Efficient and Secure Cipher Scheme for Limited IoT Devices

Hassan N. Noura[1,2], Ola Salman[1] and Ali Chehab[1]

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

[2]*Arab Open University, Department of Computer Sciences, Beirut, Lebanon*

Keywords: Cryptanalysis, Data Confidentiality, Cipher Scheme, Avalanche Effect, Key Derivation Function, Binary Diffusion Matrix, Security Analysis.

Abstract: In this paper, an effective and robust cipher scheme is proposed to cater for the resource-constrained nature of IoT devices. The proposed cipher scheme is a combination of static and dynamic cryptographic structures, towards ensuring better resistance and resilience against existing security attacks. More precisely, the proposed solution is designed to be a lightweight cipher scheme, iterating a round function just twice, along with a dynamic key-dependent block permutation. The proposed round function satisfies the required confusion and diffusion properties, and consequently, it guarantees the desired cryptographic aspects such as message and key avalanche effects. Finally, the security and performance tests confirm the effectiveness and the robustness of the proposed cipher solution in terms of security level, the associated delay and required resources.

1 INTRODUCTION

With the emergence of the Internet of Things (IoT), billions of devices will be connected to the Internet. Many of these devices are constrained in terms of power consumption, memory capacity, and/or computational capability. Such devices are used in an IoT network to collect, monitor, and process data for various types of applications. The ubiquitous connectivity of these devices makes them prone to various security threats targeting main different security aspects such as privacy, data confidentiality, integrity (data/system(s)), availability (data/system(s)), and authentication (device/user and data origin authentication). Therefore, protective mechanisms, which can be based on either cryptographic or non-cryptographic solutions, must be employed to preserve data and network security. In this paper, we aim at designing a lightweight cryptographic solution to preserve data confidentiality in the IoT domain. Given the huge amount of generated data and the limitations of IoT devices, lightweight cipher algorithms are needed to cope with the Big Data, time and resource constraints. The existing security solutions are not suitable for constrained IoT devices. For example, traditional cryptographic algorithms that provide data confidentiality, such as the Advanced Encryption Standard (AES) (Daemen and Rijmen, 2013), require a high number of rounds and operations (Noura et al., 2018), exhibiting a high overhead in terms of latency and resources. To resolve this issue, alternative chaotic cryptographic algorithms have been recently proposed. However, these algorithms suffer from several performance and security limitations such as the need for floating-point computations, finite periodicity, and complex hardware implementation. Accordingly, and to respond better to real-time IoT applications and tiny devices, recent works proposed lightweight cipher algorithms with a relatively low number of operations and rounds to minimize the latency and resources (McKay et al., 2016; Poschmann, 2009). Moreover, new lightweight cryptographic algorithms that are based on the dynamic key approach, with relatively small number of rounds, were presented in (Noura et al., 2018; Melki et al., 2018). In this context, this paper combines static and dynamic cipher structures with a low number of rounds to achieve a high level of security with minimum delay and required resources. The proposed cipher includes a simple round function that follows the substitution-diffusion structure. This function is iterated only twice to achieve the desired cryptographic properties. The advantage of the proposed cipher, compared to existing solutions (Noura et al., 2018; Melki et al., 2018), is that it satisfies the avalanche effect at the block level. The proposed cipher uses a dynamic key to produce the required cryptographic primitives for encryption/decryption. This dynamic key is based on a device secret key and a nonce, which complicates the attackers task in guessing and breaking the proposed dynamic cryptographic algorithm and the associated cryptographic primitives. Fi-

Noura, H., Salman, O. and Chehab, A.
Efficient and Secure Cipher Scheme for Limited IoT Devices.
DOI: 10.5220/0009858501420150
In *Proceedings of the 17th International Joint Conference on e-Business and Telecommunications (ICETE 2020) - DCNET, OPTICS, SIGMAP and WINSYS*, pages 142-150
ISBN: 978-989-758-445-9

nally, the security and performance analysis confirm the effectiveness and the robustness of the proposed cipher solution against several cryptographic attacks. The rest of this paper is organized as follows. The proposed key derivation scheme and cipher algorithm are described in Section 2. The security analysis is presented in Section 3. Then, the effectiveness of the proposed cipher scheme is analysed and assessed in Section 4. Finally, Section 5 concludes the paper.

