Efficient & secure cipher scheme with dynamic key-dependent mode of operation

Hassan Noura, Ali Chehab, Raphael Couturier

Efficient & Secure Cipher Scheme with Dynamic Key-Dependent Mode of Operation

Hassan N. Noura[a], Ali Chehab[a], Raphaël Couturier[b]

[a] *Electrical and Computer Engineering*
American University of Beirut (AUB)
Beirut, Lebanon
[b] *Univ. Bourgogne Franche-Comté (UBFC),*
FEMTO-ST Institute, CNRS, Belfort, France

Abstract

Security attacks are constantly on the rise leading to drastic consequences. Several security services are required more than ever to prevent both passive and active attacks such as Data Confidentiality (DC). A DC security service is typically based on a strong symmetric cipher algorithm. However, some of today's applications, such as real-time applications and those running on constrained devices, require efficient lightweight cipher schemes that can achieve a good balance between the security level and system performance. Recently, a set of lightweight cryptographic algorithms has been proposed to that end, which is based on a dynamic key approach. The dynamic structure enables the reduction of the number of rounds to the minimum possible value of just one or two rounds, which minimizes the computational overhead without degrading the security level. This paper follows the dynamic key-dependent cipher logic and proposes a new flexible lightweight technique with or without the reliance on the chaining mode of operation. Furthermore, the dynamic key changes for each input message, which leads to different cipher primitives such as substitution and permutation tables, in addition to round keys. Also, the proposed mode of operation is based on the dynamic key approach whereby blocks are selected and mixed according to a dynamic permutation table. Accordingly, different plaintext messages are encrypted differently while preserving the avalanche effect. Finally, we conduct security and performance analysis to validate the efficiency and robustness of the proposed cipher scheme as compared to traditional ciphers and to the recently proposed dynamic key-dependent ciphers.

Keywords: Lightweight flexible key-dependent cipher scheme, dynamic operation mode, security and performance analysis.

1. Introduction

Emerging systems are more prone to security threats compared to traditional networks. These systems are facing dangerous security and privacy issues through different attack types that target various security services such as confidentiality (data confidentiality and privacy), integrity (device system integrity), availability (data and system), and authentication (device/user and data origin authentication). Therefore, two types of security solutions can be used to ensure the required security services, cryptographic and/or non-cryptographic solutions. Data confidentiality, data integrity, and source authentication are achieved mainly via cryptographic algorithms.

1.1. Problem Formulation

Obviously, applications that transmit or store sensitive information must be well protected. Confidentiality is among the most dangerous passive threats mainly including eavesdropping and traffic analysis, where adversaries aim to extract the message itself, or any useful information from communicated data.

However, the existing confidentiality solutions may not be suitable for delay-sensitive systems [13]. Moreover, they are not practical within constrained devices with limited battery lifetime and limited computational power. Also, various emerging applications have stringent QoS requirements. Relying on traditional security in these scenarios may cause an overhead in terms of latency and resources, such as the case with AES [2] (Advanced Encryption Standard), which requires a large number of rounds resulting in a negative impact on the corresponding system performance. Hence the need for a cryptographic algorithm concept with low latency and required resources is a must. On the other hand, relying on chaotic cryptographic algorithms as an alternative way is not possible since chaotic cryptography requires floating-point computations, conversion of operations, in addition to finite periodicity (1D-chaotic map) and complicated hardware implementation.

Accordingly, recent works shifted towards the design of a new cryptography class, known as "Lightweight" [13, 22].

A set of lightweight cryptographic algorithms that rely on the dynamic key approach were presented by [14]-[19]. These approaches require a low number of rounds leading to the reduction of the computational complexity whilst preserving a high-security level.

1.2. Motivation and Contributions

This paper addresses the problem of securing the exchange of information in order to overcome confidentiality issues with low overhead and a small latency. We present a new efficient, flexible, lightweight yet secure cipher algorithm that adopts the dynamic key-dependent cipher concept. The proposed cipher uses a single round, which requires few operations and it ensures a good cryptographic performance. To do that, a dynamic key is generated (or updated) for each input message, which can be audio, image, video message, etc. This dynamic key is produced as a function of a secret key and a secret dynamic Nonce. The dynamic key approach makes the cryptographic primitives variable and unknown to any given attacker. As a consequence, this introduces a high complexity for attackers. Moreover, the presented technique of [16] is used to generate dynamic key-dependent substitution and permutation tables based on the Key Setup Algorithm (KSA) of RC4 and the modified KSA. In addition, these techniques ensure that the produced substitution and permutation tables can reach the desirable cryptographic performance in a dynamic key-dependent manner.

The main goal of this work is to propose a design for an alternative cipher that has a different structure than AES, and which addresses any possible weakness when used with constrained devices or when dealing with applications that require a real-time response. As such, in addition to security, the proposed scheme should exhibit the minimum possible delay and computational complexity. Accordingly, we are proposing a cipher that uses the dynamic key-dependent approach instead of the static one such as AES; this can achieve a high security level with minimum operations. With the dynamic approach, all cipher primitives such as substitution and permutation tables are unknown for the attackers and vary for each input message (or for a threshold data size in case of small messages), which makes the attack process extremely difficult. Moreover, the relation between the dynamic keys is independent and non-invertible compared to the secret key. For example, this was the main idea during the design of SHA3, which avoids the cryptographic concept of SHA1 and SHA2. In addition, we try to reduce the trade-off between the security level and performance requirements in the proposed scheme to respond better to tiny devices and real time requirements.

Our proposed cipher solution has several contributions in terms of security and system performance, and they are listed next.

System performance

- **Efficiency**: Recent lightweight symmetric ciphers such as the Hummingbird2 require at least 4 rounds as indicated in [16]. On the other hand, chaotic symmetric ciphers do require floating-point calculations and conversion operations and they are based on the multi-round structure. Our proposed cipher solution requires one simple round function to encrypt 2 blocks at a time as compared to [7]. This requires two complete rounds of substitution-diffusion operations in addition to chaining. The proposed cipher requires fewer number of operations and avoids diffusion operations to reduce the required computational complexity and required resources. Furthermore, [19] can be considered an enhancement of [7] since only one substitution or diffusion operation is required for each round (two rounds). However, this scheme uses a matrix diffusion operation, which allows the cipher scheme to achieve better efficiency. The recent cipher scheme of [16] was compared to related work to validate its effectiveness. The proposed cipher scheme requires half the computational complexity ($O(\frac{nb}{2})$) compared to the recent lightweight one round cipher scheme of [16] ($O(nb)$), where nb represents the number of blocks in one input message and it is equal to $\lceil \frac{len}{N} \rceil$. Note that len represents the length of the input message after being reshaped to a vector form, and m represents the block size (number of bytes in each message block).

  Hence, the proposed cipher requires half the computational complexity and resources compared to [16] with a similar security level. Moreover, the proposed cipher scheme without chaining (just encryption and decryption) can be realized in parallel, while the encryption algorithm of the chaining variant cannot be parallelized. In summary, the proposed cipher scheme strikes a very good balance between security and performance, which is detailed in the following.

- **Flexibility:** The proposed cipher deals with a block that has a flexible length (N bytes) that can be adjusted according to the device constraints and application.

- **Simple hardware and software implementations:** "Exclusive OR" logical operations, with look-up substitution and permutation operations make the corresponding HW (HardWare) and SW (SoftWare) implementations of the proposed cipher simple and efficient.

- **Error Tolerance**: The proposed cipher scheme shows a better resistance against channel errors compared to [7, 19] (see Section 6.1). The effect of any error in any byte of any encrypted block will affect only two bytes on both mixed blocks, and at the same block byte position. The error(s) will not affect the whole block due to the avalanche effect being achieved differently with a new dynamic key for each input message. However, the proposed cipher scheme with the chaining operation mode doubles the error effect compared to the unchained one, but the effect of the corrupted bytes is only limited to the bytes of the next blocks. Overall, the proposed cipher scheme exhibits a lower error propagation compared to standard block ciphers such as AES, which makes it useful for any error detection-correction scheme.

All these performance contributions lead to lower computational delay and resource requirements for the proposed cipher in addition to simplifying its practical implementation (HW/SW). Next, we will present the different security performance of the proposed cipher.

Security Performance

- **Dynamic Key-Dependence Approach:** The proposed cipher is based on dynamic key-dependent cryptographic primitives (substitution and permutation tables in addition to round keys) which are changed in a dynamic pseudo-random manner for each input message. The advantage is that the proposed approach uses the session key to produce a set of dynamic keys. Consequently, statistical or implementation attacks will be very challenging since different encrypted keys are being used with different physical and logical properties [19, 7, 27, 23]. Note that we adopt two different substitution tables instead of one to make the proposed cipher more secure.

- **Dynamic Operation Mode**: Traditional block ciphers use static operation mode with a sequential order of blocks for encryption and decryption. The proposed cipher scheme is based on the dynamic pseudo-random block selection operation via a dynamic permutation table. This makes the relation between the encrypted blocks even more random and complicated. Consequently, this reinforces its immunity against analytic attacks since the encryption/decryption sequential order becomes dynamic and variable for each input message. Moreover, this operation exhibit lower latency and resources overhead towards preserving the system performance advantages.

The security level and performance of the proposed cipher scheme were validated through a set of security and performance tests.

The proposed solution can be also extended to provide an authenticated encryption (AE) operation mode such as CCM [6] and GCM [5], which require two passes; the first one for the authentication and the second one for the confidentiality. On the other hand, other AE operation modes such as IAPM [11] and OCB [24] require only a single pass for authentication and encryption. All these listed AE operation modes can ensure data integrity, source authentication in addition to data confidentiality. However, within their single or double pass(es), they rely on a block cipher that requires multiple rounds r such as AES (10, 12 and 14 rounds for 128, 192 and 256 bits key length, respectively). Similarly, [10] presents an authenticated encryption mode that requires a block cipher such as AES. When compared against the optimized version of AES, the proposed cipher requires half the delay and the required resources. Note that in this paper, we focus on a cipher scheme for data confidentiality with dynamic operation modes such as ECB, CBC, and CTR, while relying on a single round with a minimum number of operations. As a future work, the scheme will be extended to provide an authentication operation mode that can achieve a significant gain (one round) in a similar manner to data confidentiality.

1.3. Organization

The rest of this paper is organized as follows. Section 2 presents the proposed key derivation function in addition to the employed techniques to construct different cipher primitives. In Section 3, the proposed cipher scheme with and without chaining is described in detail as well as the concept of the dynamic mode of operation. Next, we analyze and assess the security level of the proposed cipher scheme in Section 4. Then, in Section 5, we prove the immunity of the proposed algorithm against different kinds of existing attacks. In Section 6, the effectiveness of the proposed scheme is evaluated and confirmed. Finally, in Section 7, the conclusions are drawn along with directions for future work.

2. Proposed Key Derivation Function

All the notations used in the following description are given in Table 1, and the proposed dynamic key generation technique steps are shown in Figure 1. A shared secret session Key (SK) between two legal entities, as well as a secret Nonce are considered as input for each new session. The session key can then be renewed for each new session or depending on the system's configuration. However, the key management among legal entities (users or devices) is not the focus of this paper. For more details about the possible key management, readers can refer to [25].

For each new input message, a dynamic key (DK) is produced by hashing the secret key SK with a Nonce that can be generated in a synchronous manner between legal entities.

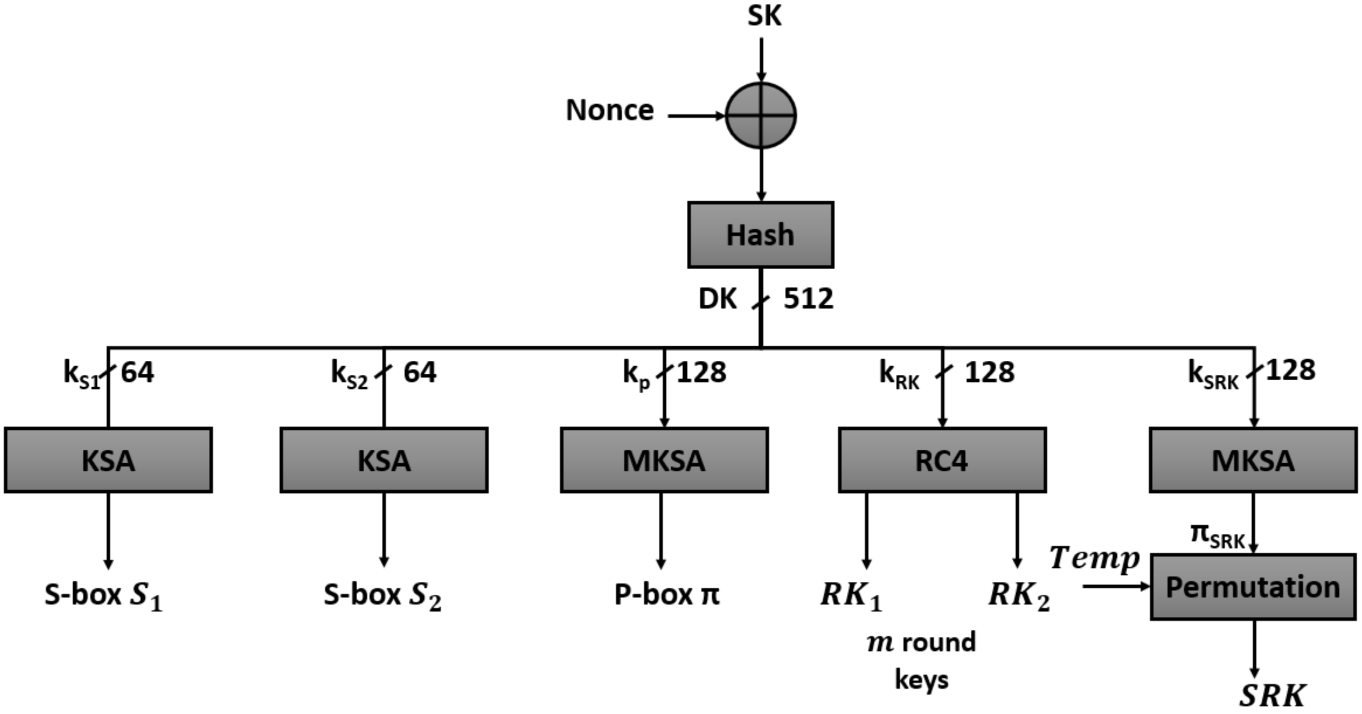


Figure 1: Proposed dynamic key derivation Function and construction cipher primitives

Since the hash function should be a secure keyed cryptographic hash function, in this paper, HMAC [8] with SHA-512 is employed since it can provide better resistance against collision and other desirable cryptographic performance. The output of this step is a dynamic key, which has a length of 64 bytes.

$$DK = HMAC_{SK}(Nonce) \tag{1}$$

Note that the Nonce changes frequently for each input message and thus, a different dynamic key is produced for each input message compared to the previous or next message. Then, the dynamic key is divided into five sub-keys $DK = \{k_{S1},\ k_{S2},\ k_P,\ k_{RK},\ k_{SRK}\}$ where each one has a length of 128 bits (16 bytes), except k_{S1} and k_{S2} that have a length equal to 64 bits (8 bytes). These dynamic sub-keys are used for different purposes as they are described below.

