar to [32]. Then, this dynamic key is used to build several efficient key-dependent diffusion and confusion primitives, which ensure a good cryptographic performance [33, 34].

The proposed cipher scheme includes several contributions that led to a high level of efficiency and security for IoT devices compared to the recent lightweight block ciphers, recent chaotic ciphers and our previous dynamic key-dependent dynamic cipher schemes.

System performance

- **Lightweight**:The minimum required number of iterations, for recent lightweight cryptographic algorithms, is 4 such as the Hummingbird2 cipher. Furthermore, the recent lightweight chaotic image encryption algorithms such as [35, 36, 37, 38, 39] use also the multi-round structure in addition to floating-point calculations and conversion operations, which introduces an important overhead in terms of latency and required resources. In addition, [36] requires asymmetric encryption, which requires more resources and introduces a higher latency [6] when compared to the symmetric one.

  There is only one chaotic cipher that was presented with a single round [40], but it requires a huge memory capacity. Also, according to [41], the results are not accurate and the approach actually requires at least 6 iterations to reach the desired cryptographic performance. Moreover, the approach suffers from maximum error propagation, since the avalanche effect propagates to the whole image instead of being restricted to the block level. Our previous

dynamic key-dependent cipher schemes [42, 32] require at least two rounds of substitution and diffusion. The scheme we propose in this paper avoids the use of a static diffusion operation such as the MixColumn transformation of AES [10] or the key-dependent integer/binary diffusion operations of [42, 32], since such operations consume a high percentage of the execution time [32, 43]. Hence, the proposed scheme requires only one round iteration of a lightweight simple and flexible round function without using any diffusion operation. As such, this minimizes the computational complexity of the proposed cipher and consequently the required latency and resources. Moreover, the proposed encryption scheme can be realized in parallel, while the decryption algorithm can be partially parallelized.

- **Flexibility:** The proposed cipher operates on data at the sub-matrix level, which can have a flexible size of ($h \times h$ bytes), to be set according to the devices' limitations. In other words, the proposed approach is configured according to the devices' characteristics.

- **Simple hardware and software implementations:** The proposed cipher is based on logical operations (exclusive or), load and store operations (substitution and permutation), which renders the corresponding hardware and software implementations to be simple and efficient.

- **Low error propagation**: Since the encryption is done at the sub-matrix level, an error that occurs in a byte of an encrypted sub-matrix will affect only two bytes and it will not affect the whole corresponding sub-matrix. In addition, the proposed cipher scheme is designed to avoid the chaining operations to limit the effect of the corrupted bytes only to both sub-matrices. This will not affect all of the two sub-matrices as in [42, 32] or the whole image as in [40]. Thus, low error propagation is guaranteed and an error correction scheme is presented, which is a great advantage for the proposed cipher scheme.

These enhancements reduce the delay of the encryption and decryption processes and simplify their corresponding hardware implementations. This is essential since each primitive has its own impact on the security and efficiency of the proposed cipher scheme.

Security Performance

- **Key Dependence Approach:** The proposed cipher is based on key-dependent substitution and permutation primitives that ensure simplicity in addition to the required cryptographic properties.

- **Dynamic Key Approach:** In contrast to the existing cipher solutions, the proposed approach is based on a dynamic key, which is variable and changes in a pseudo-random manner for each new session. The periodic interval of a session depends on the application or user requirements. For example, a new session can be established for each new input image. Therefore, the cryptanalysis process against the proposed cipher algorithm is very challenging because of the unpredictability of the cipher primitives as they change according to the dynamic key. In addition, changing the dynamic key produces different cipher primitives and consequently different encrypted/decrypted images (sensitivity is verified in Section 4). The dynamic nature of the proposed cipher provides high robustness against any kind of attacks. [32, 42, 44, 45].

- **Dynamic Sub-matrices Selection**: A dynamic pseudo-random selection operation is introduced to control and randomize the sequential order of the encryption and decryption sub-matrices. This complicates the procedure for the possible attacks. This makes the proposed cipher approach more robust compared to existing ones since the sequential order of encryption/decryption is variable and depends on the dynamic key. This step is designed and realized to achieve low latency and resources overhead towards preserving the previous advantages.

Therefore, an efficient collaboration scheme is proposed and relates substitution, pseudo-random matrices and block selection (based on a generated permutation table) to reach a better level of robustness and efficiency. This is proved according to the results of a set of performance and security tests.

Accordingly, a good lightweight, flexible, cipher candidate for MIoT is proposed. This is justified since the trade-off between system performance and the security level is reduced in addition to its simple hardware and software implementations.

1.3 Organization

The rest of the paper is organized as follows. Section 2 presents the proposed key derivation algorithm along with the proposed cipher construction primitives. In Section 3, we describe all the steps necessary to undergo the encryption and decryption processes. Then, an extensive security analysis and a performance evaluation are conducted in Section 4 to prove the robustness and effectiveness of the proposed scheme. Then, in Section 5, we prove the immunity of the proposed algorithm against different kinds of existing attacks. Finally, in Section 6, the conclusions are drawn along with directions for future work.

2 Initialization

In this section, the generation process of the dynamic key and the associated sub-keys that are used in the cipher are explained. Figure 2 illustrates the key derivation function, which takes as input a secret key SK and a nonce N_o that are unique for every session or input image. These inputs are described in the following:

- **Secret key SK:** This secret key is only shared between the communicating entities after the mutual authentication step (handshake). For better protection, the secret key is changed after a specific period of time to be specified by the underlying application. It can be renewed in different ways such as using Binary Elliptic Curve Diffie Hellman protocol (ECDH) [46].
- **Nonce N_o:** A pseudo-random generator is used to generate this $Nonce$. It is important to generate a new $Nonce$ for each input image. N_o can be sent to the receiver encrypted using the shared public key of the other entity if the asymmetric approach is used. Another way for sharing N_o is to have a good synchronization between the sender and the receiver where each entity derives it separately with no need for transmission and starting from the same seed.

Table 2: Summary of notations used.

Notation	Definition
SK	Secret Key
N_o	Nonce
DK	Dynamic Key
K_{S1}	First dynamic substitution sub-key used to construct $S-box_1$
K_{S2}	Second dynamic substitution sub-key used to construct $S-box_2$
K_{RM}	Matrix dynamic sub-key used to create RM_1 and RM_2
K_P	Permutation sub-Key used to create π
S_1	The first produced dynamic S-box
S_1^{-1}	The inverse corresponding of the first S-box
S_2	The second produced dynamic S-box
S_2^{-1}	The inverse corresponding of the second S-box
RM_1	First initial dynamic pseudo-random matrix
RM_2	Second initial dynamic pseudo-random matrix
π	A Dynamic produced permutation table (P-box)
π^{-1}	The inverse corresponding the permutation table (P-box)
C	Columns Number
R	Rows Number
P	Plane Number (for gray-scale is equal to 1)
$h \times h$	The size of each sub-matrix
α	The number of $h \times h$ sub-matrices
x_i	Original plain sub-matrix at the i^{th} index
y_i	The corresponding permuted sub-matrix of x_i
cx_i	The corresponding encrypted sub-matrix of x_i
cy_i	The corresponding encrypted sub-matrix of y_i

Then, the secret key SK and N_o are Xored and its corresponding output is hashed to produce the dynamic key DK. Next, DK is divided into four different sub-keys that form the seeds for the different cipher primitives and these are described in the following subsections. Let us indicate that the secure hash function ($SHA-512$) is selected for this step since it posses the best desirable cryptographic hash properties such as the high resistance against collision. This can assure that the produced DK is renewed for every session or input image and consequently provides different cipher primitives; which introduces randomness into the scheme. Employing dynamic key will provide high immunity against existing and modern attacks.

2.1 Dynamic Key & Sub-keys Derivation

Indeed, DK can be changed frequently as needed by the user or by the application. Furthermore, as $SHA-512$ is used, DK has 512 bits length (64 bytes) and it will be split into four sub-keys, where each one has a size 128 bits (16 bytes). These sub-keys are $\{K_{S1},\ K_{S2},\ K_{RM},\ K_p\}$ and are described in the following knowing that each of the sub-keys will be used for a different purpose.