2 PROPOSED CIPHER SCHEME

The majority of the existing cryptographic algorithms (especially the standard ones) are based on a static key and use static substitution and diffusion primitives. The main originality of this work is combining static and dynamic cryptographic approaches toward ensuring a good balance between performance and security level. The proposed cipher scheme uses the static AES substitution table which ensures the strict avalanche criterion in addition to minimum linear and differential approximations (Koo et al., 2006; Koo et al., 2003). Moreover, the proposed static binary diffusion in (Koo et al., 2006; Koo et al., 2003) is used since it ensures a high linear branch number (10 for h=16 and 12 for h=32). Furthermore, the proposed algorithm uses dynamic addition round keys and permutation tables. In the following, we detail the proposed dynamic key and cryptographic primitives derivation, cipher algorithm, and decryption process.

2.1 Dynamic Key and Cryptographic Primitives Derivation

To generate the dynamic key (DK), the device secret key SK is mixed with a $Nonce$ for each set of messages (the number of messages can be configured according to the IoT application requirements). Each generated dynamic key DK is produced by hashing the mix of the secret key and the $Nonce$ as shown by the following equation:

$$DK = h(SK \oplus Nonce) \quad (1)$$

Where h represents the secure cryptographic hash function (SHA-512). The obtained dynamic key is divided into 4 sub-keys (see Figure 1). Each of these sub-keys is employed to produce a specific dynamic key-dependent cryptographic primitive: round keys and 3 permutation tables. The modified Key Setup Algorithm (KSA) of RC4, presented in (Noura et al., 2018), is used to produce the required permutation and selection tables. The permutation table is used in the block permutation process, which is denoted $BL - Pbox$. The selection tables are generated using the technique in (Noura et al., 2019). Furthermore, two selection tables are employed, $Pbox_{SIV1}$ and $Pbox_{SIV2}$, to select the round key for the first and the second iterations, respectively. On the other hand, a set of q round keys is required, and it can be generated using any stream cipher. For example, RC4 can be used to produce these q round keys (called IV), where each round key has h bytes in length.

Algorithm 1: Proposed Encryption Algorithm.

```
procedure ENCR($M$, $S - box, SIV1, SIV2, IV, Bl\_Pbox, h$)
    $Ml \leftarrow reshape(M,\ 1,\ 1 \rightarrow size(M,1) \times size(M,2) \times size(M,3))$
    $n \leftarrow length(M1)$
    $\alpha \leftarrow \lceil \frac{n}{h} \rceil$
    $Ml \leftarrow padding(Ml, \alpha \times h - n)$
    $B \leftarrow reshape(Ml,\ \alpha,\ h)$
    for $it \leftarrow 1$ to$\alpha$ do
        $X \leftarrow B(it,\ 1 \rightarrow h)\ + IV(\ SIV1(it), 1 \rightarrow h)\%256$
        $Y \leftarrow S - box(X)$
        $Z \leftarrow Diffusion(Y)$
        $X \leftarrow Z \oplus IV(\ SIV2(it), 1 \rightarrow h)$
        $Y \leftarrow S - box(X)$
        $Z \leftarrow Diffusion(Y)$
        $Tmp(it,\ 1 \rightarrow h) \leftarrow Z$
    end for
    $Cp \leftarrow Tmp(Bl\_Pbox,\ 1 \rightarrow h)$
    $C \leftarrow reshape(Cp, 1, \alpha \times h)$
    return $C$
end procedure
```

2.2 Encryption Algorithm

The proposed cipher algorithm deals with a flexible block size (h bytes). If the number of bytes in a given message is not a multiple of h (8, 16 and 32), padding is required. Then, the input message is divided into α blocks $\{B_1,\ B_2,\ \ldots,\ B_\alpha\}$, where each block has a length of h bytes.

The proposed cipher scheme is illustrated in Figure 2, while its pseudo-code is included in Algorithm 1. The encryption scheme consists of two main steps. First, the round function is iterated for two times. Second, a block permutation process is introduced to randomize the sequential order of the encrypted blocks. In fact, the round function consists of three operations, byte addition with a round key, byte substitution and binary byte diffusion, as described next.