- **Substitution sub-key** k_{S1}: it consists of the most significant 8 bytes of DK and is used to construct the first substitution table S_1 by using the key setup algorithm of RC4 as described in [16]. Note that the substitution operation is done at the byte level and that the elements in the table S_1 have values between 0 and 255.

- **Substitution sub-key** k_{S2}: it consists of the next most significant 8 bytes of DK and is used to construct the second substitution table S_2, as described previously.

- **Permutation sub-key** k_P: it represents the next most significant 16 bytes of DK and is used to construct a flexible permutation table π of length nb by using the modified

Table 1: Table of Symbols used

Symbol	Definition
SK	A shared secret Session Key
$Nonce$	A dynamic Nonce and it is changed for each input message
DK	A Dynamic Key and it is updated for each input message
k_S	Substitution sub-Key
S_i	i^{th} dynamic substitution table (256 elements)
k_p	Permutation sub-Key
π	Dynamic permutation table
k_{RK}	Seed for a stream cipher to produce RK
$RK1$	The first set of m dynamic round keys
$RK2$	The second set of m dynamic round keys
k_{SRK}	A selection Round Key and it is used to produce SRK
SRK	a selection round key table and it is used to decide which round keys (from $RK1$ and $RK2$) are used for each input block encryption
len	length of input message after reshaped to a table form.
nb	Number of blocks in one input message and it is equal to $\lceil \frac{len}{N} \rceil$
N	Number of bytes in one block message
m	Number of the different round keys generated
M	The original message
m_i	The i^{th} original plain block
C	The encrypted message
c_i	The i^{th} encrypted block
τ	For applications with small-size messages, the update of the dynamic key and cipher primitives are done after a certain configured data threshold length (τ)
l	represents the maximum size of small-size messages
δ	the size of a set of small-size messages and it is equal to $\lfloor \frac{\tau}{l} \rfloor$.

key setup algorithm of RC4, which was presented in [16]. The values of the elements in the permutation table π range from 1 to nb.

- **Round Key generation sub-key** k_{RK}: this consists of the next most significant 16 bytes. This step can be realized in two different manners:

1. **PRNG variant:** k_{RK} is divided into two equal parts (k_{RK1} and k_{RK2}) and the length of each part is 8 bytes and each one is used as a seed for a stream cipher in order to generate, for each iteration, two pseudo-random blocks (RK_1 and RK_2).

2. **Pre-generate variant:**k_{RK} is used as a seed for an efficient stream cipher that will be iterated to produce $2 \times m \times N$ bytes keystream, which is divided into two equal parts ($m \times N$). As such, two sets of m round keys $RK1 = \{RK_{1,1},\ \ldots, RK_{1,m}\}$ and $RK2 = \{RK_{2,1},\ \ \ldots, RK_{2,m}\}$ are generated. Each $RK_{1,j}$ or $RK_{2,j}$ for $j = 1,\ 2,\ \ldots,\ m$ has N bytes. These round keys can be generated using any stream cipher. In this paper, the RC4 stream cipher [21] is used, with k_{RK} as a seed to produce $2 \times m \times N$ bytes key-stream. The first $m \times N$ is used to form $RK1$, which consists of m round keys and each one having N bytes key-stream. Similarly, the next $m \times N$ is used to form $RK2$. In addition, m is a configured parameter and it can be selected according to the memory constraints. The importance of this variant compared to the previous one is that the encryption/decryption process can be realized in parallel (with the proposed dynamic Electronic-Code-Book (ECB) if parallel computation is possible). However, this variant is suitable for a small message with only few blocks $\leq nb$ to maintain security such that different pseudo-random blocks (RK_1 and RK_2) are used for each input block and consequently, to avoid any randomness issue.

- **Dynamic Round Key selection** k_{SRK}: this represents the least significant 16 bytes of DK and is used to construct a Selection Round Key (SRK) table of length $\lceil nb/2 \rceil$ and its corresponding values range from 1 to m. A possible technique to generate SRK is by constructing an initial table with m elements $Temp[i] = i$ for $i = 1,\ 2,\ \ldots,\ m$. Then, this table is repeated for $\lceil \frac{nb}{m} \rceil$ (copies of $Temp$) to have a length equals or greater than nb and its values are preserved ranging from 1 to m. In fact, $Temp$ after repeated might it have a length greater than nb, so a truncation operation is required to have only nb elements. Then, the modified KSA of RC4 [16] is iterated with k_{SRK} to construct a permutation table π_{SRK} with $\frac{nb}{2}$ elements. SRK is obtained by permuting the repeated table $Temp$ using the permutation table π_{SRK}. The permuted $Temp$ is called $SRK = Temp(\pi_{SRK})$ table. SRK is used to select the round key, which is used to mix with a couple of input blocks in a non-linear manner.

In this scheme, any bit difference in the secret key or Nonce will provide a different dynamic key. Therefore, the proposed cipher approach ensures a high key sensitivity since all cipher primitives are related to the dynamic key.

At the legitimate destination, the inverse substitution S^{-1} table is required. For this purpose, the original substitution S should be produced first in a similar manner to the source side. Then, the inverse substitution table can be obtained using S according to the following equation:

$$S^{-1}[S[i]] = i \text{ for } i = 0,\ 1,\ 2,\ \ldots,\ 255 \tag{2}$$

2.1. Adaptation of the Key Derivation Function for Low Data Rate Applications

Even though the proposed scheme is designed essentially for multimedia contents, yet it can also be adapted for small messages (low data rate applications), as the scheme is flexible and can be easily configured. For application with small-sized messages, the update of the dynamic key and cipher primitives are not done for each input message and can be performed after a certain configured data threshold length (τ) that can be considered as the size of a set of small-sized messages (called δ), the dynamic key is updated and consequently the cipher primitives. $\delta = \lfloor \frac{\tau}{l} \rfloor$, and l represents the maximum size of small-sized messages. In fact, δ is based on τ; a low value of τ means a high level of security but requires more overhead in terms of delay and resources.

On the other hand, for each δ small messages, we propose to update S_1 in function of S_2 and S_2 in function of S_1 after each message is encrypted. This will prevent attackers from detecting any useful information about a repeated small message. The proposed update substitution primitive after each message encryption is as follows:

$$\begin{aligned} Temp &= S_1 \\ S_1 &= S_1(S_2 >> mod(it,\ 256)) \\ S_2 &= S_2(Temp) \end{aligned} \tag{3}$$

where $S_2 >> v$ circularly shifts the values in the substitution table S_2 by v elements. Besides, *it* represents the counter update iteration and it is incremented by 1 after each update time. This adaptation of the proposed key derivation function for low data rate applications is to show that the proposed cipher can be also efficient for small-message applications and to provide more information about its configuration.

3. Proposed Cipher Scheme

The proposed cipher can handle any type of data messages such as an image, video, audio or text file. Also, the proposed cipher can be performed with or without chaining. First, the proposed scheme is presented without chaining, and it can be considered as a dynamic ECB mode (D-ECB) in order to avoid the issues associated with the static ECB mode [4]. This is done by selecting the blocks to be encrypted in a dynamic pseudo-random order instead of the typical sequential order. The block selection is based on a dynamic permutation table that is derived from the dynamic key that changes for each input message. Moreover, we perform a pseudo-random mixing of two blocks instead of one to increase the randomness of the ciphertext and to make cryptanalysis even more difficult.

For each input message, a dynamic key is produced and accordingly, the different cipher primitives are produced:

1. Two sets of m round keys $RK1$ and $RK2$;

2. A selection table SRK

3. Two substitution tables (S_1 and S_2);

4. and permutation table (π)

All these cipher primitives are required in the encryption algorithm (see Eq. 4 and 6). These same primitives are used in the decryption process except for the substitution tables that are replaced by their inverses, S_1^{-1} and S_2^{-1}. The input message is padded if necessary (nb should be even), and then, divided into nb blocks $M = m_1, m_2, \ldots, m_{nb}$. Moreover, each block has a length of N bytes, where N is a configuration parameter and it can change according to the application or device constraints. A lower value of N is preferable for real-time applications.

3.1. Encryption/Decryption Algorithms

The proposed encryption algorithm consists of two sub-functions: BlocksSelection, and RoundFunction(RF), which are described below.

3.1.1. ***Blocks Selection***

The proposed cipher encrypts two input blocks in each iteration, ($m_{\pi(i+\frac{nb}{2})}$ and $m_{\pi(i)}$) where $i = \{1\ ,\ 2,\ \ldots, \frac{nb}{2}\}$. Note that, $m_{\pi(i)}$ represents the $\pi(i)^{th}$ input block and both input blocks are selected according to the dynamic permutation table π. This complicates the cryptanalysis since there is no sequential relationship between the mixed blocks and the neighboring encrypted blocks. Then, the proposed round function RF is iterated on each couple of input blocks ($m_{\pi(i+\frac{nb}{2})}$ and $m_{\pi(i)}$).

3.1.2. ***Round Function*** *RF*

The round function uses both substitution tables S_1 and S_2 and the selection round key table SRK in addition to two round keys ($RK1$ and $RK2$) that can be generated in a recursive manner for each input block in the first variant of round key generation. While a set of m round keys ($RK1$ and $RK2$) is generated at the initialization step in the second variant of the round key generation. In fact, in the following, we present only the second variant. However, a slight modification is needed when the first round key generation is used, which is the use of the produced round keys $RK1_i$ and $RK2_i$ instead of $RK1_{SRK(i)}$ and $RK2_{SRK(i)}$, respectively for the i^{th} input block.

In each iteration, a couple of input blocks ($m_{\pi(i+\frac{nb}{2})}$ and $m_{\pi(i)}$) are selected and encrypted to produce two encrypted blocks ($c_{\pi(i)}$ and $c_{\pi(i+\frac{nb}{2})}$) as shown in Eq. 4.

$$\begin{aligned} c_{\pi(i)} &= S_2(m_{\pi(i)} \oplus S_1(m_{\pi(i+\frac{nb}{2})} \oplus RK1_{SRK(i)})) \\ c_{\pi(i+\frac{nb}{2})} &= S_1(S_2(m_{\pi(i)} \oplus c_{\pi(i)}) \oplus RK2_{SRK(i)}) \end{aligned} \tag{4}$$

Note that the $\pi(i)^{th}$ encrypted block $C_{\pi(i)}$ is obtained by mixing (xor) $m_{\pi(i+\frac{nb}{2})}$ with the $SRK(i)^{th}$ round key of $RK1$ ($RK1_{SRK(i)}$). Then, its corresponding output is substituted using the first substitution table S_1. Next, the substituted output is mixed with the $\pi(i)^{th}$ message plain block $m_{\pi(i)}$. Finally, another substitution is done on the output by using the substituting S_2 table.

On the other hand, the $\pi(i+\frac{nb}{2})$ encrypted block $C_{\pi(i+\frac{nb}{2})}$ is obtained by mixing (xor) $C_{\pi(i)}$ with the $\pi(i)^{th}$ message block $m_{\pi(i)}$, followed by applying the substitution operation on this output by using S_2. Then, its corresponding output is mixed with the $SRK(i)^{th}$ round key of $RK2$ ($RK2_{SRK(i)}$). Finally, the output is substituted by using S_1.

As seen previously, the mixing between blocks depends on the permutation table π that changes for each input message. As such, all blocks will be encrypted similarly to form the encrypted message C, which will be transmitted securely to the desired destination or to securely stored locally. The encryption algorithm of the first variant is presented in Algorithm 1.