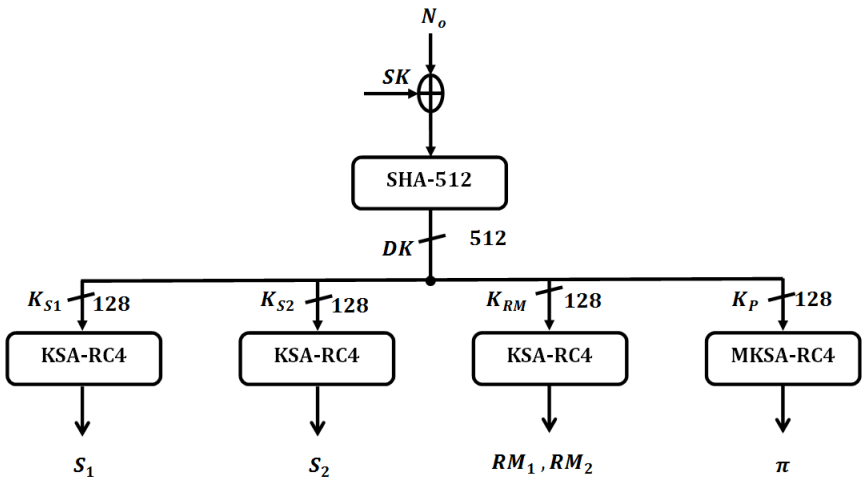


Figure 2: The Proposed Dynamic Sub-Keys Generation Scheme.

- **First substitution sub-key** K_{S1}: It consists of the most significant 16 bytes of DK.
- **Second substitution sub-key** K_{S2}: It consists of the next most significant 16 bytes of DK.
- **Dynamic matrices sub-key** K_{RM}: K_{RM} consists of the third most significant 16 bytes.
- **Permutation sub-key** K_P: Finally, the least significant 16 bytes of DK.

Table 2 shows all the notations used in this paper. The derived dynamic key is renewed for each input image and any bit changed in it will lead to a completely different set of sub-keys and consequently different cipher primitives will be produced. Next, the construction of the proposed cipher primitives that are based on these four sub-keys is described.

2.2 Construction of Cipher Primitives

We aim to design a simple, yet very effective lightweight cipher algorithm with only one round, which can be used with constrained devices in multimedia IoT. According to Shannon, round function should mandatory ensure the confusion and diffusion properties. While this round function should be iterated for multi-iterations to consider it as a successful cipher scheme. This is the logic of the existing symmetric block cipher algorithms since static substitution and diffusion primitives are mainly used. Moreover, as mentioned previously, static keys would render the system vulnerable to many future threats [44, 45, 47]. Fortunately, a dynamic key approach can provide a better-desired security level against existing and future powerful attacks [32]. This is ensured since all the used sub-keys depend on the produced dynamic key and consequently, cipher primitives become variable, which prevents attackers to recover any information from the collected set of original encrypted images. More important, this helps cryptographic engineering reduce the required round number of iterations and consequently, this will reduce the required resources and latency that are both necessary to preserve the main functionality of MIoT devices. Therefore, defining key dependent cipher primitives with a high level of efficiency and security is necessary for the proposed

dynamic approach. In the following, the proposed techniques to construct the key dependent cipher primitives are explained.

2.2.1 Dynamic Substitution Primitive

In general, substitution operation is used to ensure the confusion property, and it is considered as a non-linear primitive. The proposed cipher scheme requires two substitutions table (S_1 and S_2). In this paper, we propose to use the KSA of the $RC4$ stream cipher algorithm [48] to produce the required two substitution boxes (S-boxes). KSA algorithm is described in Algorithm 1, where an input key with L bytes is introduced to produce a dynamic substitution table S as an output. The size of the sub-substitution dynamic keys (K_{S1} and K_{S2}) is set to $L = 16$ bytes.

Algorithm 1 KSA for RC4

```
1: procedure RC4_KSA(K = {k_1, k_2, ..., k_L}, L)
2:     for i ← 0 to 255 do
3:         S[i] ← i
4:     end for
5:     j ← 0
6:     for i ← 0 to 255 do
7:         j ← (j + S[i] + k[j mod L]) mod 256
8:         swap(S[i], S[j])
9:     end for
10:    return S
11: end procedure
```

Therefore, K_{S1} is used as a key for the KSA algorithm to produce the first substitution table S_1. Similarly, S_{k2} is used to produce the second dynamic substitution $S - box$, S_2. Moreover, the proposed technique to build key-dependent substitution tables S showed good robustness and cryptographic strength according to several criteria that are summarized as follows:

- **Linear Probability Approximation Function (LPF)**: represents the maximum nonlinear probability of an S-box. Towards reaching a better resistance against linear attacks, LPF should be very low. For $L \geq 4$, the average LPF value stabilizes and becomes close to $2^{-4.8}$.
- **Differential Probability Approximation Function (DPF)**: represents the maximum differential uniformity of an S-box. Similarly, to provide a better resistance against differential attacks, DPF should reach a very low value. For $L \geq 4$, the average of DPF converges to the minimum possible value, which is $2^{-4.5}$.
- **Strict Avalanche Criterion (SAC)**: An S-box S satisfies the SAC property if a single input bit change causes the output substituted bit to change with a probability of at least half. For $L \geq 4$, the produced $S - boxes$ become more close to the ideal value. This criterion is important, since it quantifies the sensitivity probability against any modification on any bit and it helps to ensure the avalanche effect if a good diffusion primitive is used.
- **Output Bit Independence Criterion (BIC)**: This criterion is important to quantify the independence among the bits (for each byte) of the produced substitution table. It specifies

that two output bits, j and k for each substituted byte, should change independently when a single input bit i is changed. In fact, the BIC value become very close to the desired value (0.5) for $L \geq 4$.

Table 3: The values of LPF, DPF, SAC, and BIC for $L = 4$ iterations.

LPF	DPF	SAC	BIC
$2^{-4.8}$	$2^{-4.5}$	0.5	0.51

The average values of these criteria are shown in Table 3 for $L = 4$. The obtained results were sufficient to indicate that the proposed construction technique of key-dependent substitution produces a robust and efficient substitution table (S-box). Furthermore, S_1 and S_2 make the proposed cipher algorithm with one round immune against differential and linear attacks since they are changed in a pseudo-random manner.

On the other hand, the inverse substitution table is necessary for the decryption process. Indeed, as the produced S is bijective, the inverse of S, S^{-1}, can be obtained easily by the following operation $\mathrm{S}^{-1}[\mathrm{S(i)}]=\mathrm{i}$.

2.2.2 Dynamic Selection Sub-matrices

The proposed cipher algorithm requires producing a dynamic key dependent flexible permutation table π. Indeed, π is not only used to permute the sub-matrices of the input image (α sub-matrices), but also to control the encryption/decryption processes. Let us indicate that the proposed cipher scheme requires two sub-matrices as input, one at a time. In fact, the second sub-matrix is chosen according to the permutation table π. The proposed technique to build the dynamic flexible permutation table π is similar to that of the substitution tables. It is based on a minor modification of the KSA of RC4 (see Algorithm 1), which requires to replace α instead of 255 in lines 2 and 6. The modification is presented in KSA-RC4 (simple and efficient) to be sure that the same hardware implementation of KSA can be used to construct the substitution and permutation primitives in a real implementation and to reach a lower number of GE.

Therefore, K_p is used as a seed for the proposed modified KSA algorithm to build the flexible key dependent permutation table π with length α elements. Moreover, The i^{th} original/encrypted sub-matrix (x_i) requires the $\pi(i)^{th}$ original/encrypted ($X_{\pi(i)}$) to be encrypted or decrypted, respectively. Where $\pi(i)$ represents the value of the π at the i^{th} index and $1 \leq \pi(i) \leq \alpha$. The process of permutation is realized by employing a swap function, where (i) and $(\pi(i))$ are the original and permuted sub-matrix positions of the image, respectively.

2.2.3 Dynamic Pseudo Random Matrices

The proposed cipher requires two sub-matrices RM_1 and RM_2 in the encryption/decryption process to ensure better randomness properties and to remove any existing patterns from the encrypted sub-matrices. $RC4$ algorithm with K_{RM} is used to produce $2 \times h^2$ bytes, where the first h^2 bytes are used to form RM_1 and the last h^2 bytes ares used to form RM_2. Let us indicate that the PRGA

algorithm of $RC4$ should be used in addition to KSA in this step.

Note that the choice of RC4 is due to its simple software and hardware implementations and its ability to generate substitution and permutation primitives with good cryptographic performance. In this paper, RC4 is used only to produce the cipher primitives and not for the encryption/decryption process. However, any other key-dependent substitution/permutation generation algorithm can be used as well.

3 Encryption and Decryption Algorithms

In this section, the different steps of the encryption and decryption algorithms are shown and they are described in Figures 3 and 4, respectively.
The proposed algorithm is symmetric and is based on a secret key SK shared between the sender and the receiver. As stated earlier, this key is employed with a $Nonce$ to produce a dynamic key, which is split to obtain four sub-keys that will be used to construct the primitives of the encryption/decryption processes. This cipher is based on only **one round** since a dynamic key with a large size is used. In the encryption process, an input image of size $C \times R \times P$ is divided into α sub-matrices $\{x_1,\ x_2,\ \ldots,\ x_\alpha\}$. Each sub-matrix has a square size equal to $h \times h$ bytes. If the number of bytes of an image is not a multiple of h^2, a padding operation is performed to adjust the size of the last sub-matrix (x_α). In addition, h can be equal to 4, 8, 16 or 32. On the other hand, the sub-matrices number α is obtained as follows:

$$\alpha = \frac{R \times C \times P}{h^2} \tag{1}$$

In the rest of the paper, we fix h to 8. Note that h can be changed according to the device limitations.