- The first operation mixes an input message block

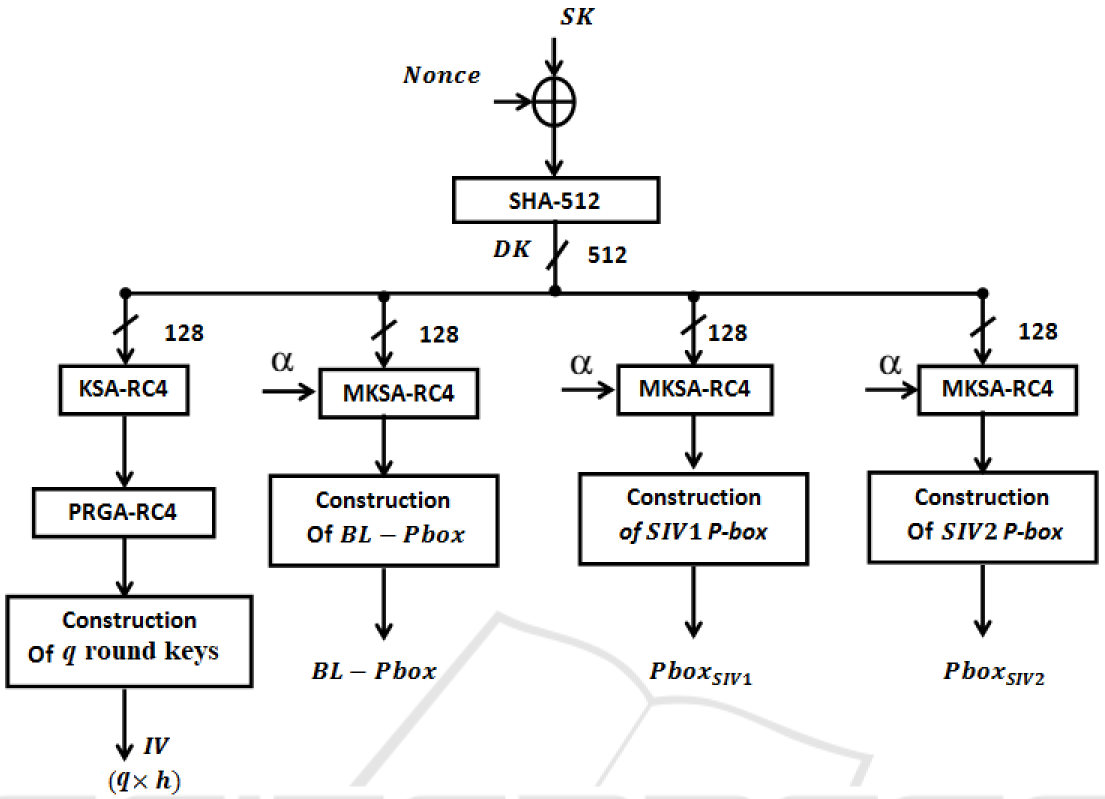


Figure 1: Architecture of the proposed key derivation function.

with a round key represented by the Initial Vector (IV), which is dynamically selected, according to the first selection table $Pbox_{SIV1}$, for the first round and to the second selection table $Pbox_{SIV2}$, for the second round. In this operation, the arithmetic addition modulo 256 is used in the first round and the logical ”exclusive or” is used in the second one.

- Next, the substitution operation is performed based on AES S-box.
- Then, a diffusion operation is performed based on a static binary mixing matrix, G.

In fact, running the proposed cipher for two rounds achieves a high level of randomness in addition to message and key avalanche effects (a slight bit change in the plain-block or dynamic key must produce completely different encrypted blocks). In the following, we summarize the different steps of the proposed round function.

2.2.1 Addition with a Round Key Operation

The proposed addition with a round key operation uses two different instructions. The arithmetic addition modulo 256 is used for the first round, while the logical ’exclusive or’ is used for the second one. The addition is performed between an input block and a selected IV (SIV), representing the round key, as shown in the following equations.

Equation 2 illustrates the addition process.

$$x = A(B,SIV) = \begin{cases} B + SIV \ mod \ 256 & r=1 \\ B \oplus SIV & r=2 \end{cases} \quad (2)$$

Equation 3 illustrates the inverse addition operation that should be employed in the decryption process.

$$B = A^{-1}(X,\ SIV) \begin{cases} X \oplus SIV & r=1 \\ X - SIV \ mod \ 256 & r=2 \end{cases} \quad (3)$$

B, X, SIV have each a length of h bytes. B and X represent the input plain and encrypted blocks respectively, while SIV is dynamically chosen for each block and for each round according to two permutation tables ($Pbox_{SIV1}$, and $Pbox_{SIV2}$) from a set of q different IVs.