Algorithm 1 The proposed One Round Encryption Algorithm without Chaining Operation mode.

```
procedure One_Round_Encryption($X$)
    for $i = 1$ to $\frac{nb}{2}$ do
        $cX[\pi[i]] = S_2(S_1(X[\pi[i]] \oplus RK_1[SRK[i]]) \oplus X[\pi[i+\frac{nb}{2}]])$
        $cX[\pi[i+\frac{nb}{2}]] = S_1(S_2(X[\pi[i+\frac{nb}{2}]] \oplus cX[\pi[i]]) \oplus RK_2[SRK[i]]))$
    end for
end procedure
```

The decryption algorithm differs only by using the inverse round function RF^{-1} that has the reverse order of the round function. Moreover, RF^{-1} uses the inverse substitution tables S_1^{-1} and S_2^{-1}. In the following, RF^{-1} is described:

$$\begin{aligned} m_{\pi(i)} &= S_2^{-1}(S_1^{-1}(c_{\pi(i+\frac{nb}{2})}) \oplus RK2_{SRK(i)}) \oplus c_{\pi(i)} \\ m_{\pi(i+\frac{nb}{2})} &= S_1^{-1}(S_2^{-1}(c_{\pi(i)}) \oplus m_{\pi(i)}) \oplus RK1_{SRK(i)} \end{aligned} \tag{5}$$

3.2. Proposed cipher scheme with chaining mode of operation

The second variant of the proposed cipher scheme uses a chaining mode of operation. Two initial vectors are required for this variant $IV1$ and $IV2$. They may be initialized to zeros (all bytes are equal to 0) for the first couple of blocks and then updated as presented in the following equation:

$$\begin{aligned} c_{\pi(i)} &= S_2(m_{\pi(i)} \oplus S_1(m_{\pi(i+\frac{nb}{2})} \oplus IV1 \oplus RK1_{SRK(i)})) \\ c_{\pi(i+\frac{nb}{2})} &= S_1(S_2(m_{\pi(i)} \oplus IV2 \oplus c_{\pi(i)}) \oplus RK2_{SRK(i)}) \\ IV1 &= c_{\pi(i)} \\ IV2 &= c_{\pi(i+\frac{nb}{2})} \end{aligned} \tag{6}$$

The encryption algorithm of the second variant is described in Algorithm 2.

Algorithm 2 The proposed One Round Encryption Algorithm with Chaining Operation mode.

```
procedure One_Round_Encryption($X$, $IV1$, $IV2$)
    for $i = 1$ to $\frac{nb}{2}$ do
        $cX[\pi[i]] = S_2(S_1(X[\pi[i]] \oplus IV1 \oplus RK_1[SRK[i]]) \oplus X[\pi[i+\frac{nb}{2}]])$
        $cX[\pi[i+\frac{nb}{2}]] = S_1(S_2(X[\pi[i+\frac{nb}{2}]] \oplus IV2 \oplus cx_i) \oplus RK_2[SRK[i]]))$
        $IV1 \leftarrow X[\pi[i]]$
        $IV2 \leftarrow X[\pi[i+\frac{nb}{2}]]$
    end for
end procedure
```

Similarly to the first variant, the decryption algorithm uses the inverse round function RF^{-1} with chaining operation mode as presented in the following equation:

$$\begin{aligned} m_{\pi(i)} &= S_2^{-1}(S_1^{-1}(c_{\pi(i+\frac{nb}{2})}) \oplus RK2_{SRK(i)}) \oplus c_{\pi(i)} \oplus IV2 \\ m_{\pi(i+\frac{nb}{2})} &= S_1^{-1}(S_2^{-1}(c_{\pi(i)}) \oplus m_{\pi(i)}) \oplus RK1_{SRK(i)} \oplus IV1 \\ IV1 &= c_{\pi(i)} \\ IV2 &= c_{\pi(i+\frac{nb}{2})} \end{aligned} \tag{7}$$

The second variant can be considered as a Dynamic Cipher Block Chaining (D-CBC) since the chaining operation, in addition of the selection and mixing, are based on the dynamic permutation table. To the best of our knowledge, this is the first work that proposes a dynamic CBC mode with dynamic block selection and mixing. The importance of this solution is that it further complicates and randomizes the relationship among encrypted blocks. This leads to a higher level of randomness and makes it more immune against attacks.

3.3. Comparison Between Cipher Variants

The proposed cipher without the chaining operation can be executed in parallel, which is not possible with the the chaining variant. As such, the encryption scheme without chaining ensures a minimum encryption latency. However, parallel decryption is possible for both variants. Moreover, the chained scheme requires just an additional "exclusive or" operation for each encrypted block compared to the unchained variant. Therefore, the second variant requires a slightly longer execution time, as shown in Figure 14, compared to the encryption scheme without the chaining operation for $N \leq 50$.

Moreover, the security level of the chaining operation mode is higher due to its higher level of randomness. Also, the chaining variant that is associated with a dynamic permutation table makes the relation between input blocks more complicated. Table 2 presents a short comparison between the proposed cipher scheme with or without chaining.

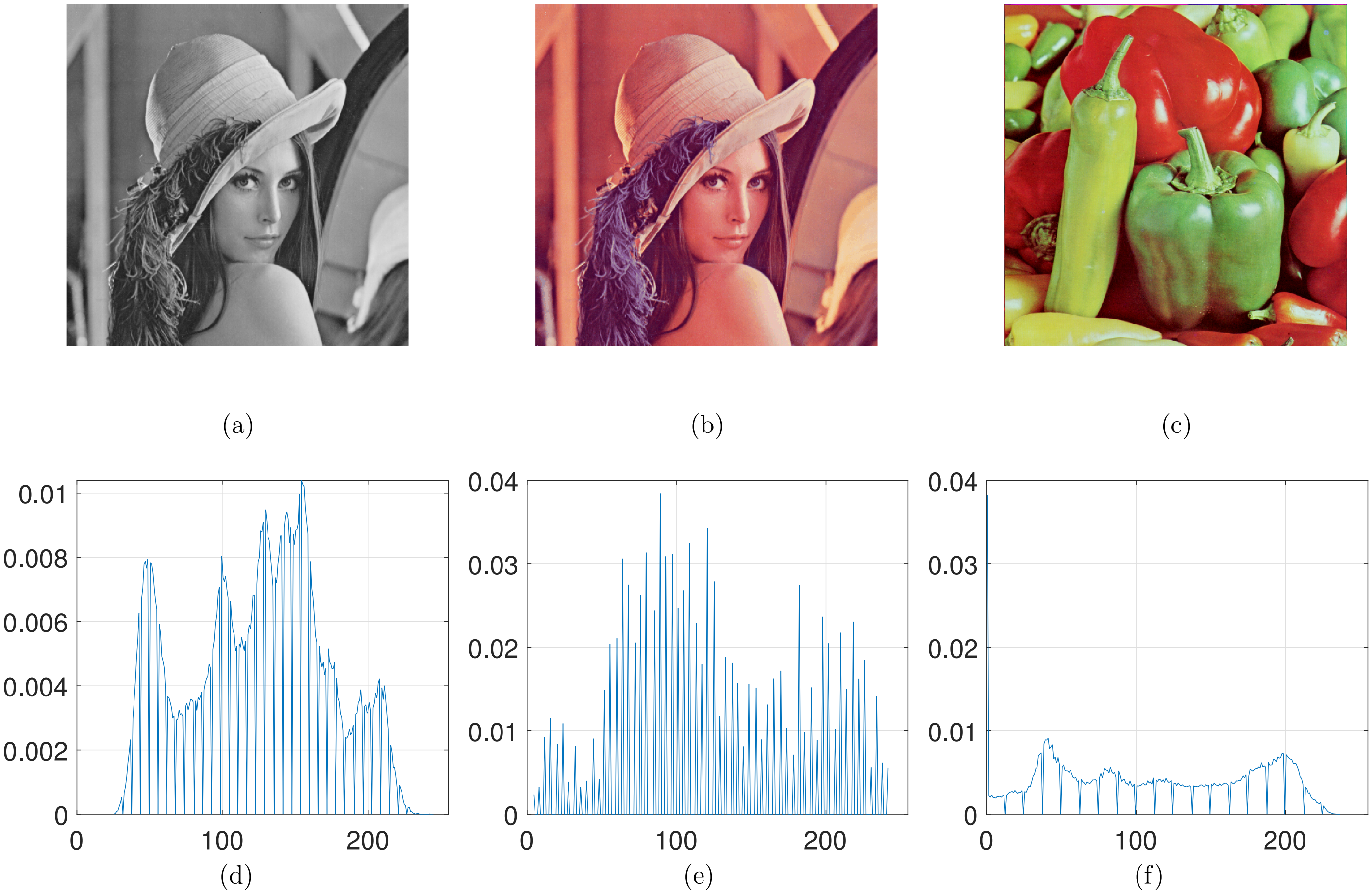


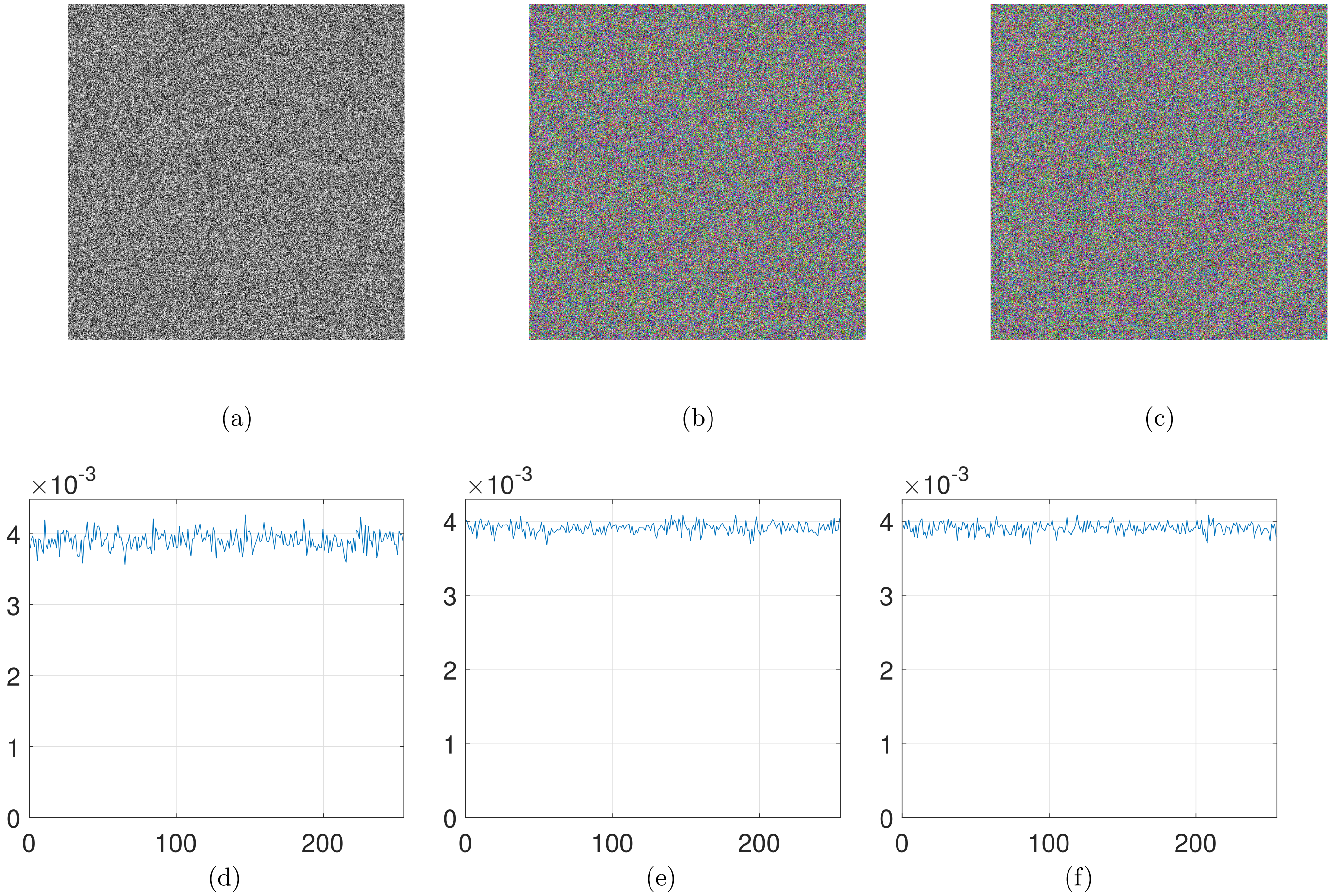
Figure 2: (a) Original Lenna images (a) gray, (b) color, and (c) pepper image. The corresponding PDF of original images (d)-(f), respectively.

Table 2: Comparison between the proposed approach with and without chaining

Metric	**Without Chaining**	**With chaining**
Error propagation	+	++
Parallel encryption	Yes	No
Parallel decryption	Yes	Yes
Delay	Lower one	More delay
Randomness	+	++
Robustness (makes the relation among input blocks more complicated)	+	++

4. Security Analysis

A cipher scheme is considered to be secure and strong, if it can resist implementation-related and analytic attacks such as statistical, differential, chosen/known plaintext/ciphertext,


(a) (b) (c)

(d) (e) (f)

Figure 3: The corresponding encrypted images using the proposed cipher without chaining (a), (b), and (c) in addition to their corresponding PDF (d)-(f), respectively.

in addition to brute-force attacks [20, 26]. In this section, the high level of immunity of the proposed scheme against these attacks is demonstrated. The Lenna and pepper standard images are used as input messages (bytes) for all security and performance tests and for $N = 16$.