3.1 Encryption Algorithm

Algorithm 2 summarizes the proposed encryption algorithm that can be divided into four sub-functions, which are $\mathsf{Sub-MatrixSelection}$, $\mathsf{Function-f}$, $\mathsf{Function-g}$, $\mathsf{SwitchOperation}$ and they are described in the following:

3.1.1 Sub-Matrix Selection

In general, the first step in the encryption/decryption processes is to control the selection of the second sub-matrix of the input image. Usually, this step is introduced to add more randomness and to eliminate the sequential relation between the neighboring sub-matrices elements in an image to introduce more difficulty to attackers. As indicated previously, a dynamic permutation table π is used to control the selection of the second sub-matrix in the encryption/decryption process. A pair of sub-matrices (x_i and y_i) is selected to be encrypted/decrypted at one time. x_i represents the i^{th} sub-matrix of X that can be accessed via $X[i]$ and $i = \{1\ ,\ ,\ ,\alpha\}$. While $y_i = x_{\pi(i)}$ represents the $pi(i)^{th}$ sub-matrix and can be accessed via $X[\pi[i]]$. Next, each couple of sub-matrices x_i and y_i will undergo different operations. Each sub-matrix will be subjected to a different nonlinear function. The sub-matrix x_i will go through a function f, while the sub-matrix y_i will undergo another function g. Both functions are explained below.

3.1.2 Function f

In this step, both dynamic S-boxes, S_1 and S_2, are being used. First, x_i is substituted by employing S_1, and then, the output is Xored with the first dynamic matrix RM_1 and the sub-matrix y_i. This is represented by the following equation.

$$O_i = RM_1 \oplus S_1(x_i) \oplus y_i \tag{2}$$

Next, the output O_i will be subjected to another substitution operation, which employs S_2 as expressed by the following equation:

$$cx_i = S_2(O_i) \tag{3}$$

Function f can be summarized as follows:

$$f = S_2(S_1(x_i) \oplus RM_1 \oplus y_i) \tag{4}$$

3.1.3 Function g

The second round function g is applied in parallel to function f. Sub-matrix y_i will be subjected to this function. First, y_i will undergo a substitution operation by using S_2. Then, the output is Xored with the two dynamic matrices RM_1 and RM_2. This is illustrated by the following equation:

$$O'_i = RM_1 \oplus S_2(y_i) \oplus RM_2 \tag{5}$$

Then, the output O'_i goes through another substitution operation using S_1, and the output will be denoted as cy_i.

$$cy_i = S_1(O'_i) \tag{6}$$

Function g can be summarized as follows:

$$g = S_1(S_2(y_i) \oplus RM_1 \oplus RM_2) \tag{7}$$

3.1.4 Switch Operation

After computing cx_i and cy_i, these two results are switched and hence, cx_i will take the position of cy_i and vice versa. This will add more randomness and will remove any sequential relation between the permuted-substituted sub-matrices.

Finally, all the sub-matrices will be reshaped to form the encrypted image I', which will be sent securely to the desired receiver or will be safely stored. In the next subsection, the decryption algorithm at the receiver side is explained.

3.2 Decryption Algorithm

The decryption scheme is presented in Figure 4. After receiving the encrypted image, the receiver will use the decryption algorithm to recover the original image. The decryption algorithm has minor modifications compared to the encryption algorithm, which are using the inverse function of f (f^{-1}) and of g (g^{-1}). In addition, these inverse functions require also the inverse substitution tables S_1^{-1} and S_2^{-1}.

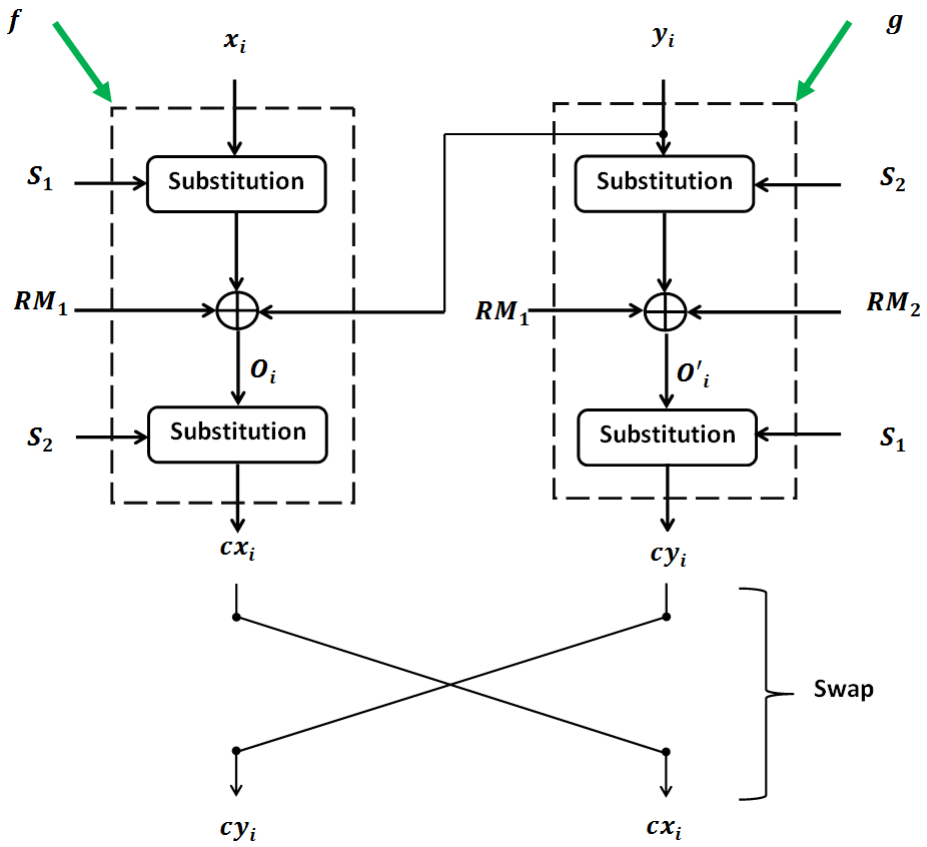


Figure 3: The Proposed Encryption Scheme.

Algorithm 2 The proposed One Round Encryption Algorithm.

procedure ONE_ROUND_ENCRYPTION(X)
  for $i = 1$ to α **do**
    $x_i = X[i]$
    $y_i = X[\pi[i]]$
    $cx_i = S_2(S_1(x_i) \oplus RM_1 \oplus y_i)$
    $cy_i = S_1(S_2(y_i) \oplus RM_1 \oplus RM_2)$
    $X[i] = cx_i$
    $X[\pi[i]] = cy_i$
  end for
end procedure

The other primitive such as π is preserved and there is no need for the inverse permutation box π^{-1}, since the swap function is repeated at the decryptor side. First, the received encrypted image will be reshaped to α sub-matrices to start the decryption process.

Then, the decryption process as indicated in Algorithm 3 will be realized. In the following, we describe only the inverse functions f^{-1} and g^{-1} that are the only difference between the encryption