2.2.2 Substitution Operation

Mainly, this step is introduced to ensure the confusion property. The static AES substitution table is used to substitute the addition block X, as indicated by the following equation.

$$Y = AES - Sbox(X) \quad (4)$$

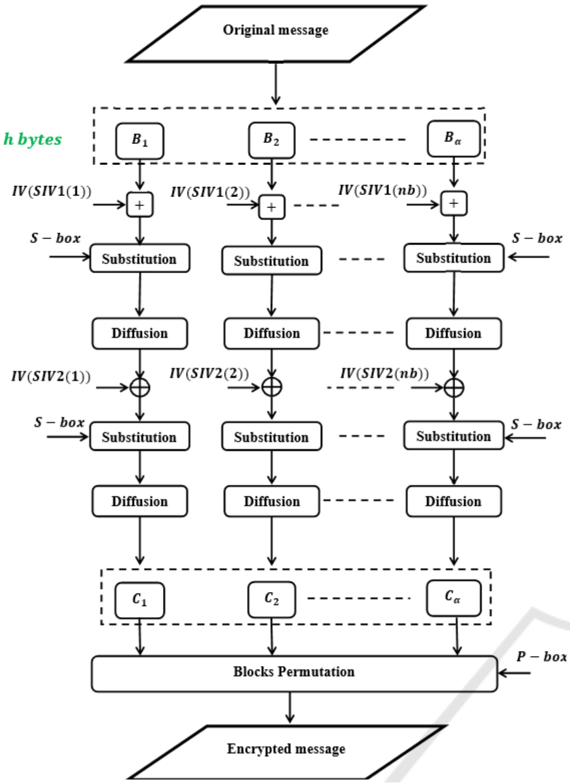


Figure 2: Architecture of the proposed encryption algorithm.

2.2.3 Binary Diffusion Matrix Form

The diffusion process is applied in the binary finite field. We select a static binary diffusion matrix G for each value of h since it only requires the logical ”exclusive or” operation and consequently, lower execution time compared to other diffusion operations (Noura et al., 2019; Koo et al., 2006; Koo et al., 2003).

Figure 3-(a) shows a visual representation of the selected binary matrix G for h=32 (Koo et al., 2006) and its corresponding inverse matrix G^{-1} is shown in Figure 3-(b). For The visual representation of G and G^{-1}, the blue color indicates that the index is equal to 0, otherwise it is 1. Note that the employed matrix should have a maximum linear branch number. Therefore, for $h = 8$, the Camellia binary diffusion matrix is chosen (Aoki et al., 2000), while for $h =$ 16 and 32, the binary matrix of (Koo et al., 2003) and (Koo et al., 2006) are chosen, respectively. The methods proposed in (Koo et al., 2003; Koo et al., 2006) offer some optimization by finding common patterns in the diffusion matrix, which enables the reduction of the required diffusion computational complexity. This will consequently decrease the required execution time of the diffusion and inverse diffusion operations. On the other hand, the proposed solution requires only 2 iterations and consequently less execution time compared to (Koo et al., 2003) and (Koo et al., 2006), which require 8 and 6 iterations, respectively.

In fact, the different index values in each vector (G_i) should be equal to 1 corresponding to the byte introduced in the diffusion process (see (Koo et al., 2006; Noura et al., 2019) for more details about the binary diffusion process). The diffused byte is the result of m XORed bytes, where m represents the number of elements of the diffusion vector with the corresponding index is equal to 1.

2.3 Blocks Permutation

This step consists of permuting the encrypted blocks. It is designed to randomize the order of encrypted blocks independently of the employed operation mode. The block permutation operation is performed using the produced dynamic $P-box$ that has a length of α elements.