4.1. Statistical Analysis

Statistical attacks benefit from a weak level of randomness and uniformity. Therefore, to guard against statistical attacks, the ciphertext should exhibit high randomness and uniformity levels. Several statistical tests were presented in [16] and they were carried out on the obtained ciphertext to validate the uniformity and randomness levels such as Probability Density Function (PDF) and entropy analysis to assess the uniformity, while the recurrence and correlation coefficient between plain and encrypted messages are used to analyze the randomness and the independence property.

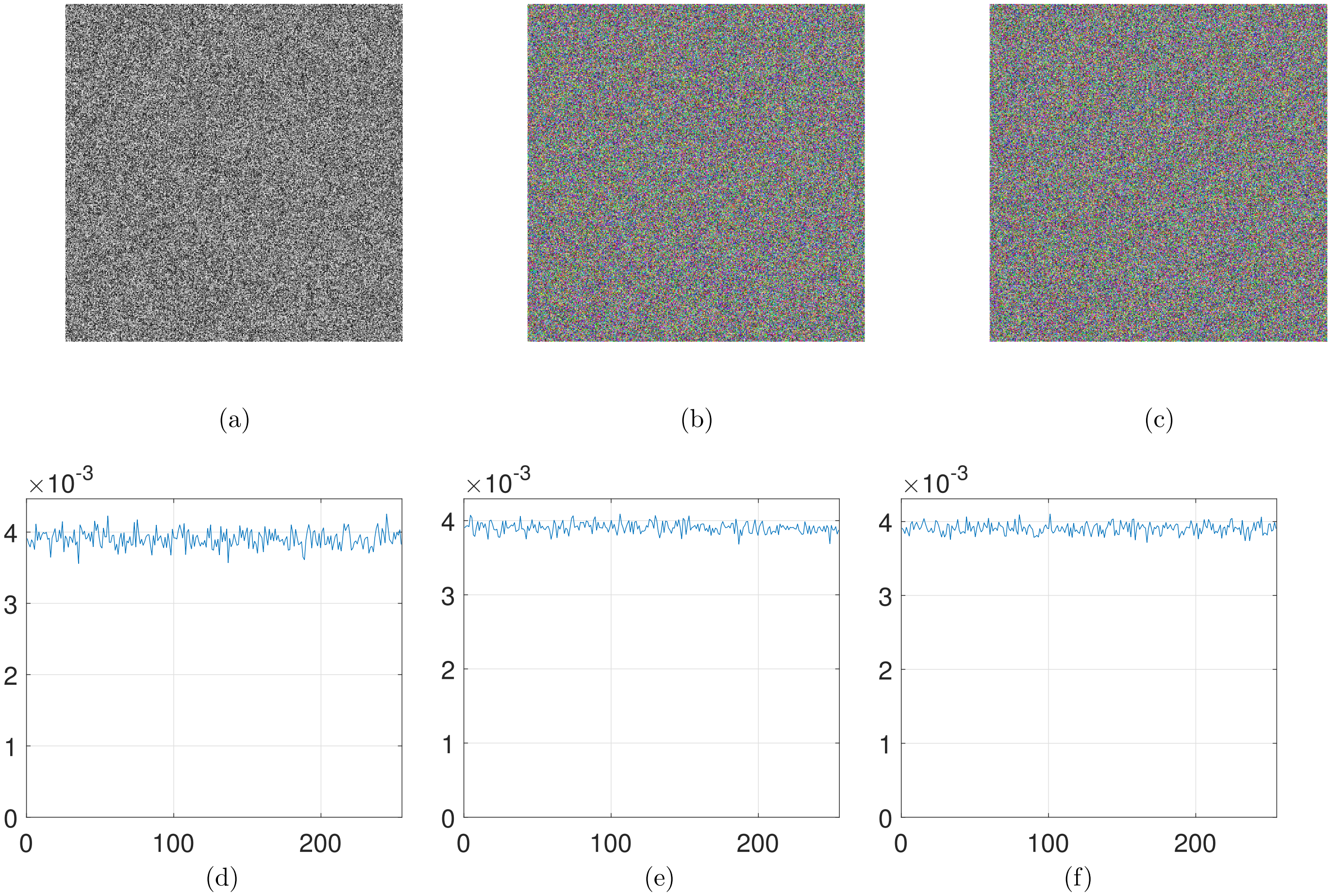
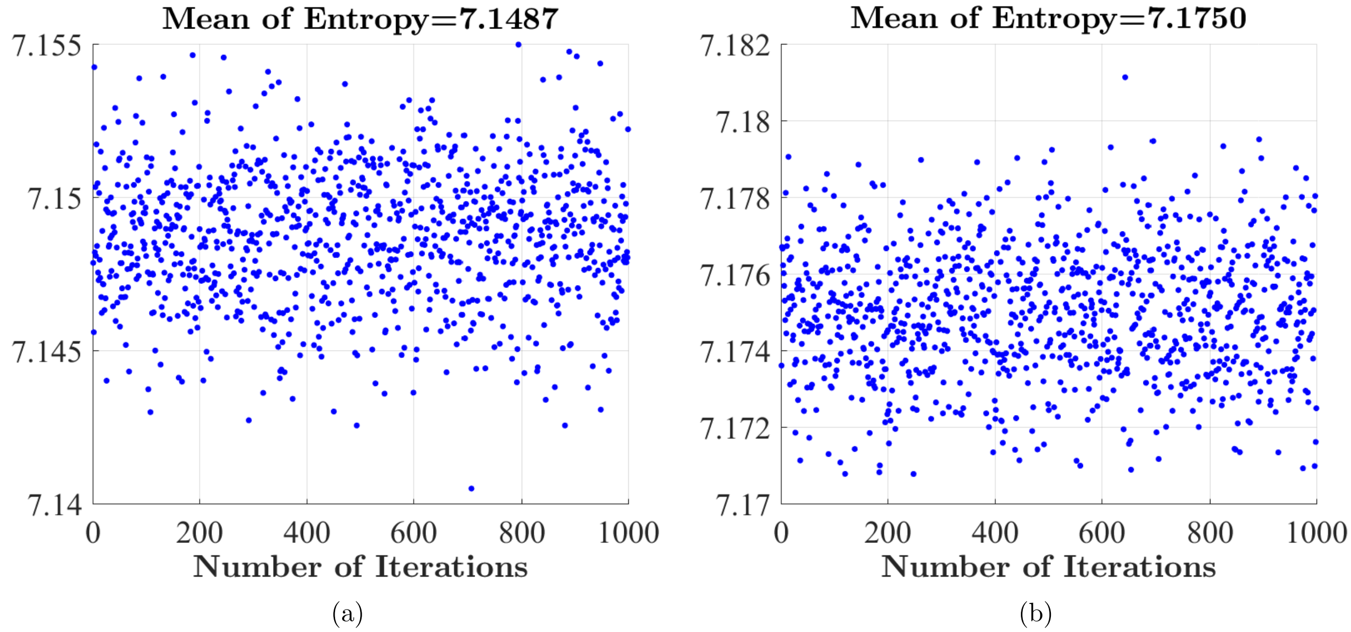


Figure 4: Encrypted Lenna images (a) gray, (b) color, and (c) pepper image with the proposed cipher with chaining operation mode and their corresponding PDF (d)-(f), respectively.

4.1.1. Uniformity Analysis

To resist statistical attacks such as frequency attacks, the ciphertext should satisfy the uniformity property. This means that the frequency of all symbols in the encrypted message should be very close to a uniform distribution; each symbol should have an occurrence probability close to $\frac{1}{n}$, where n represents the symbols' space and it is equal to 8 for byte messages. This can be justified visually and statistically. Visually, it can be proved by plotting the PDF of the encrypted message. The PDF of standard original images and their corresponding encrypted ones are shown in Figure 2, 3, and 4. The visual results clearly indicate that the PDFs of the encrypted images follow the uniform distribution and all symbols have an occurrence probability close to $\frac{1}{256} = 0.039$. Moreover, the cipher scheme with chaining operation achieves a better level of uniformity compared to the unchained variant as seen in Figure 3, and 4.

Moreover, the entropy test at the block level, as described in [16], is used to validate this result. The uniformity at the block level is satisfied if its corresponding entropy value is close to $log_2(N)$, which is the desired value according to [16]. Figure 5 represents the variation of the entropy values for the encrypted Lenna image at the block level using a random key and using 256 bytes as input block lengths for $N = 16$. According to the obtained entropy results, the encrypted blocks always have an entropy close to the desired value (8). Numerical statistical results of the entropy tests are presented in Table 3, which confirm that the uniformity property is achieved.



(a)

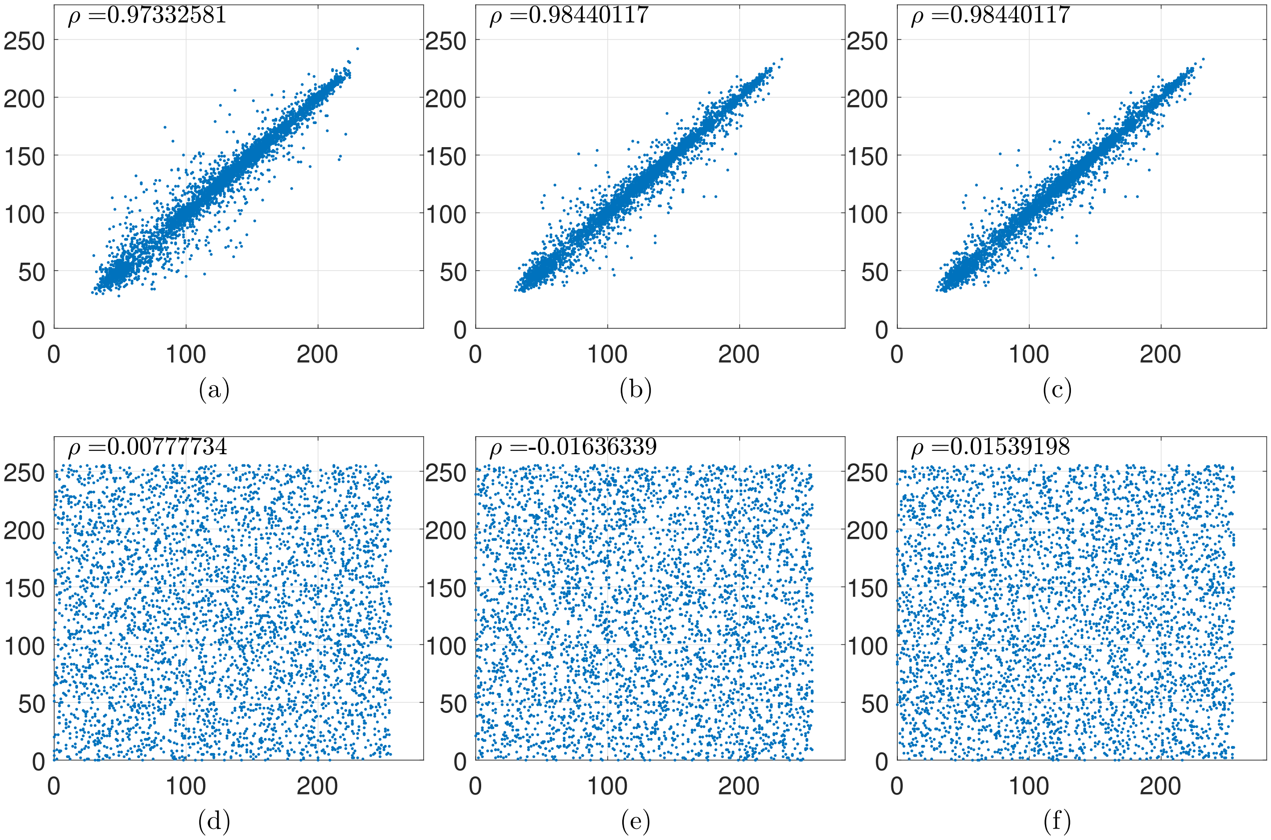
(b)

Figure 5: Variation of the entropy versus its corresponding block index for the encrypted Lenna image using the proposed cipher (a) without and (b) with chaining operation at the block level (256-byte length) and for a random dynamic key with $N = 16$, respectively.

4.1.2. Randomness Analysis

A high level of randomness should exist in the encrypted message. In order to quantify the randomness level, two different tests can be applied: i) the correlation between adjacent elements or ii) the difference between the original and encrypted messages. To quantify the correlation among adjacent bytes, $2,000$ couples of adjacent bytes from the original and encrypted images were selected and in all possible directions (horizontal, vertical and diagonal). The correlation between encrypted adjacent elements is uniformly distributed in space, which is not the case of the original image as can be seen in Figure 6. In addition, the correlation coefficient of adjacent elements of the encrypted ”Lenna” image in horizontal, vertical and diagonal directions versus 1,000 random keys is shown in Figure 7 and all values are very low and close to 0. This indicates that the proposed scheme (with or without chaining) eliminates the spatial redundancy.

On the other hand, the difference test, which quantifies the percentage of the difference between original and encrypted messages should be close to 50% at the bit level. Figure 8 shows the difference between the original and encrypted Lenna images. Table 3 contains numerical statistical results of the difference tests. All these results clearly indicate that



(a) (b) (c)

(d) (e) (f)

Figure 6: Correlation of adjacent pixels in original gray Lenna: (a) horizontally, (b) vertically and (c) diagonally. Correlation in adjacent pixels in ciphered Lenna with a random dynamic key:(d) horizontally, (e) vertically and (f) diagonally.