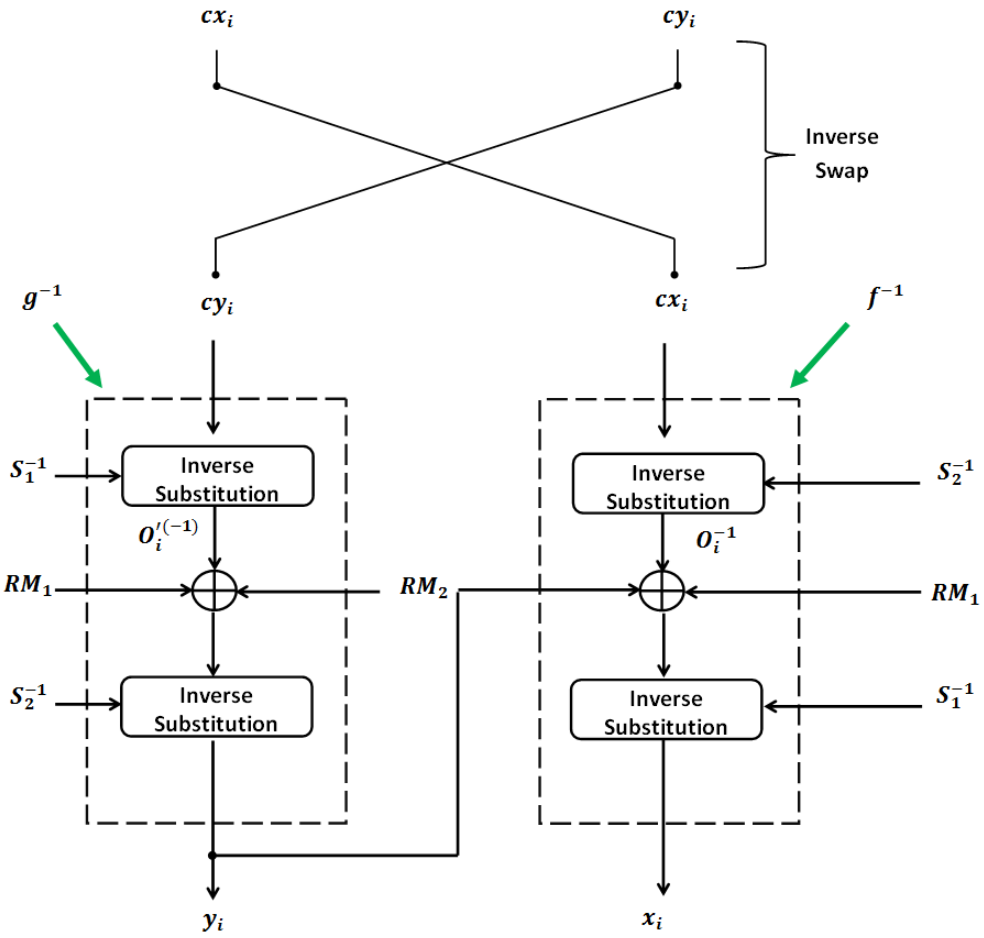


Figure 4: The Proposed Decryption Scheme.

and decryption algorithms.

3.2.1 Inverse of function f (f^{-1})

Each sub-matrix (cx_i) with its corresponding selected sub-matrix cy_i will be processed together. Next, the inverse substitution operation is realized by using S_2^{-1} and is applied on cx_i. Then, this result, O_i^{-1}, will be Xored with RM_1 and y_i, and then subjected to another inverse substitution operation by using S_1^{-1}. This is shown in the following equations:

$$O_i^{-1} = S_2^{-1}(cx_i) \tag{8}$$

After that, the resultant will be the following :

$$x_i = S_1^{-1}(O_i^{-1} \oplus RM_1 \oplus y_i) \tag{9}$$

As it is shown clearly, we need to know y_i and consequently x_i will be deduced. This is where the inverse function g^{-1} is used.

3.2.2 Inverse of function g (g^{-1})

First, S_1^{-1} will be applied on the resultant cy_i. Then, this result will be Xored with RM_1 and RM_2 and will be subjected to another inverse substitution operation S_2^{-1}, respectively as seen in the following equations.

$$O_i'^{(-1)} = S_1^{-1}(cy_i) \tag{10}$$

$$y_i = S_2^{-1}(O_i'^{(-1)} \oplus RM_1 \oplus RM_2) \tag{11}$$

Finally, after obtaining y_i, x_i can be calculated using Equation 7:

$$x_i = S_1^{-1}(O_i^{-1} \oplus RM_1 \oplus y_i) \tag{12}$$

Algorithm 3 One round decryption

```
procedure ONE_ROUND_DECRYPTION(X)
    for i = 1 to α do
        cx_i = X[i]
        cy_i = X[π[i]]
        y_i = S_2^{-1}(S_1^{-1}(cy_i) ⊕ RM_1 ⊕ RM_2)
        X[i] = S_1^{-1}(S_2^{-1}(cx_i) ⊕ RM_1 ⊕ y_i)
        X[π[i]] = y_i
    end for
end procedure
```

In conclusion, this is a simple cipher that reaches the confusion and diffusion properties with just a single round via a dynamic key dependent manner. The efficiency and robustness are demonstrated in the following sections.

4 Security Analysis

In this section, a security analysis for the proposed scheme is performed to demonstrate its robustness against all known confidentiality attacks such as statistical, differential, chosen/known plain-text, and brute-force attacks [49, 50, 51]. Several security experiments were conducted using the standard image Lenna. These experiments are based on different security measures like: statistical analysis, visual degradation, sensitivity. Statistical results of all security tests are shown in Table 4. Accordingly, The obtained results validate the robustness of the proposed approach.

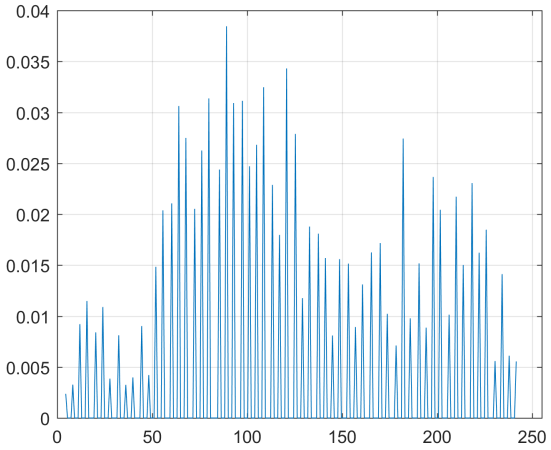
4.1 Statistical Analysis

A cipher scheme requires specific random properties in order to resist efficiently statistical attacks [52]. To prove the effectiveness of the proposed model, several statistical security tests were carried out to validate the uniformity and the independence properties.

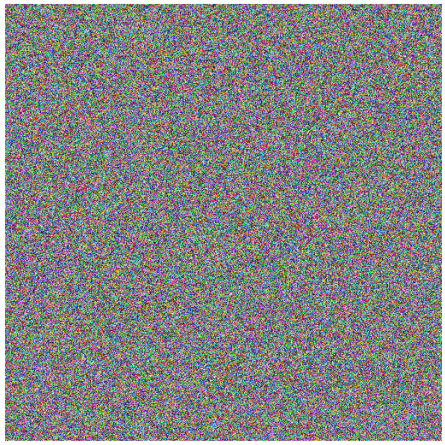
4.1.1 Uniformity Analysis

The encrypted image should possess certain random properties to resist the common statistical attacks. The most commonly used one is the PDF of the encrypted image that should be uniform. This requires each symbol to have a probability close to $\frac{1}{n}$, where n is the number of symbols. The PDF of the original plain-image and its corresponding cipher-image are both shown in Figure 5. It can be seen that the PDF of the encrypted image using the proposed scheme is similar to a uniform distribution with a value close to 0.039 ($\frac{1}{256}$). To validate this result at the sub-matrix level, an entropy test is performed and this is described next.

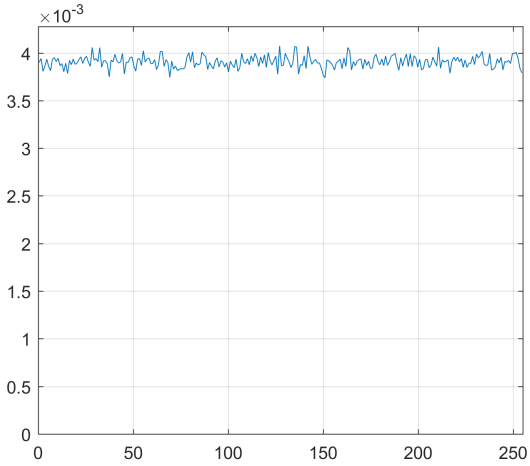
(a)



(b)



(c)



(d)

Figure 5: (a) Original Lena, (b) PDF of original Lena with size $512 \times 512 \times 3$, (c) Encrypted Lena, (d) PDF of encrypted Lena

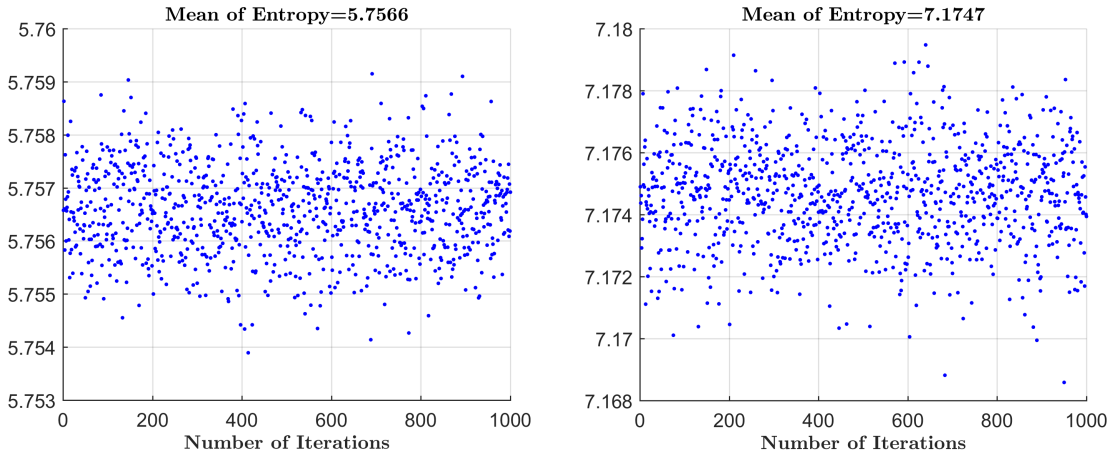
4.1.2 Entropy Test

The entropy of a sequence M is a value expressed in bits and permits to quantify the uncertainty level [53]. The entropy can be calculated as follows:

$$H(M) = -\sum_{i=1}^{n} p(m_i) \log_2 \frac{1}{p(m_i)} \tag{13}$$

where $p(m_i)$ represents the occurrence probability of the symbol m_i and n is the number of symbols. The proposed test measures the entropy at the sub-matrix level, where each sub-matrix has a length equal to h^2 bytes. This permits to quantify the uniformity at the sub-matrix level and not on the whole image. If the entropy of a sub-matrix is close to $\log_2(h^2)$, this indicates that it has a uniform distribution.