2.4 Decryption Algorithm

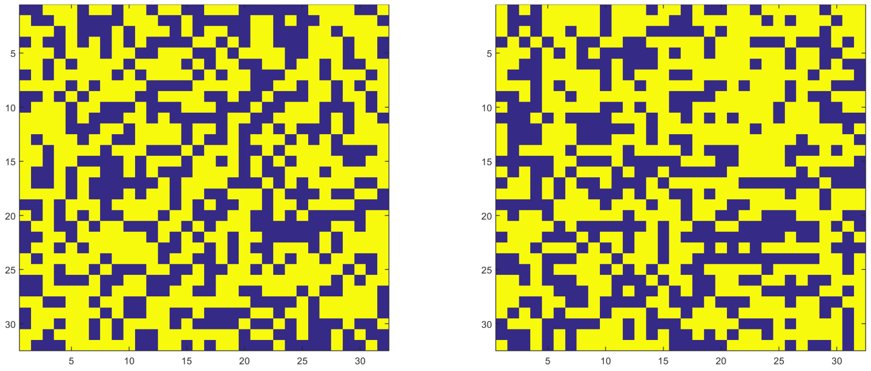
The decryption process consists of applying the inverse block permutation first, then two rounds of the inverse round function. The inverse round function consists of applying the round function in the reverse order and using the inverse AES substitution table in addition to the inverse diffusion matrix G^{-1}. The inverse addition operation is already defined previously and the same round keys should be used. Algorithm 2 details the decryption steps to recover the plain block.

3 SECURITY ANALYSIS

In this section, a comprehensive security analysis is conducted to demonstrate that the proposed cipher is immune against several attacks such as statistical, differential, chosen/known plain-text, and brute-force attacks. In the following tests, the chosen plaintext messages follow the normal distribution with mean equal to 128 and standard deviation equals to 16.

3.1 Resistance against Statistical Attacks

A ciphertext should exhibit a high degree of randomness to resist statistical attacks (Noura et al., 2018; Noura et al., 2017). The proposed cipher scheme should therefore ensure the independence and uniformity criteria. The uniformity of the ciphertext can be



(a) Binary Diffusion Matrix

(b) Corresponding Invertible matrix of (a)

Figure 3: The binary diffusion matrix presented in (Koo et al., 2006) (a) and its corresponding inverse one (b).

Algorithm 2: Proposed decryption algorithm.

```
procedure DECR($C$, $Inv - S - box, SIV1, SIV2, IV, Inv - Bl\_Pbox, h$)
    $Cl \leftarrow reshape(M,\ 1,\ 1 \rightarrow size(C,1) \times size(C,2) \times size(C,3))$
    $n \leftarrow length(C1)$
    $\alpha \leftarrow \lceil \frac{n}{h} \rceil$
    $Cl \leftarrow reshape(Cl,\ \alpha,\ h)$
    $Cp \leftarrow Tmp(Cl\_Pbox,\ 1 \rightarrow h)$
    for $it \leftarrow 1$ to$\alpha$ do
        $Z \leftarrow Inverse\_Diffusion(Cp(it,\ 1 \rightarrow h))$
        $Y \leftarrow Inv - S - box(Z)$
        $X \leftarrow Y \oplus IV(\ SIV2(it), 1 \rightarrow h)$
        $Z \leftarrow Inverse\_Diffusion(X)$
        $Y \leftarrow Inv - S - box(Z)$
        $X \leftarrow Y\ - IV(\ SIV1(it), 1 \rightarrow h)\%256$
        $Tmp(it,\ 1 \rightarrow h) \leftarrow X$
    end for
    $D \leftarrow reshape(Tmp, 1, \alpha \times h)$
    $D \leftarrow Eliminate\_Padding(D)$
    $D \leftarrow reshape(D,\ L,\ C,\ P)$
    return $D$
end procedure
```