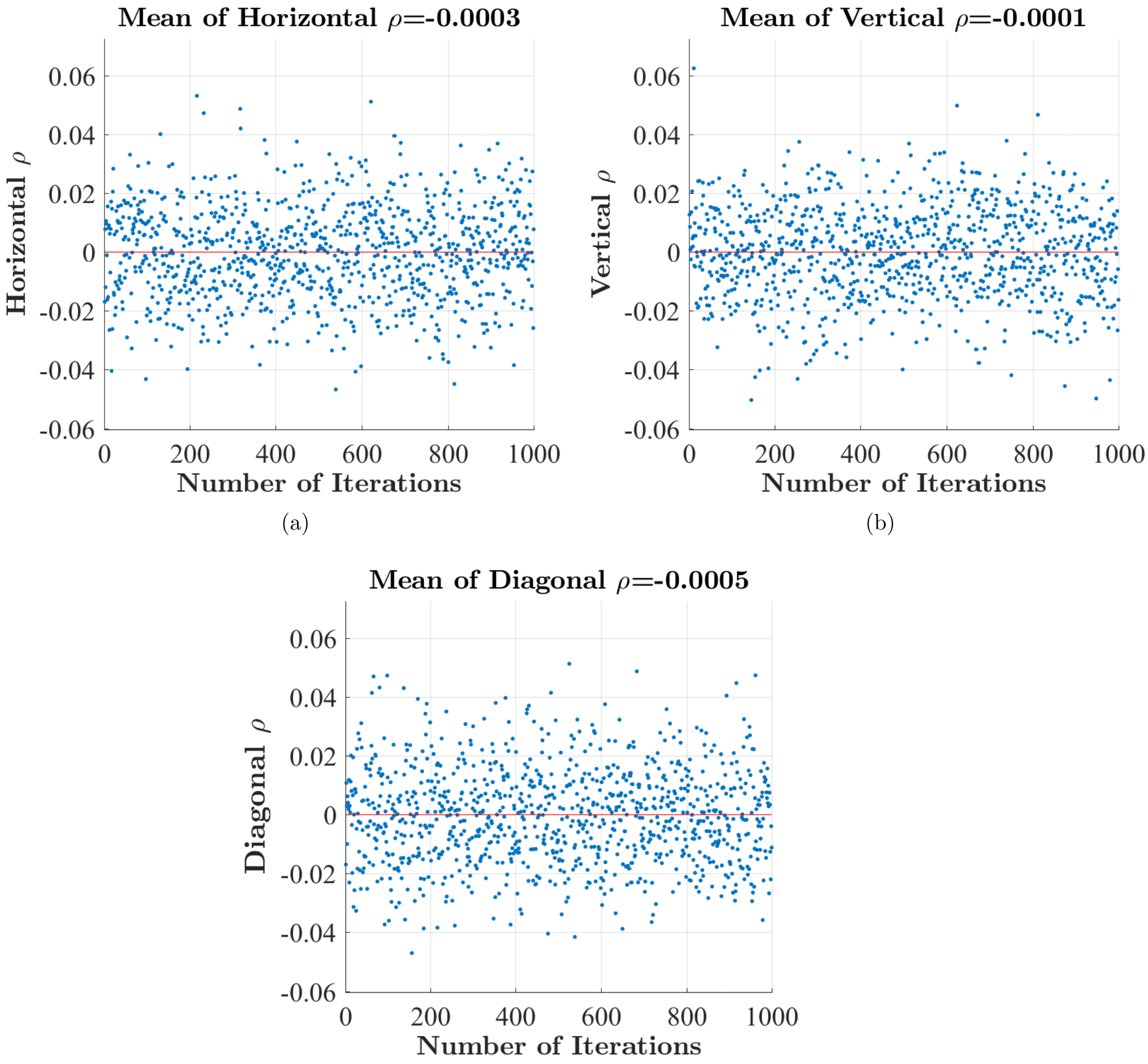
the proposed cipher achieves the required independence between original and encrypted messages.

*4.1.3. **Statistical tests with TestU01 and ”practrand”***

The proposed cipher with either PRNG or pre-generated round keys ($RK1$ and $RK2$) has been tested with 100 different secret keys with TestU01 [12] and ”practrand” [3]. The D-ECB with PRNG variant in addition to both variants of D-CBC (with PRNG or pre-generated round keys) passed successfully all the randomness tests of ”TestU01” and ”practrand” with all the tested keys: an all zeros message (all byte elements are equal to zero) of size 512 × 512 was used. The two tests are very challenging randomness tests and hence, they confirm the high randomness level of the produced ciphertext.

4.2. Sensitivity Test

The sensitivity test is used to validate the avalanche effect of the message and the key. These tests are done to quantify the difference percentages between the encrypted messages



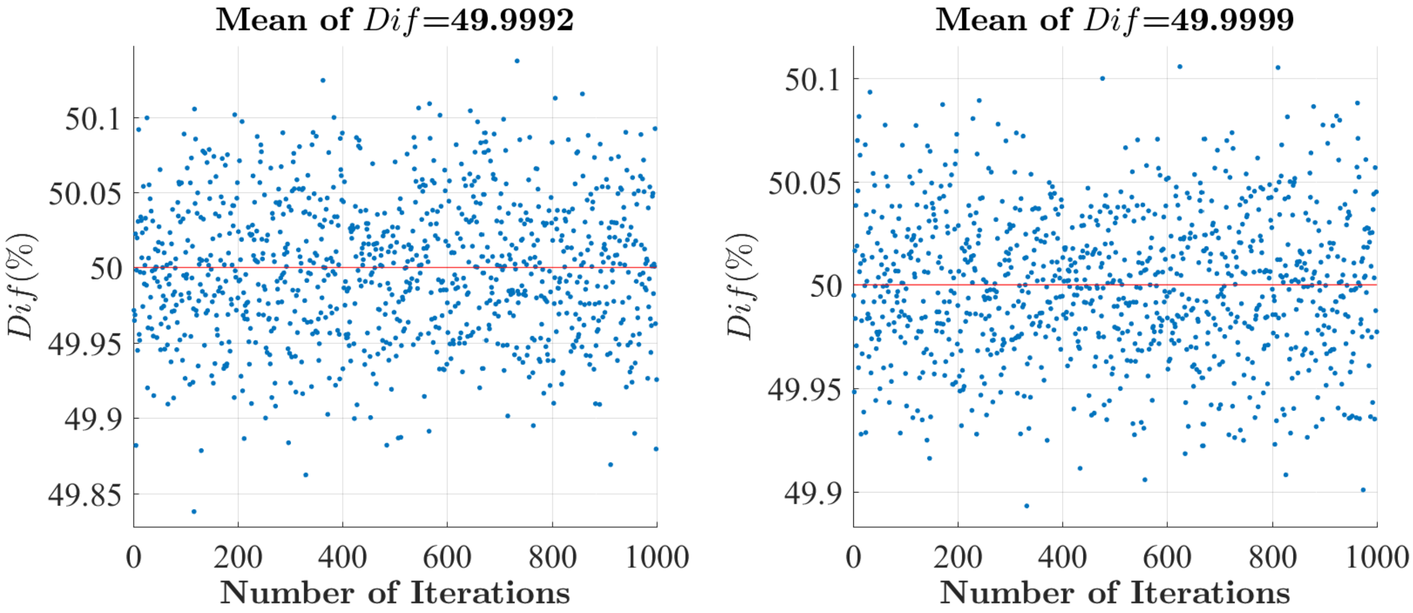
(a)

(b)

(c)

Figure 7: Variation of the correlation coefficient of adjacent pixels in ciphered gray Lenna images:(a) horizontally, (b) vertically and (c) diagonally, respectively using the proposed scheme without Chaining.

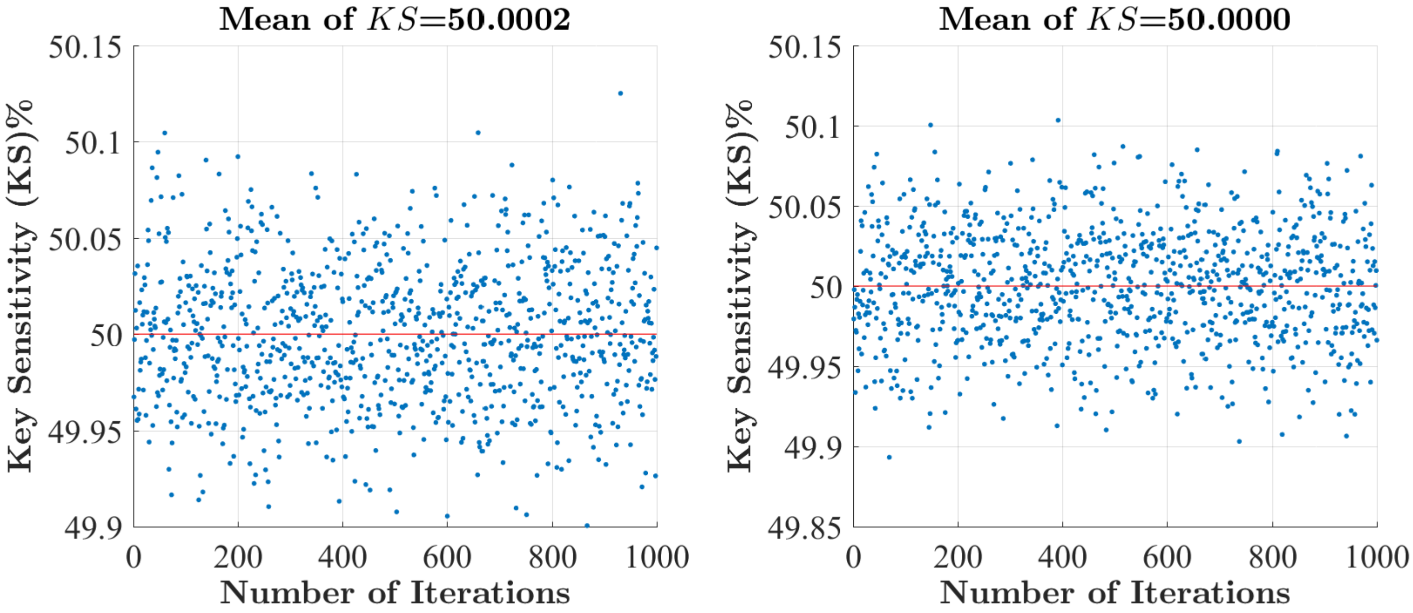
when one bit differs in the original message or in the secret (also dynamic) key. The desired value is 50% difference at the bit level.



(a) Without Chaining

(b) With Chaining

Figure 8: Difference between original and encrypted images against 1,000 random dynamic keys for the proposed cipher without (a) and with (b) chaining operation mode.



(a) Without Chaining

(b) With Chaining

Figure 9: key sensitivity against 1,000 random dynamic keys for the proposed cipher without (a) and with (b) chaining operation mode.

4.2.1. Key Sensitivity Test

Figure 9 shows the dynamic key sensitivity for 1,000 random dynamic keys, and Table 3 shows the numerical statistical results for both cipher variants. According to the results, the difference between both encrypted messages is very close to the desired value. Consequently, this indicates that the proposed cipher scheme satisfies the required level for key sensitivity.

Moreover, a visual example is shown in Figure 10 for a decrypted gray and colored Lenna images with a one-bit difference in the dynamic key. Consequently, the decrypted image with incorrect dynamic key (one-bit difference) carries no useful information about the original image. This justifies that any slight modification in the secret key or in the dynamic key will lead to a different decrypted image that does not contain any useful information about the original message.