The entropy test for the original and encrypted Lena images at the sub-matrix level and with a random dynamic key for $h = 8$ is shown in Figure 6. According to the results, the entropy of the encrypted sub-matrices has a value close to the desired value of 6, in case of $h = 8$, and close to 7.17 for $h = 16$, which indicates that the uniform distribution is ensured. Hence, the redundancy at the sub-matrix level is eliminated.



(a) $h = 8$ (b) $h = 16$

Figure 6: The entropy test of the sub-matrices of the encrypted Lena image with a random dynamic key for $h = 8$ (a) and $h = 16$ (b).

4.1.3 Test Correlation Between Original and Cipher Images

Removing spatial redundancy will certainly result in an efficient cipher scheme [54, 55]. Therefore, ensuring a lower correlation (coefficient very close to zero) among encrypted image pixels indicates that the employed cipher has a high level of randomness. This proves that the employed cipher can resist statistical attacks. The correlation test is performed by taking randomly $N = 4,066$ pairs of adjacent pixels from the original and encrypted Lenna images. The correlation is quantified in

horizontal, vertical and diagonal directions and the coefficient correlation r_{xy} can be calculated as follows:

$$r_{xy} = \frac{\text{cov}(x, y)}{\sqrt{D(x) \times D(y)}} \tag{14}$$

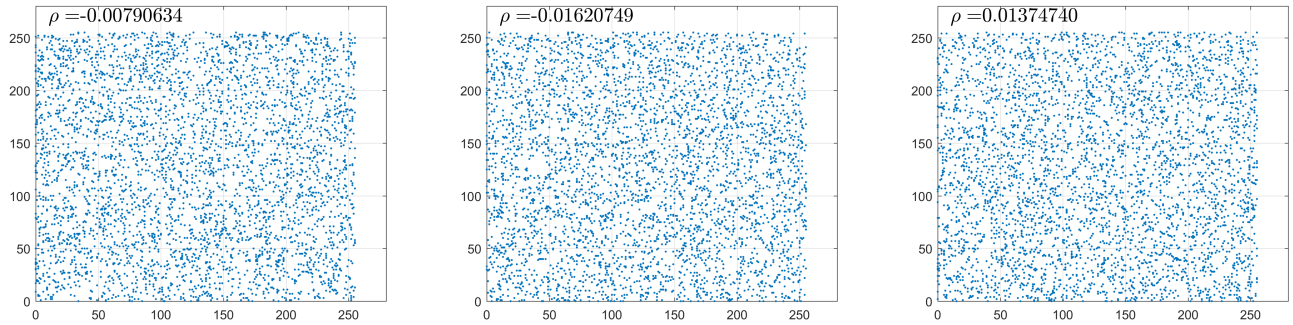
where

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

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

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

Obviously, the correlation between adjacent pixels in the plain image is high and its corresponding correlation coefficient is close to 1 [42, 32]. Figure 7 shows the correlation between adjacent pixels in the different directions for a random secret key. While, Figure 8 shows the variation of the coefficient correlation in the different directions for 1000 encrypted images, where each encrypted image uses a different secret key. According to these results, the ciphered images have a very low correlation coefficient (close to 0), which clearly shows that the proposed scheme drastically reduces the spatial redundancy.



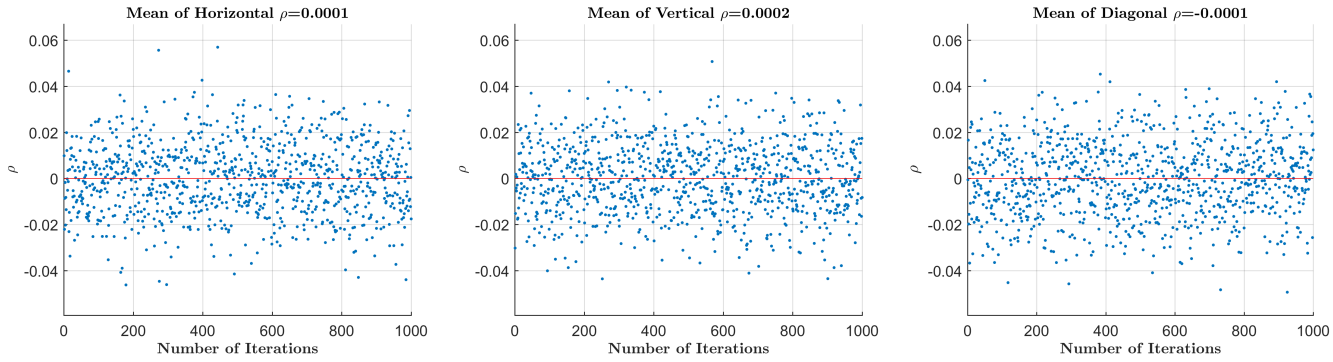
(a) (b) (c)

Figure 7: Adjacent pixels correlation for the ciphered Lena image with one random secret key: (a) horizontally, (b) vertically and (c) diagonally.

4.2 Visual Degradation

The degradation of the original image must be verified such that the visual content of the ciphered image must not be recognized. A hard visual degradation indicates a big difference between the original and the cipher images.

Two metrics are used to quantify the visual quality of encryption: the Peak Signal-to-Noise Ratio (PSNR) [56] and the Structural Similarity Index (SSIM) [57].



(a) (b) (c)

Figure 8: Coefficient Correlation of adjacent pixels in encrypted Lenna: (a) horizontally, (b) vertically and (c) diagonally versus 1000 random secret keys.

PSNR represents the cumulative squared error between the original and encrypted images. A low PSNR value, compared to a reference image, indicates the occurrence of a hard distortion. On the other hand, the SSIM index [58], permits the extraction of the structural information. The SSIM value ranges between 0 and 1. A low value of SSIM indicates a big difference in terms of structure between the original and encrypted images, while a higher value indicates the opposite.

We calculated the PSNR and SSIM values between the original and the encrypted Lena image for $1,000$ dynamic keys. The results are shown in Figure 9. As shown, the average PSNR value is 8.5894 dB, which is a low value. This confirms that the proposed cipher produces a large difference between the original and the encrypted images. Similarly, the maximum SSIM value for $1,000$ dynamic keys did not exceed 0.04, which confirms a high and adequate visual distortion. As such, sufficient visual degradation is achieved since no useful information or any pattern could be revealed from the encrypted image.

4.3 Difference Between Plain and Cipher Images

The difference between original and encrypted images at the bit level must reach a value very close to the ideal one (50%). We show the percent variation of the bit difference between the original and cipher Lena images for $1,000$ random dynamic keys in Figure 10. The results show that the percentage difference is always close to 50%. Hence, the proposed cipher satisfies the independence criteria.

4.4 Sensitivity Test

To avoid differential attacks, the relation between two encrypted images must be studied. Any slight difference in the plain image or in the key (usually one bit difference) must drastically affect the resultant encrypted image. As the percentage of change increases, the scheme will have better sensitivity. Two types of sensitivity require to be tested are: **Plain Sensitivity (PS)** and **Key Sensitivity (KS)**.