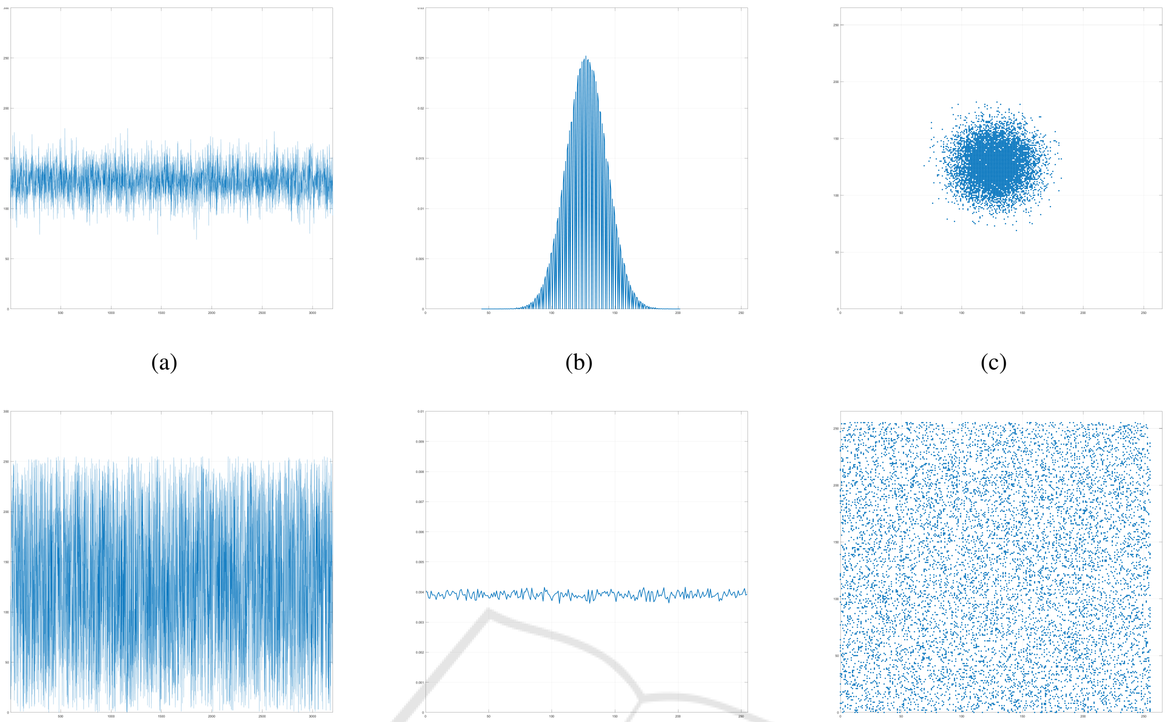
shown visually by analyzing the corresponding Probability Density Function (PDF) and it can be validated using several statistical tests such as the entropy test. In addition, a visual presentation of the encrypted message recurrence can verify the independence criterion of the ciphertext. Moreover, the difference percentage between the original and encrypted messages can also be used to prove the independence between plaintext and ciphertext. These tests were applied and the results, as shown next, clearly confirm that the proposed scheme achieves the required properties and consequently, can guard against statistical attacks.

3.1.1 Uniformity Analysis

To resist common statistical attacks, the encrypted messages should satisfy the randomness property. Therefore, we employ the PDF test to analyze the distribution of the encrypted messages. The PDF of the encrypted messages indicate whether or not the ciphertext distribution is uniform. If the ciphertext distribution is uniform, then each symbol (here each byte) has an occurrence probability close to $\frac{1}{n}$, where n is the number of symbols (here is equal to 256). The amplitude, the corresponding PDF, and the recurrence of selected original messages are shown in Figure 4 a-c. Additionally, the corresponding encrypted messages amplitude, PDF, and recurrence are shown in Figure 4 d-f. It can be observed from the PDF of the encrypted messages, using the proposed cipher scheme, that each symbol has a uniform distribution. Furthermore, the numerical results of ciphertext symbols occurrence probability (Figure 4 e) are very close to 0.039, which represents the ideal value for this test $\frac{1}{256}$.

Moreover, the entropy analysis of the encrypted messages is computed and presented in Figure 5. Here, the entropy is computed for each original message of 256 bytes. The results indicate that the encrypted messages have an entropy close to the ideal value, which is equal to $log_2(n)$ =8 (Noura et al., 2018). Therefore, the proposed cipher algorithm is sufficiently secure against entropy attacks.

On the other hand, the independence (probability of difference at the bit level) between plain and encrypted messages are computed and shown in Figure 6-a). The resulting difference average value is nearly 50%, which is the ideal value. Hence, the encrypted messages are totally different compared to the original ones. As such, the proposed approach satisfies the desired independence propriety.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4: Amplitude variation of the original message (a) in addition to its corresponding probability density function (b), and the original message recurrence (c). Amplitude variation of the obtained corresponding ciphertext (d) in addition to its corresponding probability density function (e), and the ciphertext recurrence for a random dynamic key($h = 32$) (f).