4.2.2. Message Sensitivity Test

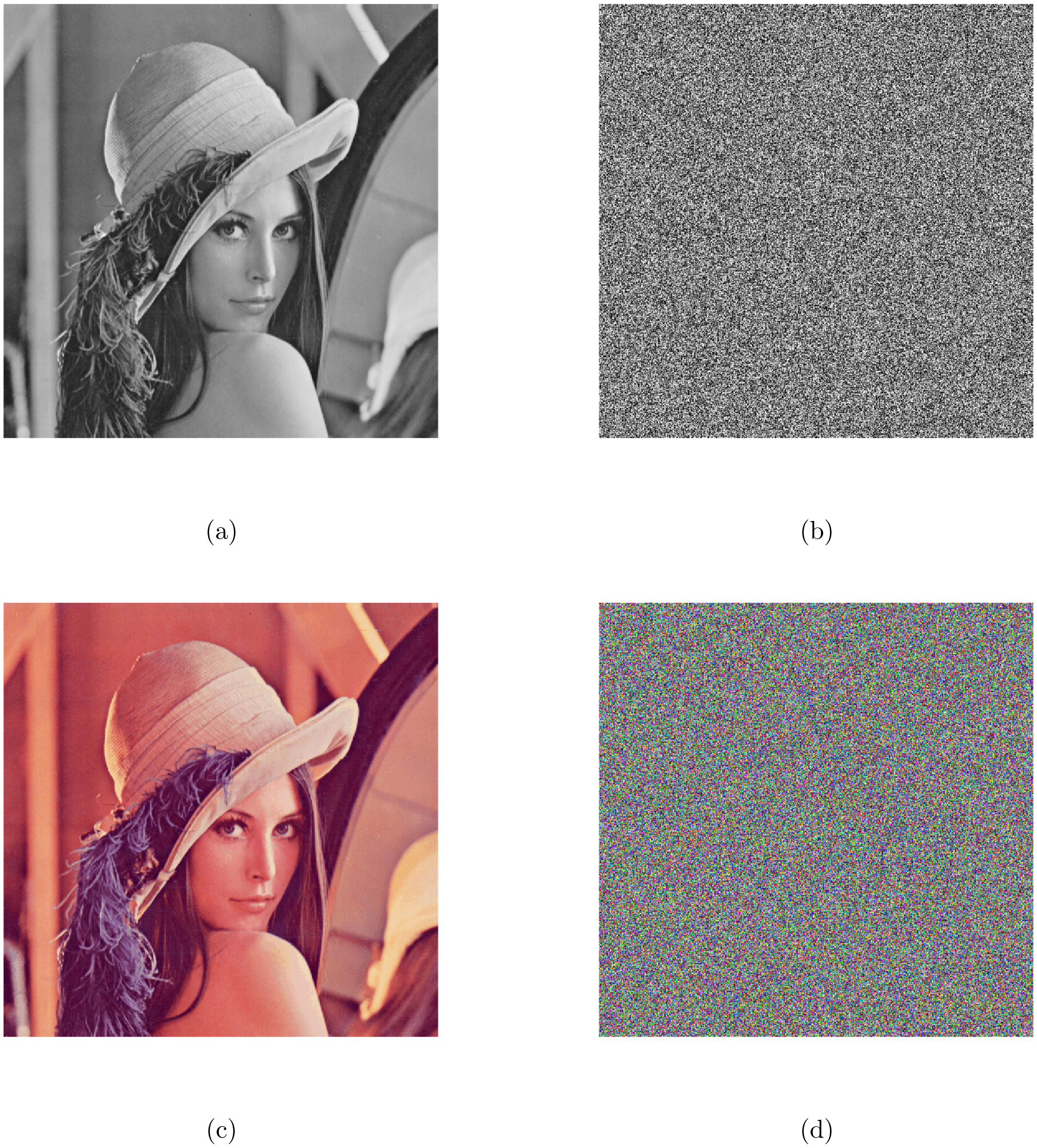
Since the proposed cipher is based on the dynamic key-dependence approach, the dynamic key changes for each input message. Hence, the same message will be encrypted under different dynamic keys. Thus, different encrypted messages will be obtained and with a difference close to 50% as seen in Figure 9. As such, the proposed cipher satisfies the message sensitivity (avalanche effect) by benefiting from the dynamic key approach. Finally, the proposed cipher, with or without chaining, achieves the required message and key sensitivity.

4.3. Visual Degradation

The authors in[16, 19] use PSNR and SSIM metrics (discussed in [9]) to quantify the visual degradation between original and encrypted images. Figure 11 shows the variation of PSNR and SSIM between the original and the encrypted Lenna images versus $1,000$ random dynamic keys. The PSNR has always low values between [9.18, 9.2749] with a mean equals to 9.25 dB and this indicates that encrypted images exhibit hard degradation. In addition, the SSIM values also validate the PSNR results and they are low and vary within [0.0289, 0.043], which is very low and close to zero. Accordingly, no useful visual information about the original image could be revealed from the encrypted image.

5. Analytic, Brute force, and Implementation Attacks

Furthermore, the cipher operation is dynamic since two blocks are randomly selected and mixed in each iteration as opposed to the sequential selection process in traditional ciphers. The selection of input blocks (and chained block in case of the cipher scheme with chaining operation) is based on a dynamically generated pseudo-random permutation table. Moreover, high levels of randomness and uniformity are achieved in the proposed cipher according to Figure 6, 7, 3 and 4, respectively. This indicates that statistical attacks cannot recover any useful information from the ciphertext. Moreover, the independence between original and encrypted messages is validated according to Figure 8. In addition, the proposed cipher scheme uses two different dynamic substitution tables (S_1 and S_2) in a dynamically mixed manner for each input image to increase the non-linearity between original and encrypted


(a)

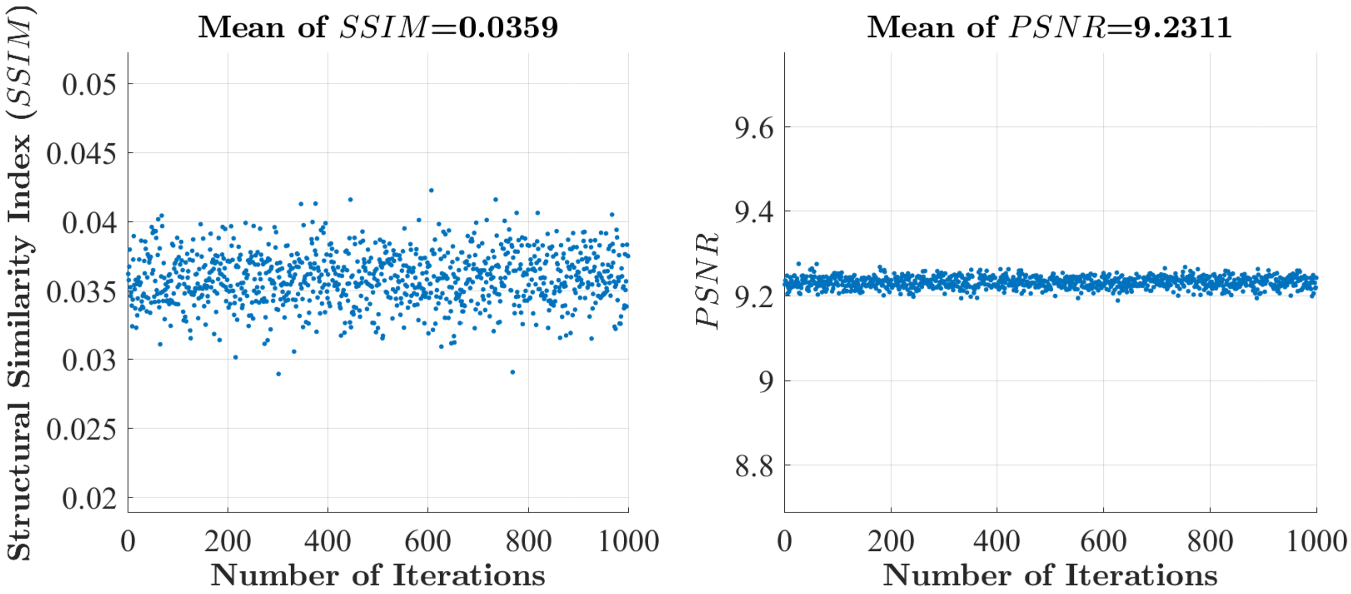
(b)

(c)

(d)

Figure 10: Decrypted gray and color Lenna image using the proposed scheme (without CBC chaining) with its corresponding correct dynamic key (a) and (c) respectively, and with one bit error in the dynamic key used (b) and (d).

blocks. This makes the linear attacks really difficult to be realized. Also, introducing the dynamic chaining operation mode complicates further the relation between input and original messages. The required secret key sensitivity is satisfied since a secure keyed hash function is used and it ensures a high resistance against collision, so the probability to produce the same dynamic key is really close to $\frac{1}{2^{512}}$. Moreover, the dynamic key sensitivity is confirmed according to Figure 9. Hence, the key-related attacks are also very hard to be realized.



(a)

(b)

Figure 11: Histogram of the obtained $PSNR$ and $SSIM$ (between the original and the encrypted Lenna images) for $1,000$ dynamic keys using the proposed scheme without the operation mode.

Furthermore, since the dynamic key changes for each input message, along a high level of randomness, uniformity, and key sensitivity, differential attacks become very complicated and difficult to succeed. On the other hand, each collected encrypted message is encrypted differently with a different dynamic key and consequently, with different cipher primitives. This makes the proposed cipher extremely hard to break by any analytic, or implementation attack. All analytic attacks are unable to break the dynamic cipher scheme since they are designed to break static ciphers with static cipher primitives, which is not the case of the proposed cipher scheme. Finally, the size of the secret key can be 128, 196, or 256 bits, the size of the Nonce is 512 bits, and the size of the dynamic key is 512 bits; these sizes are large enough to make brute force attacks unfeasible.

As a conclusion, the proposed cipher is immune against the different well-known attacks and possibly future ones and its line of defense is based on the dynamic key-dependence approach coupled with a dynamic mode of operation.

6. Performance Analysis

In this section, we analyze the performance of the proposed cipher scheme towards quantifying its effectiveness. Two important metrics are presented in detail, the effect of error propagation and the associated latency.

6.1. Effect of error propagation

The proposed cipher without chaining can be considered as a dynamic ECB mode. The effect of any bit error in the encrypted block c_i will be constrained only to the corresponding

Table 3: Statistical results of the proposed cipher with and without chaining operation using gray Lenna image and for 1,000 random keys.

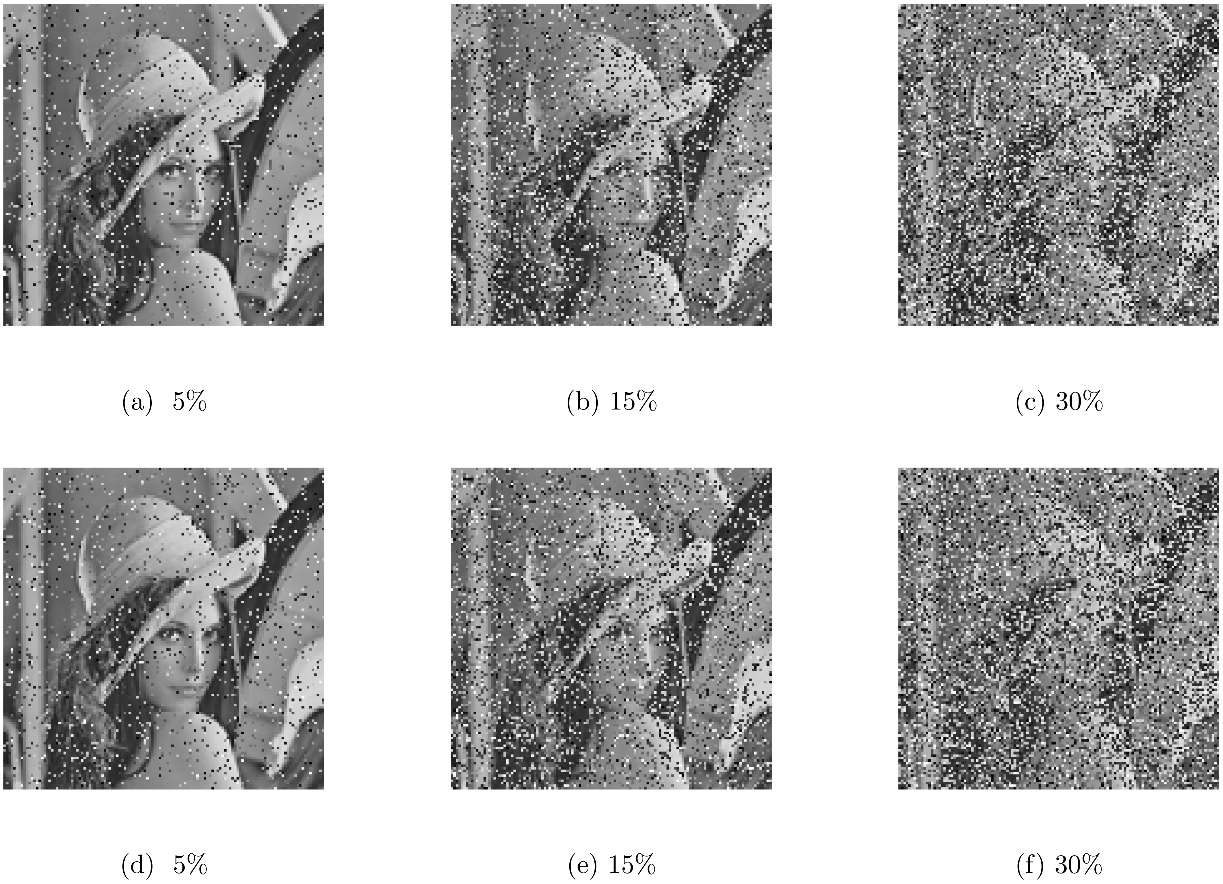
Proposed Scheme Without Chaining Operation				
	Min	*Mean*	*Max*	*Std*
Dif	49.9005	50.0002	50.1251	0.0360
KS	49.8378	49.9992	50.1372	0.0464
H_E	7.1405	7.1487	7.1550	0.0021
ρ_h	-0.0467	-0.0003	0.0532	0.0156
ρ_v	-0.0503	-0.0001	0.0625	0.0153
ρ_d	-0.0470	-0.0005	0.0513	0.0161
PSNR	9.1881	9.2311	9.2749	0.0132
SSIM	0.0289	0.0359	0.0422	0.0019

Proposed Scheme With Chaining Operation				
	Min	*Mean*	*Max*	*Std*
Dif	49.8932	50.0000	50.1034	0.0353
KS	49.8931	49.9999	50.1055	0.0351
H_E	7.1708	7.1750	7.1811	0.0017
ρ_h	-0.0573	-0.0005	0.0461	0.0158
ρ_v	-0.0448	-0.0001	0.0472	0.0157
ρ_d	-0.0518	-0.0006	0.0518	0.0166
PSNR	9.1988	9.2307	9.2645	0.0093
SSIM	0.0304	0.0359	0.0420	0.0017

bytes in the mixed couple of decrypted blocks (m_i and $m_{\pi(i)}$). Moreover, ECB limitations and the trade-off between the avalanche effect and local block error propagation are avoided in the proposed scheme by using the dynamic key approach. Furthermore, the advantage of the proposed cipher is that it limits the effect of error to a byte instead of the whole block as is the case with the traditional block cipher [4], due to the required avalanche effect property.