Plain-text Sensitivity: This type of sensitivity is not relevant to the proposed algorithm since different dynamic keys can be used for each input image. Accordingly, the scheme produces

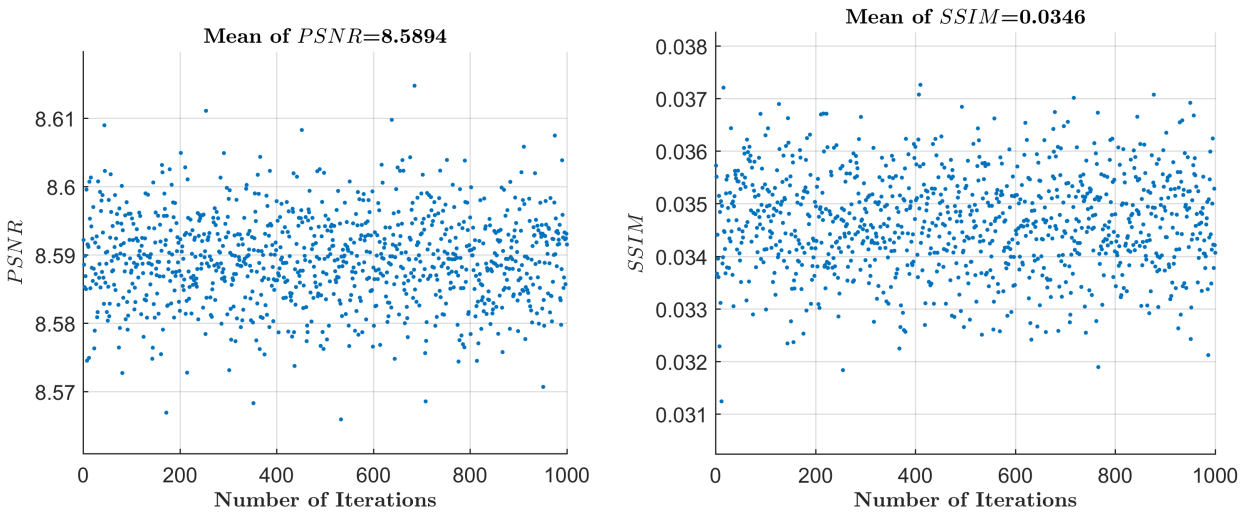


Figure 9: variation of the PSNR (a) and SSIM (b) between original and encrypted Lena images versus $1,000$ dynamic keys.

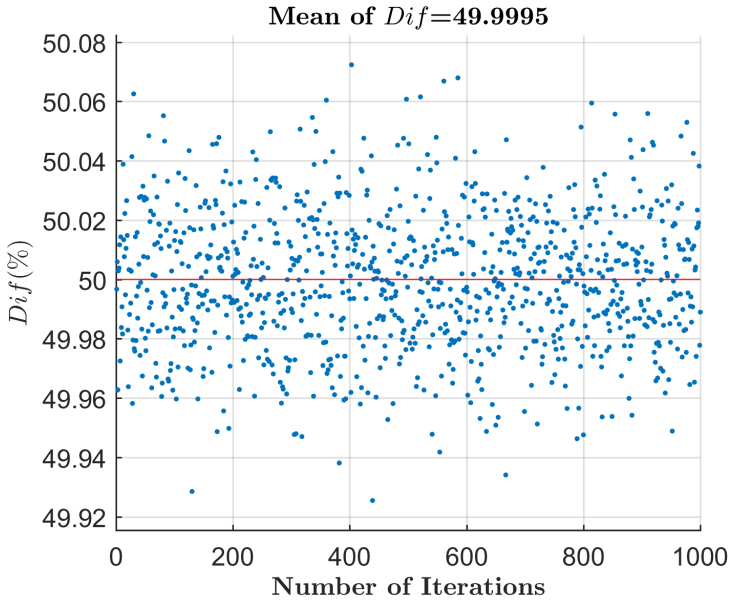


Figure 10: Percentage Difference between plain and ciphered Lena for 1,000 random dynamic keys.

totally different cipher images. Hence, the cipher successfully meets the avalanche effect due to the dynamic key approach and this will not provide any information about the secret key.

Concerning the **Key Sensitivity** test, it is one of the most important tests and permits to quantify the sensitivity against a slight change in the secret key. In fact, the proposed key derivation function is based on a secret key and an initial vector. To study the key sensitivity, two

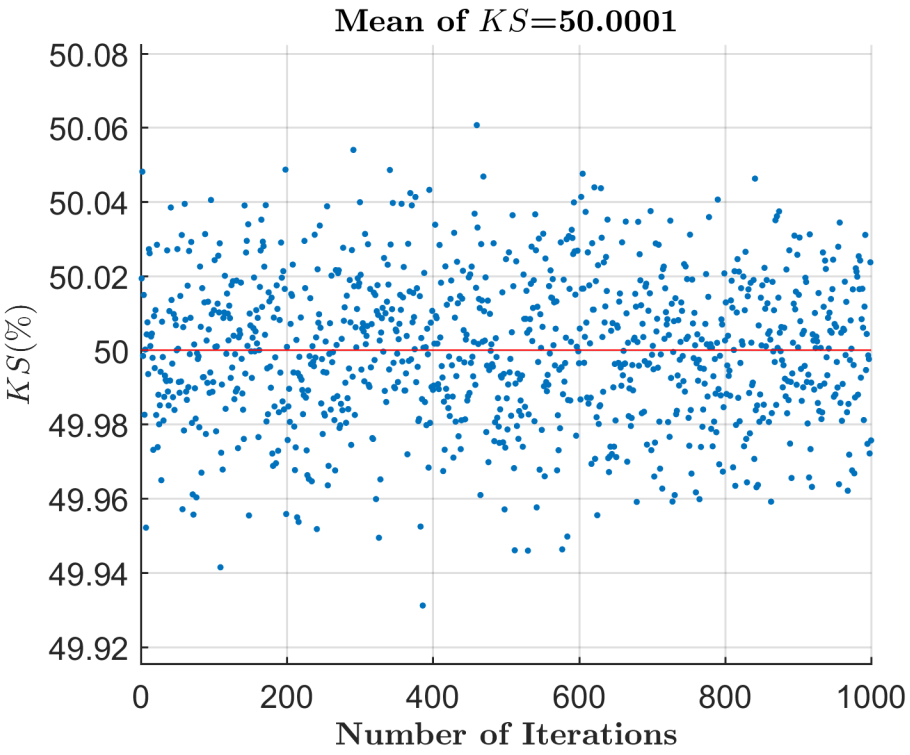


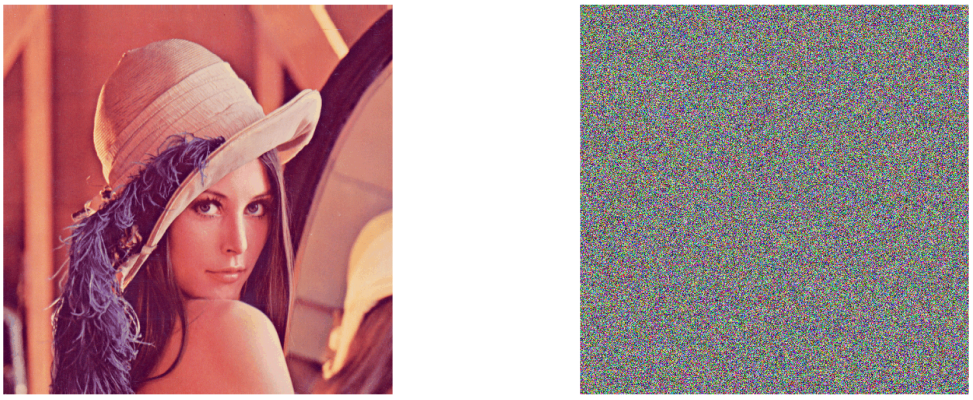
Figure 11: Key sensitivity against 1,000 random dynamic keys.

dynamic keys are used: DK_1 and DK_2 that only differ by one random bit. The two plain-images are encrypted separately and the Hamming distance of the corresponding cipher-images C_1 and C_2 is computed and illustrated in Figure 11 against $1,000$ random dynamic keys. The majority of values can be seen to be close to the optimal value of (50 %), indicating that the proposed encryption model is robust and has enough strength against any minor change in the dynamic key.

A decrypted Lena image is shown in Figure 12, which was decrypted using a dynamic key with a one-bit error. It is clear that this algorithm is highly sensitive to the dynamic key and any change in the latter will lead to a different decrypted image with no useful information. This test, in addition to the tests done previously, guarantees that a high sensitivity level and a high randomness degree are achieved with the proposed cipher scheme.

4.5 Cryptanalysis: Resistance against well-known types of attacks

The proposed scheme was tested using a set of security tests repeated 1,000 times to prove its immunity against attacks. Next, we present a brief cryptanalysis discussion to demonstrate the security of the cipher and its ability to resist the existing and modern confidentiality attacks. Different statistical tests were performed and they proved that the proposed cipher satisfies the uniformity and independence properties. Hence, a high randomness level is achieved in a dynamic manner, which makes the proposed cipher immune against statistical attacks.



(a) (b)

Figure 12: Decrypted Lena image with its corresponding correct dynamic (a) and with one bit error in the dynamic key used (b).

Table 4: Statistical results of the listed tests by using original Lena image with the proposed cipher scheme and for 1,000 random keys.

Proposed Scheme				
	Min	Mean	Max	Std
Dif	49.9254	49.9995	50.0724	0.0229
KS	49.9311	50.0001	50.0607	0.0196
$H-E$ (h=8)	5.7539	5.7566	5.7591	0.0008
$\rho-h$	-0.0462	0.0001	0.0569	0.0154
$\rho-v$	-0.0617	-0.0005	0.0529	0.0157
$\rho-d$	-0.0503	-0.0001	0.0455	0.0158
PSNR	8.5659	8.5894	8.6147	0.0065
SSIM	0.0312	0.0346	0.0373	0.0009

Moreover, we performed sensitivity tests on the proposed cipher scheme. The results proved that the cipher exhibits a high level of sensitivity, which makes it immune against key-related attacks.