Concerning the proposed cipher with chaining, any bit(s) error in the encrypted block c_i, will affect four bytes in four different blocks (one byte for each block) ($m_{\pi(i)}$, $m_{\pi(i+\frac{nb}{2})}$, $m_{\pi(i+1)}$, and $m_{\pi(i+\frac{nb}{2}+1)}$). This variant doubles the effect of error propagation compared to the non-chaining one. However, this effect is low as the proposed cipher limits the effect to its corresponding byte instead of block (N bytes). Compared to the original CBC, the effect of any bit error is $N+1$ bytes, where N is the corresponding erroneous block and 1 is the byte in the next neighboring block.

In Figure 13, the effect of errors at the bit level (uniform random distribution within the



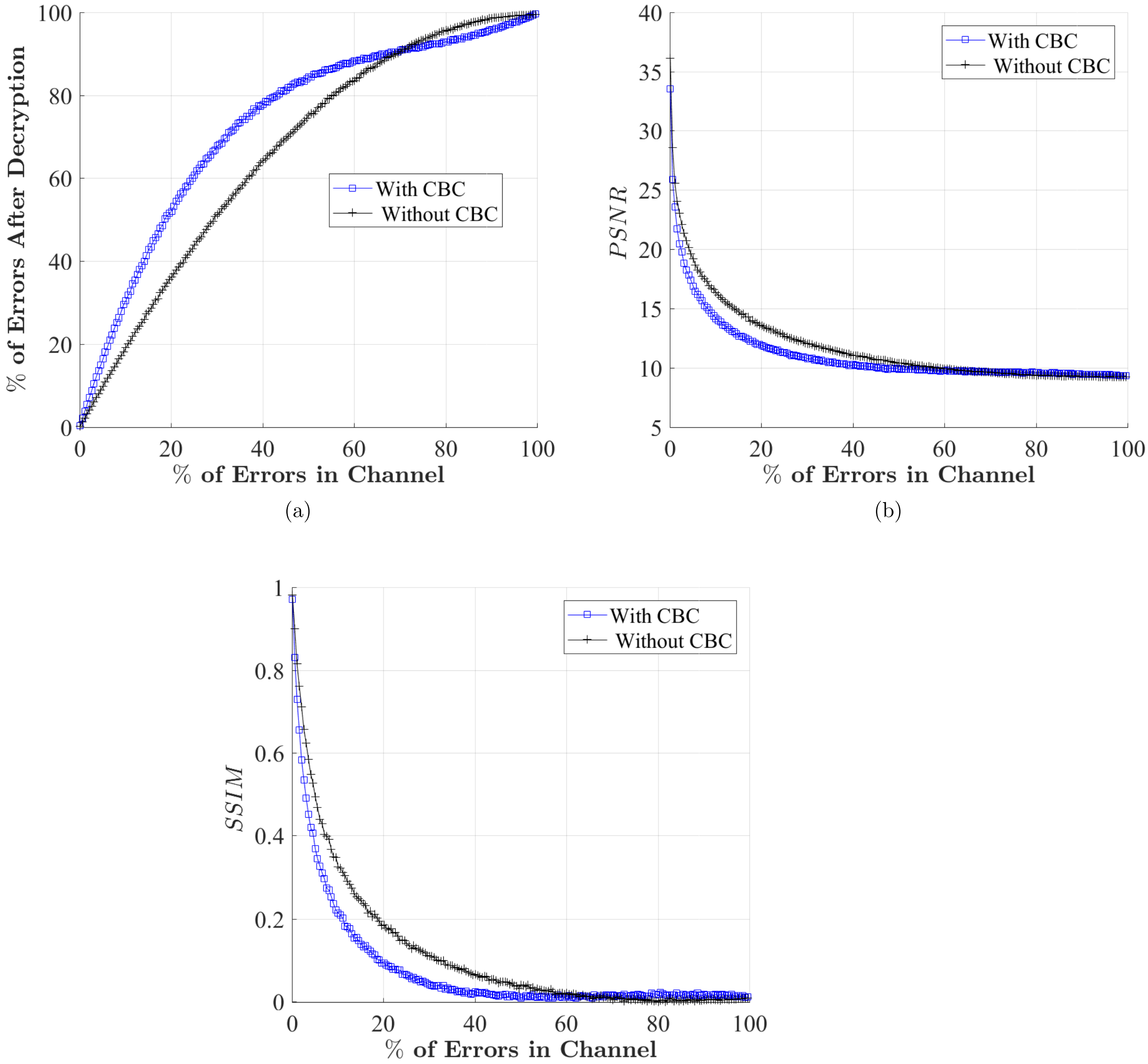
(a) 5% (b) 15% (c) 30%

(d) 5% (e) 15% (f) 30%

Figure 12: Decrypted images in function of different errors percentage using the proposed scheme without (a)-(d) or with (e)-(h) chaining operation mode.

encrypted image) on the proposed cipher scheme with and without chaining are shown. The impact of errors on the proposed cipher scheme is low compared to traditional block ciphers with ECB, CBC or CFB, which exhibit a high error propagation rate (2% of random uniform errors are sufficient to destroy image). Also, visual results of the decrypted noisy images are shown in Figure 12. The effect of errors on the visual degradation of the proposed scheme is shown in Figure 13. The results confirm a low error propagation and consequently, low visual degradation.

When comparing the recent dynamic key-dependent cipher solutions such as [7, 19], we can see that a bit error in any byte of the encrypted block C_i, will affect three blocks $\{\hat{m_{i-1}},\ \hat{m_i},\ \hat{m_{i+1}}\}$ in the decrypted message. Two of them $\{\hat{m_i},\ \hat{m_{i+1}}\}$ have random bit errors that occur independently in any bit position with an expected probability of $\frac{1}{2}$ and the third block $\hat{m_{i-1}}$ has only one specific bit error in the same bit-error position. However, for the proposed scheme without chaining, a bit error introduces only a bit error at the same corresponding byte position for the both mixed blocks, which is equal to the presented solution of [16]. However, with the chaining operation, the error effect is doubled but in



(a)

(b)

(c)

Figure 13: The average variation of the $PSNR$ and $SSIM$ of the proposed cipher scheme versus the percentage of errors.

parallel the randomness and security levels are increased. Actually, for both variants, this result clearly indicates that the proposed cipher exhibits low error propagation compared to the recent dynamic key-dependent cipher schemes.

6.2. Computational Delay

The main objective of the proposed cipher approach is to reach a high level of security with the minimum number of operations. The objective is to reduce the computational complexity, latency and resources (especially energy) for the data confidentiality process.

The required delay of the proposed cipher, with and without chaining, is presented and quantified. To assess the total associated delay, we quantify several delays as follows:

1. T_S denotes the required substitution execution time for a block of N bytes.
2. T_{xor} denotes the required "exclusive-or" execution time between two blocks of N bytes.
3. T_{Sl} denotes the required time to select a couple of input blocks

Therefore, the total Computational Delay (CD) of the proposed scheme with and without chaining to encrypt two blocks is:

$$CD_{D-ECB} = 4 \times T_S + 4 \times T_{xor} + T_{Sl} \tag{8}$$

$$CD_{D-CBC} = 5 \times T_S + 6 \times T_{xor} + T_{Sl} \tag{9}$$

while the total computation delay of the standard AES in [2] to encrypt one block is:

$$CD_{AES} = rT_S + (r+1)T_{xor} + (r-1)T_D + rT_{SR} \tag{10}$$

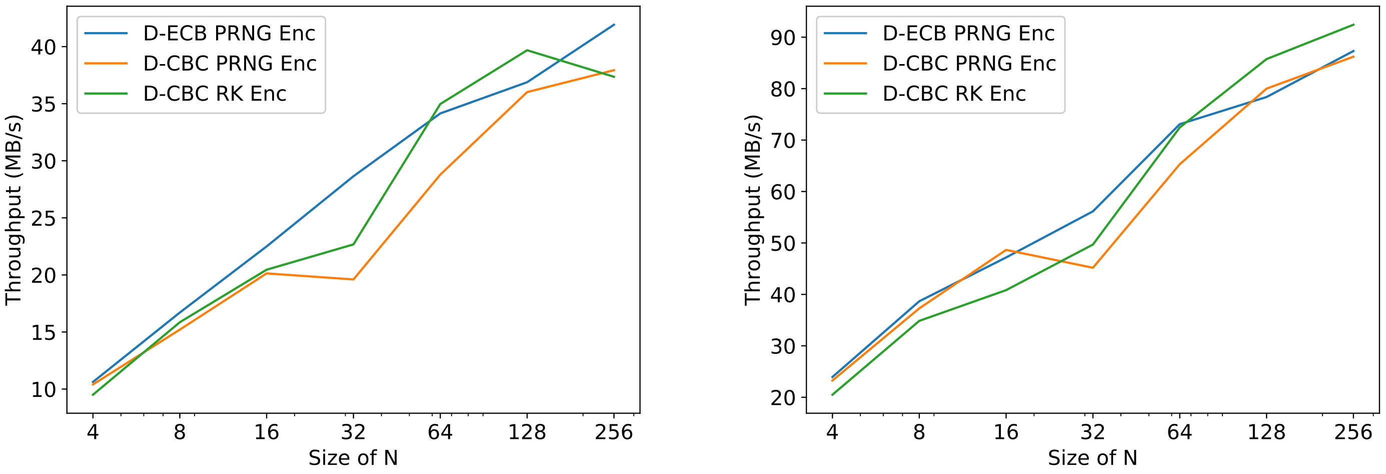
where T_D represents the required delay for the AES Mix-column operations (for all 4 columns) and which has a very high delay compared to other AES operations. T_{SR} represents the required delay for the AES "Shift-rows" operations and r represents the number of rounds. The minimum value of r is 10 for 128 bits secret key, and hence, the minimum AES computation delay is given by:

$$CD_{AES(r=10)} = 10T_S + 11T_{xor} + 9T_D + 10T_{SR} \tag{11}$$

Consequently, the AES computation delay is larger compared to the proposed one with or without chaining. In addition, the proposed solution avoids diffusion operations such as mix-columns of AES towards reducing the required delay. Accordingly, the proposed scheme requires less computational complexity compared to the AES standard cipher for 128-bit length secret key. In fact, AES for 192 and 256 bits secret keys, r is equal to 12 and 14, respectively, and they also require more execution time compared to a 128-bit secret key.

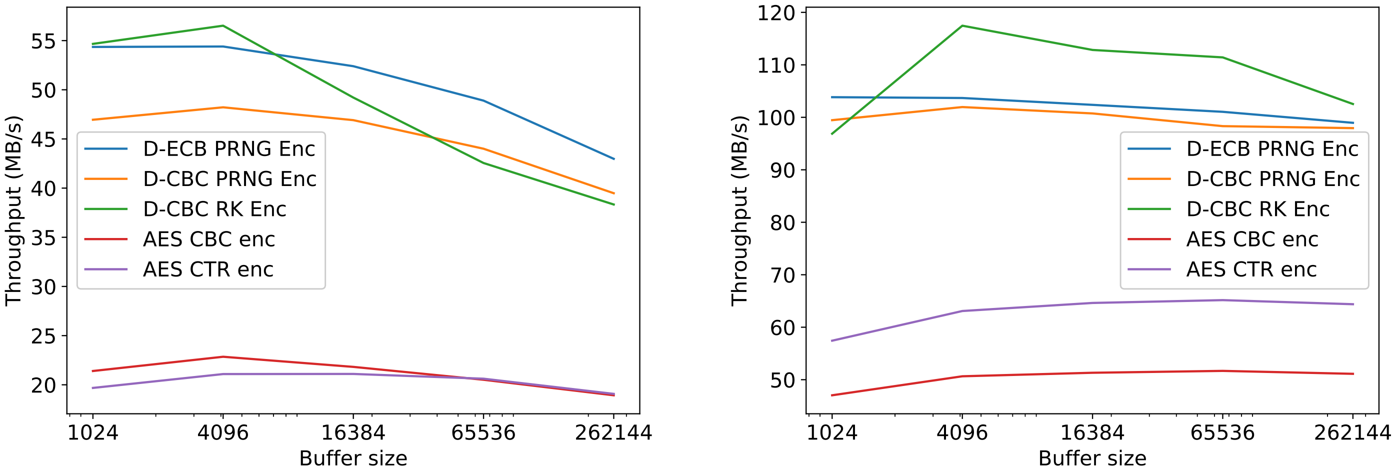
Moreover, the required computational delays of the key derivation function and construction of cipher primitives are described and quantified below. To assess the total associated delay, we quantify several delay components as follows:

1. T_H denotes the required hash time for a block of N bytes.
2. T_{KSA} denotes the required RC4-KSA execution time.



(a) (b)

Figure 14: Throughput in megabytes computed with the average execution times (10000 times) of encryption of the color Lenna image ($512 \times 512 \times 3$) for the 3 variants of our approach with size of N ranging from 4 to 256 on (a) Raspberry Pi0 and (b) Raspberry Pi3.



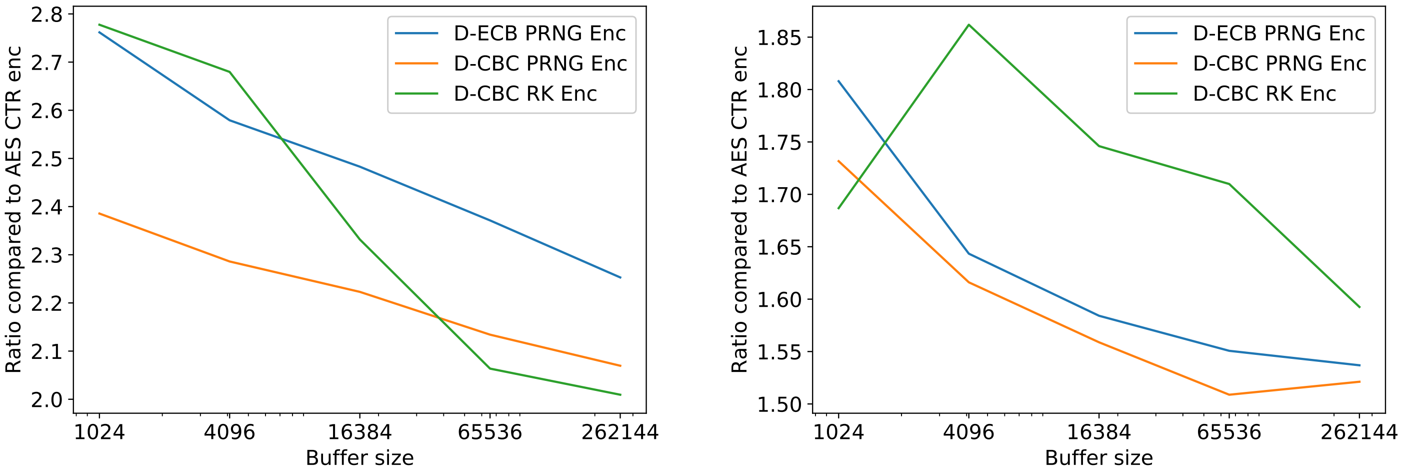
(a) (b)

Figure 15: Throughput in megabytes computed with the average execution times (10000 times) of encryption for the 3 variants of our approach compared to encryption for 2 AES variants for message sizes ranging from 1024 to 262144, with $N = 256$ on (a) Raspberry Pi0 and (b) Raspberry Pi3.

3. $T_{MKSA}(x)$ denotes the required execution time of the modified KSA of RC4 for a table with x elements.

4. T_{PRNG} denotes the required execution time of the Pseudo-Random Number Generation of RC4.

$$CD_{KDF} = T_{xor} + T_H + 3 \times T_{KSA} + T_{MKSA}(nb) + T_{MKSA}(m) + T_{PRNG} \qquad (12)$$



(a)

(b)

Figure 16: Gain of using our approach compared to AES CTR with $N = 256$ on (a) Raspberry Pi0 and (b) Raspberry Pi3.

RC4 is a simple stream cipher, which exhibits a low computational delay and it is being used to construct the cipher primitives such as substitution and permutation tables in addition to round keys. However, it will still introduce a negative effect for small-sized messages, and this is the reason for adopting a different key derivation function for the case of low data rate applications. In such a case, we update the dynamic key and cipher primitives after δ small-sized messages (which is depending on τ threshold data bytes. In addition, all cipher primitives are constant, except the two substitution tables (S_1 and S_2) that are updated after each encrypted message. Therefore, the proposed solution is independent of the message size but depends on the configurable threshold data length. Decreasing τ leads to increase in the security level and the required delay and resources and vice-versa. Finally, δ can be configured according to the application context and the required security level.

6.3. Experiments on Raspberry Pi

The proposed cipher with the dynamic operation modes (ECB and CBC) was implemented in C, and the round keys are generated based on a PRNG or by pre-generating two sets of round keys $RK1$ and $RK2$ before encryption/decryption process, as described previously. As such, four variants are possible, D-ECB using PRNG, D-ECB using pre-generated round keys, D-CBC using PRNG and D-CBC using pre-generated round keys. In fact, ECB with pre-generated round keys was not considered because this variant failed the ”TestU01” and ”practrand” randomness tests.

The other variants that passed the randomness tests are D-CBC with PRNG or with two set of round keys in addition to D-ECB with PRNG. These variants are analyzed and compared to the optimized AES of OpenSSL on two Raspberry Pi devices (Raspberry Pi0, and Raspberry Pi3).

Figure 14 shows the throughput of the proposed approaches when using Raspberry Pi devices and the colored Lenna image, of size 512 × 512 × 3, as input image versus different sizes of N. According to the obtained results, lower execution time is required as N increases. Therefore, a higher value of N is preferable for real-time IoT applications. Moreover, it can be seen that the best block size is $N = 256$.

The experiment results in Figure 15 show that the proposed approaches outperform the OpenSSL implementations of AES with the CBC and the CTR modes. It should be noted that OpenSSL uses optimized assembly instructions towards decreasing the required delay. On the Raspberry Pi0, the fastest variant algorithm was the D-ECB PRNG while on the two other Raspberry Pi, it was D-CBC with pre-generated round keys. In general, both versions of D-CBC have quite similar execution times.

The throughput ratio between the proposed cipher variants and AES-CTR are presented in Figure 16 for different classes of Raspberry Pi. It represents the gain of using the proposed cipher variants instead of using the optimized AES-CTR. These results indicate clearly that the proposed variants require at least half of the AES-CTR execution time.

On the other hand, devices like Raspberry (Pi0 and Pi3) or Arduino are not suitable for AES to achieve the best performance due to their limited processing capabilities. In general, the majority of tiny devices, requires a solution similar to the proposed solution. This paper addresses this issue and provides to that end an efficient cipher solution.

Table 4: Throughput of AES, SPECK and SIMON ciphers [1], One round of [16] and the proposed ones (D-ECB and D-CBC) versus different Raspberry Pi devices

Cipher(Key size, block size)	**Raspberry Pi0**	**Raspberry Pi2**	**Raspberry Pi3**
AES-CTR(128,128)	1.90e+07	2.44e+07	6.22e+07
Speck (256, 128)	1.46e+07	1.36e+07	2.81e+07
Speck (64, 32)	5.32e+06	5.49e+06	9.25e+06
Simon (256, 128)	4.99e+06	4.99e+06	9.98e+06
Simon (64, 32)	4.1e+06	4.14e+06	7.4e+06
Encryption One round [16]	2.43e+07	3.96e+07	7.13e+07
Proposed One (D-ECB PRNG)	4.19e+07	5.47e+07	9.24e+07
Proposed One (D-CBC with PRNG)	3.79e+07	4.95e+07	8.62e+07
Proposed One (D-CBC with pre-generated round keys)	3.73e+07	4.97e+07	8.73e+07

The throughput values of the proposed cipher variants and related ciphers are presented in Table 4 for different classes of Raspberry Pi. In addition, Table 5 represents the ratio between Speck, Simon [1] and the proposed cipher with D-ECB variant. According to the results, Simon requires less execution time compared to Speck. Simon and Speck require at least 6.5 and 9 times overhead compared to the proposed cipher with D-ECB variant. This indicates clearly that the proposed cipher achieves good performance compared to these

Table 5: Ratio between the proposed cipher with D-ECB compared to Speck and Simon.

Cipher(Key size, block size)	**Raspberry Pi0**	**Raspberry Pi2**	**Raspberry Pi3**
Speck (256, 128)	34.85	24.96	30.42
Speck (64, 32)	12.70	10.04	10.01
Simon (256, 128)	11.91	9.13	10.80
Simon (64, 32)	9.76	7.56	8.01

ciphers. Moreover, the throughput of the proposed one is increased by a factor of 72%, 38% and 29% compared to the previous one of [16] with Raspberry Pi0, Raspberry Pi2, and Raspberry Pi3, respectively.

Moreover, let us indicate that the proposed variant D-CBC reduces the throughput compared to the D-ECB with a factor between 7% and 10%. Independent of D-CBC delay overhead, both variants achieve high throughput compared to AES, SPECK, SIMON [1] and [16] according to Table 5.

As a conclusion, the proposed cipher scheme (with or without chaining) achieves low error propagation and requires less delay compared to the existing cryptographic algorithms [1, 16] according to Table 4, which makes it a good candidate for limited devices and for real-time applications.

7. Conclusion and Future Work

The existing secure standard symmetric ciphers are based on a static multi-round function structure that is iterated for a large number of rounds to achieve the desired security level. This introduces a trade-off between the security level and performance. The target of this paper is to solve this issue by designing an efficient cipher scheme that can achieve the required security level with minimum possible operations. As such, in this paper, we present a new efficient cipher scheme with low delay and computational resources. The main innovation of this work is based on a simple secure one round function and by adopting the dynamic operation mode such as ECB and CBC. The selection of two blocks (and previous ones for CBC operation mode) for each iteration is done according to a dynamic permutation table. This cipher solution is suitable for constrained devices and real-time applications. A set of security and performance tests were performed on the proposed cipher (with or without chaining) to prove its efficiency and its robustness against attacks.

This work will be extended in the future to realize an authentication-encryption operation mode with one single pass and also with a single round that requires low computational complexity and resources. This ensures data integrity and source authentication in addition to data confidentiality with low delay and resources overhead.

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 EIPHI Graduate School (contract "ANR-17-EURE-0002").

[1] 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.

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

[3] John D.cook. Testing rngs with practrand: Xoroshiro, xorshift, mt, pcg. `https://www.johndcook.com/blog/2017/08/14/testing-rngs-with-practrand/`, August 2017.

[4] Morris Dworkin. Recommendation for block cipher modes of operation. methods and techniques. Technical report, NATIONAL INST OF STANDARDS AND TECHNOLOGY GAITHERSBURG MD COMPUTER SECURITY DIV, 2001.

[5] Morris Dworkin. Nist special publication 800-38d. *NIST Special Publication*, 800:38D, 2007.

[6] Morris J Dworkin. Sp 800-38c. recommendation for block cipher modes of operation: The ccm mode for authentication and confidentiality. 2004.

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

[8] Tim Grembowski, Roar Lien, Kris Gaj, Nghi Nguyen, Peter Bellows, Jaroslav Flidr, Tom Lehman, and Brian Schott. Comparative analysis of the hardware implementations of hash functions sha-1 and sha-512. In *Information Security*, pages 75–89. Springer, 2002.

[9] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *Pattern recognition (icpr), 2010 20th international conference on*, pages 2366–2369. IEEE, 2010.

[10] Tzonelih Hwang and Prosanta Gope. Robust stream-cipher mode of authenticated encryption for secure communication in wireless sensor network. *Security and communication networks*, 9(7):667–679, 2016.

[11] Charanjit S Jutla. Encryption modes with almost free message integrity. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 529–544. Springer, 2001.

[12] Pierre L'Ecuyer and Richard J. Simard. Testu01: A c library for empirical testing of random number generators. *ACM Trans. Math. Softw*, 33(4):22:1–22:40, 2007.

[13] Kerry A McKay, Larry Bassham, Meltem Sönmez Turan, and Nicky Mouha. Report on lightweight cryptography. *NIST DRAFT NISTIR*, 8114, 2016.

[14] Reem Melki, Hassan N Noura, Mohammad M Mansour, and Ali Chehab. An efficient ofdm-based encryption scheme using a dynamic key approach. *IEEE Internet of Things Journal*, 2018.

[15] Hassan N Noura, Ali Chehab, Mohamad Noura, Raphaël Couturier, and Mohammad M Mansour. Lightweight, dynamic and efficient image encryption scheme. *Multimedia Tools and Applications*, pages 1–35, 2018.

[16] Hassan N Noura, Ali Chehab, Lama Sleem, Mohamad Noura, Raphaël Couturier, and Mohammad M Mansour. One round cipher algorithm for multimedia iot devices. *Multimedia Tools and Applications*, pages 1–31, 2018.

[17] Hassan N Noura and Damien Courousse. Method of encryption with dynamic diffusion and confusion layers, June 9 2016. WO Patent App. PCT/EP2015/078,372.

[18] Hassan N Noura, Mohamad Noura, Ali Chehab, Mohammad M Mansour, and Raphaël Couturier. Efficient and secure cipher scheme for multimedia contents. *Multimedia Tools and Applications*, pages 1–30, 2018.

[19] Hassan N 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.

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

[21] Goutam Paul and Subhamoy Maitra. *RC4 stream cipher and its variants*. CRC press, 2011.

[22] Axel York Poschmann. Lightweight cryptography: cryptographic engineering for a pervasive world. In *PH. D. THESIS*. Citeseer, 2009.

[23] 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.

[24] Phillip Rogaway, Mihir Bellare, and John Black. Ocb: A block-cipher mode of operation for efficient authenticated encryption. *ACM Transactions on Information and System Security (TISSEC)*, 6(3):365–403, 2003.

[25] Rodrigo Roman, Cristina Alcaraz, Javier Lopez, and Nicolas Sklavos. Key management systems for sensor networks in the context of the internet of things. *Computers & Electrical Engineering*, 37(2):147–159, 2011.

[26] William Stallings. *Cryptography and network security: principles and practice*. Pearson Upper Saddle River, NJ, 2017.

[27] 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.