On the other hand, the proposed cipher can resist brute force attacks since the secret key has a flexible size of either 128, 196, 256 or 512 bits, and the Nonce has a size of 512 bits. Moreover, the size of the dynamic key is also 512 bits. Therefore, the size of the static secret key, dynamic key and Nonce are sufficient to make the brute force attack unfeasible.

More importantly, the dynamic key-dependent structure plays a significant role in making the proposed cipher scheme immune against the current and future powerful attacks such as chosen/known plain/cipher text attacks.

In conclusion, the security level of the proposed cipher scheme is confirmed. In the following

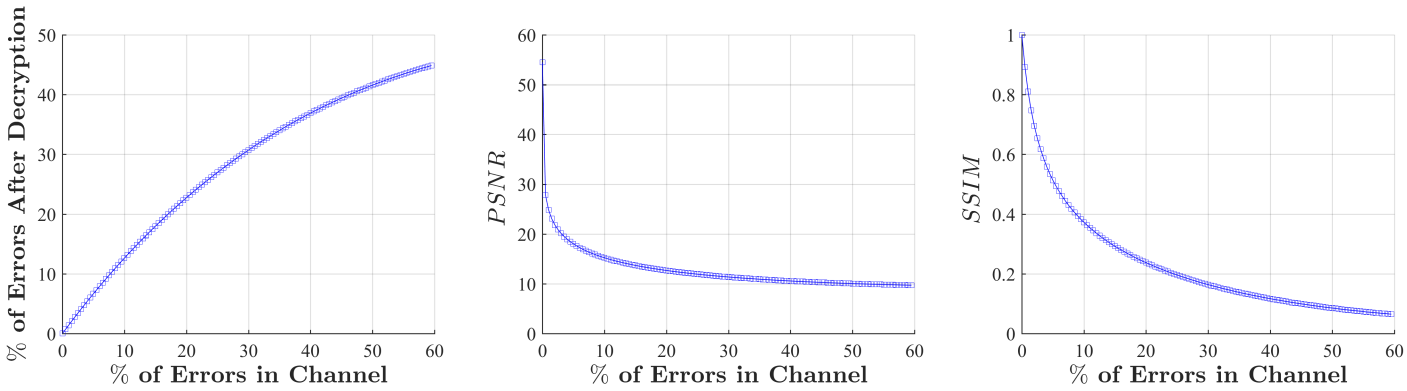
section, we verify the efficiency of the proposed cipher as a good cipher candidate.

5 Performance Analysis

In this section, the performance of the proposed cipher scheme is analyzed to validate its effectiveness. Several performance experiments were done such as studying the effect of the error propagation and measuring the execution time. The results indicate clearly the efficiency of the proposed cipher.

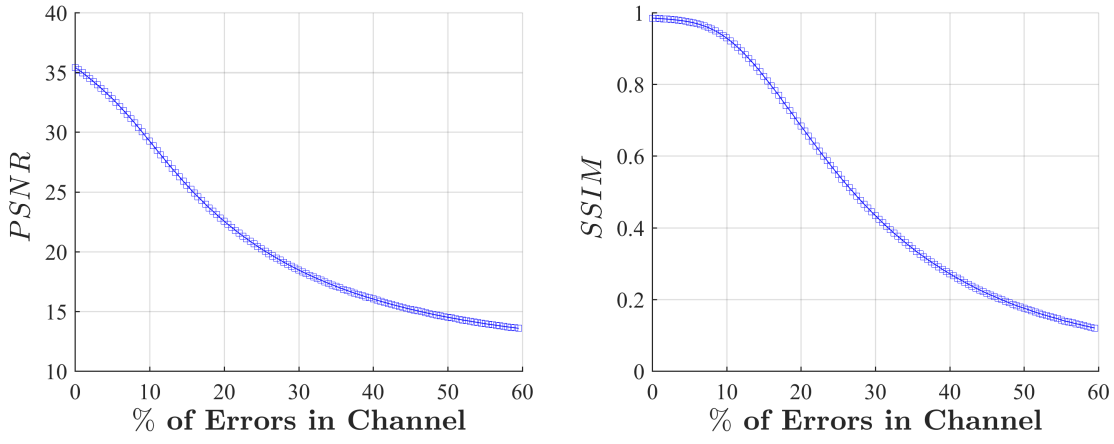
5.1 Error Propagation

Communication channels typically suffer from noise (interference) or from severe fading. When an image is transmitted over such a channel, an error in one or more bits may occur; that is a '0' may flip to '1' and vise versa. It is very important that such an error does not propagate all over the image as in [40], otherwise, the decrypted image will be severely degraded. In fact, the lower the error propagation, the more effective and practical the cipher scheme will be. In the proposed cipher, the error will **only** affect the corresponding sub-matrices (x_i, y_i). More precisely, the effect of a bit error introduces only a specific sub-matrix error at the same byte position of the error in the decrypted image. Hence, the error in the proposed scheme is not propagated randomly to both sub-matrices as in [42, 32]. In order to quantify the visual degradation, we use again the two well-known parameters, PSNR and SSIM. The variation of SSIM and PSNR versus the percentage of errors are shown in Figure 13. The proposed solution shows a linear difference, and the variations of SSIM and PSNR show that the scheme is immune to a highly erroneous channel. This conclusion is confirmed by showing the decrypted image corresponding to a large number of errors (Figure 15). In addition, when an image is decrypted with errors, the tests showed that applying a filter to clean the image can solve the noise problem. In our simulations, a random noise was added (up to 20%) and a median filter was capable of removing the added noise. Note that the size of the median filter should not exceed the size of the sub-matrix, which is $h \times h$. This is shown in Figure 14 and 16 and it is clear that the filtered images are flawless with no errors appearing visually.



(a) (b) (c)

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



(a) (b)

Figure 14: Variation of PSNR and (b) SSIM versus the percentage of errors in channel for the proposed approach after applying a median filter.

(a) 1% (b) 5% (c) 9% (d) 15% (e) 20%

Figure 15: Decrypted images in function of the percentage of errors in channel for the proposed approach.

(a) 1% (b) 5% (c) 9% (d) 15% (e) 20%

Figure 16: The result of applying a median filter to Lena decrypted images with different percentages of errors.

5.2 Execution Time

An efficient cipher scheme must reach low computational complexity to ensure low latency and consequently low resources and energy consumption.

In order to show the efficiency of the proposed cipher scheme for IoT devices, a comparison with AES OpenSSL was performed. OpenSSL is commonly used and considered as one of the most important and efficient cryptographic libraries that can provide a robust, commercial-grade, and full-featured toolkit for the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols. On the other hand, the proposed approach has been implemented in Matlab, C and Java. Therefore, to show a better system performance, the "C" implementation is selected to be compared with the AES OpenSSL implementation on two very common IoT hardware **Raspberry Pi Zero W (wireless) and Raspberry Pi 2**. The "Raspberry Pi Zero" has a Broadcom BCM2836 SoC with a 1 GHz single-core ARM1176JZF-S. The "Raspberry Pi 2" has a Broadcom BCM2836 SoC with a 900 MHz 32-bit quad-core ARM Cortex-A7 processor.

We record the average time (for $1,000$ iterations) to encrypt the plain Lena image of size $512 \times 512 \times 3$. For the Raspberry Pi Zero W (called RPi Zero later) and for the Rasberry Pi 2 (called RPi 2 later) the optimal size of blocks is 32.

The encryption and decryption times of our one round approach and of AES (128 bits) are presented in Table 5. The time ratio shows that the proposed approach is 7% faster on the RPi Zero for the encryption process and 21% faster in the case of decryption. On the RPi 2, the gain in encryption is 29% and 33% in decryption. It should be noted that the proposed algorithm is completely written in C while OpenSSL uses assembly optimization [59]. Despite this optimization, an important reduction in encryption and decryption times is achieved. This primary result indicates clearly that, by optimizing the proposed approach and by employing assembly optimization, a better reduction in latency and energy consumption can be achieved. In fact, this is our future perspective. Moreover, in Figure 17, we show the variation of the execution time for the encryption and decryption algorithms versus h is presented. The experiments were performed on two different hardware devices, RPi W and RPi 2. The results indicate clearly that increasing h reduces the required latency at the expense of additional memory overhead. Therefore, the choice of h depends on the latency and hardware requirements; the proposed approach provides the user with the opportunity to choose the value of h depending on the application requirements. In fact, when devices have high memory capacity, a high value of h can be chosen (16 or 32). While, for low-cost devices that have limited memory capacity, a low value of h must be chosen (4 or 8). In this paper, extensive security tests are performed with h equals to 8, which represents a good balance between computational complexity and memory consumption.

Table 5: The mean encryption time (in seconds) of AES and the proposed cipher approach for $512 \times 512 \times 3$ Lena image and for $1,000$ iterations.

Hardware	Algorithm	Encryption Time (s)	Decryption Time (s)
RPi Zero W	One round	0.0388	0.0340
	AES	0.0418	0.0432
RPi 2	One round	0.0260	0.0251
	AES	0.0367	0.0374

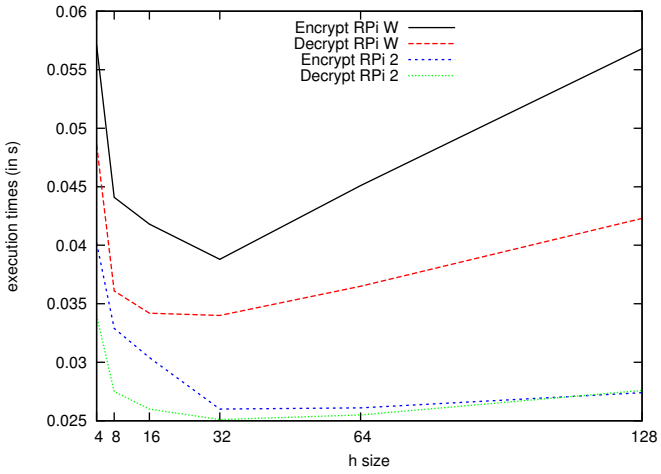


Figure 17: Execution times on RPi W and RPi 2 versus h.

6 Conclusions and Future Work

A single-round, flexible, dynamic, key-dependent lightweight cipher scheme targeted for multimedia IoT has been presented. The scheme has been shown to be efficient and secure, with fast execution time. The scheme is based on a dynamic structure in contrast to standard techniques. This will provide better robustness against different powerful attacks because of the different substitution and permutation primitives in addition to the two dynamic pseudo-random matrices that are generated in a dynamic manner. Moreover, the proposed substitution and diffusion primitives ensure the desirable cryptographic performance in an efficient manner and simple hardware implementation. The proposed cipher scheme requires only one iteration and its corresponding round function consists of simple operations, which addresses the limitations of IoT multimedia devices. An extensive security analysis revealed that the proposed approach is strong enough against different kinds of attacks. Finally, the results clearly showed that the scheme outperforms the optimized AES implementation of OpenSSL, which indicates that the approach is more suitable for delay-sensitive multimedia applications.

As a future work, the cipher will be further optimized via an assembly optimization to achieve a better reduction in latency and required resources. Additionally, the proposed cipher scheme should be adapted to be a post crypto-compression scheme to ensure the format compliance. A possible implementation can be realized for compressed images such as JPEG and JPEG2000.

Acknowledgement

This paper is partially supported with funds from the 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] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The Internet of Things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[2] Harald Sundmaeker, Patrick Guillemin, Peter Friess, and Sylvie Woelfflé. Vision and Challenges for Realising the Internet of Things. *Cluster of European Research Projects on the Internet of Things, European Commision*, 3(3):34–36, 2010.

[3] Antonio J Jara, Miguel A Zamora-Izquierdo, and Antonio F Skarmeta. Interconnection framework for mHealth and remote monitoring based on the Internet of Things. *IEEE Journal on Selected Areas in Communications*, 31(9):47–65, 2013.

[4] Victor Chang. Data analytics and visualization for inspecting cancers and genes. *Multimedia Tools and Applications*, pages 1–15, 2017.

[5] Daniel Adrianto and Fuchun Joseph Lin. Analysis of security protocols and corresponding cipher suites in etsi m2m standards. In *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, pages 777–782. IEEE, 2015.

[6] Jorge Granjal, Edmundo Monteiro, and Jorge Sá Silva. Security for the internet of things: a survey of existing protocols and open research issues. *IEEE Communications Surveys & Tutorials*, 17(3):1294–1312.

[7] Ruhul Amin, SK Hafizul Islam, Pandi Vijayakumar, Muhammad Khurram Khan, and Victor Chang. A robust and efficient bilinear pairing based mutual authentication and session key verification over insecure communication. *Multimedia Tools and Applications*, pages 1–26, 2017.

[8] Sheeraz A Alvi, Bilal Afzal, Ghalib A Shah, Luigi Atzori, and Waqar Mahmood. Internet of multimedia things: Vision and challenges. *Ad Hoc Networks*, 33:87–111, 2015.

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

[10] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer Verlag, Berlin, Heidelberg, New York, 2002.

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

[12] Saurabh Singh, Pradip Kumar Sharma, Seo Yeon Moon, and Jong Hyuk Park. Advanced lightweight encryption algorithms for iot devices: survey, challenges and solutions. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–18, 2017.

[13] Amir Moradi and Axel Poschmann. Pushing the limits: a very compact and a threshold implementation of aes. In *Eurocrypt*, volume 6632, pages 69–88. Springer, 2011.

[14] Dag Arne Osvik, Joppe W Bos, Deian Stefan, and David Canright. Fast software aes encryption. In *Fast Software Encryption: 17th International Workshop, FSE 2010, Seoul, Korea,*

Februara 7-10, 2010 Revised Selected Papers, volume 6147, page 75. Springer Science & Business Media, 2010.

[15] Benjamin Buhrow, Paul Riemer, Mike Shea, Barry Gilbert, and Erik Daniel. Block cipher speed and energy efficiency records on the msp430: System design trade-offs for 16-bit embedded applications. In *International Conference on Cryptology and Information Security in Latin America*, pages 104–123. Springer, 2014.

[16] Stefan Tillich and Johann Großschädl. Instruction set extensions for efficient aes implementation on 32-bit processors. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 270–284. Springer, 2006.

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

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

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

[20] Ahmed Khattab, Zahra Jeddi, Esmaeil Amini, and Magdy Bayoumi. *RFID Security: A Lightweight Paradigm*. Springer, 2016.

[21] Hyubgun Lee, Kyounghwa Lee, and Yongtae Shin. Aes implementation and performance evaluation on 8-bit microcontrollers. *CoRR*, abs/0911.0482, 2009.

[22] C. Evans-Pughe. Bzzzz zzz [ZigBee wireless standard]. *IEE Review*, 49(3):28–31, 2003.

[23] Shahid Raza, Adriaan Slabbert, Thiemo Voigt, and Krister Landernäs. Security considerations for the wireless hart protocol. In *Proceedings of the 14th IEEE international conference on Emerging technologies & factory automation*, ETFA'09, pages 242–249, Piscataway, NJ, USA, 2009. IEEE Press.

[24] R Nithya and Deepa S Kumar. Where aes is for internet, simon could be for iot. *Procedia Technology*, 25:302–309, 2016.

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

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

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

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

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

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

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

[32] 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.

[33] Carlisle Adams and Stafford Tavares. Good s-boxes are easy to find. In *Conference on the Theory and Application of Cryptology*, pages 612–615. Springer, 1989.

[34] Liam Keliher and Henk Meijery. A new substitution-permutation network cipher using key-dependent s-boxes.

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

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

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

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

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

[40] Safwan El Assad and Mousa Farajallah. A new chaos-based image encryption system. *Signal Processing: Image Communication*, 41:144–157, 2016.

[41] Hassan Noura, Lama Sleem, and Raphaël Couturier. A revision of a new chaos-based image encryption system: Weaknesses and limitations. *CoRR*, abs/1701.08371, 2017.

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

[43] Salim Muhsin Wadi and Nasharuddin Zainal. High definition image encryption algorithm based on aes modification. *Wireless personal communications*, 79(2):811–829, 2014.

[44] Peng Zhang, Yixin Jiang, Chuang Lin, Yanfei Fan, and Xuemin Shen. P-coding: secure network coding against eavesdropping attacks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[45] LN Pradeep and Aniruddha Bhattacharjya. Random key and key dependent s-box generation for aes cipher to overcome known attacks. In *International Symposium on Security in Computing and Communication*, pages 63–69. Springer, 2013.

[46] Victor S Miller. Use of elliptic curves in cryptography. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 417–426. Springer, 1985.

[47] Christof Paar and Jan Pelzl. *Understanding cryptography: a textbook for students and practitioners*. Springer Science & Business Media, 2009.

[48] RL Rivest. The rc4 encryption algorithm. rsa data sec. *Inc.(March 1998)*, 1992.

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

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

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

[52] 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.

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

[54] 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.

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

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

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

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

[59] Daniel J Bernstein, Bernard Van Gastel, Wesley Janssen, Tanja Lange, Peter Schwabe, and Sjaak Smetsers. Tweetnacl: A crypto library in 100 tweets. In *International Conference on Cryptology and Information Security in Latin America*, pages 64–83. Springer, 2014.