3.2 Resistance against Key-related Attacks

In this section, we assess the proposed cipher scheme against different sorts of key-related attacks. In the following, we present the key sensitivity test results in addition to analyzing the effect of weak keys on the proposed scheme security.

3.2.1 Weak Key Effect

Unlike the static structure used by the standard symmetric cipher algorithms, the proposed cipher scheme relies on the dynamic key structure. The proposed cipher uses a pseudo-random function to produce a dynamic key, which is used to produce several cryptographic primitives (permutation and selection tables in addition to round keys). Then, at each interval (depending on the configuration), the dynamic secret key is updated and consequently a new set of cryptographic primitives is generated. Thus, if in one interval, there is a weakness in the dynamic key, this will not affect the previous or next ciphertext messages security. This limits the effects of weak dynamic session keys.

3.2.2 Key Sensitivity Test

This test aims at evaluating the secret key sensitivity against any slight change. In fact, the proposed key derivation function is based on a secret key and a cryptographic nonce. In this test, two dynamic keys are used: DK_1 and DK_2 that differ in only one random bit. Next, the plain-message (a) of Figure 4 is encrypted using these keys. To evaluate the difference between the obtained cipher-messages using DK_1 and DK_2, the Hamming distance of the corresponding encrypted cipher-messages C_1 and C_2 is computed and illustrated in Figure 6-(c) for 1,000 iterations, each time with different DK_1 and DK_2. The Hamming distance between these two cipher-messages is computed as follows:

$$KS = \frac{\sum_{k=1}^{T} C_1 \oplus C_2}{T} \times 100\% \qquad (5)$$

$$= \frac{\sum_{k=1}^{T} (E_{DK_1}(I)) \oplus (E_{DK_2}(I))}{T} \times 100\%$$

Where, T is the bits length of the plain and encrypted messages. It is clear from Figure 6-c that the majority of difference percentage values are close to the optimal value (50 %). These results indicate that the proposed encryption algorithm is robust and can withstand any adjustment(s) in the secret key or nonce. Thus, the proposed cipher can resist key-related attacks.

3.3 Resistance against Linear and Differential Attacks

The plain-text/ciphertext sensitivity test aim at evaluating the ability of cipher scheme to resist against linear and differential attacks (chosen/known plaintext/ciphertext attacks). In other words, this test demonstrates the level of sensitivity of the proposed cipher against any variation(s) on the plain-block message. Hence, the following scenario is realized: First, two plain blocks B_1 and B_2, which have only one bit difference, are encrypted separately to produce two cipher blocks C_1 and C_2. Then, the Hamming distance between these two cipher-blocks is computed as shown in Eq 5.

This test is iterated for $1,000$ random plain-blocks as shown in Figure 6-(b). The obtained mean value is close to 50%, which means that more than 50% of the corresponding cipher-block changes. Therefore, the proposed approach exhibits a high block sensitivity against any change(s) on the plain-block, producing a totally different encrypted block. Therefore, ensuring the avalanche effect in a dynamic pseudo-random manner helps in resisting linear and differential attacks.

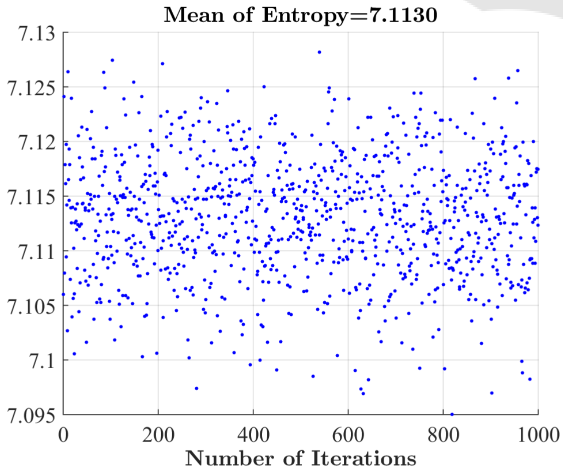


Figure 5: Entropy analysis of encrypted messages (each 256 bytes) using 1,000 random dynamic keys.

3.4 Discussion and Cryptanalysis

The proposed cipher approach ensures both the diffusion and confusion properties. Moreover, it satisfies the randomness and uniformity statistical characteristics, which guarantees immunity against statistical attacks.

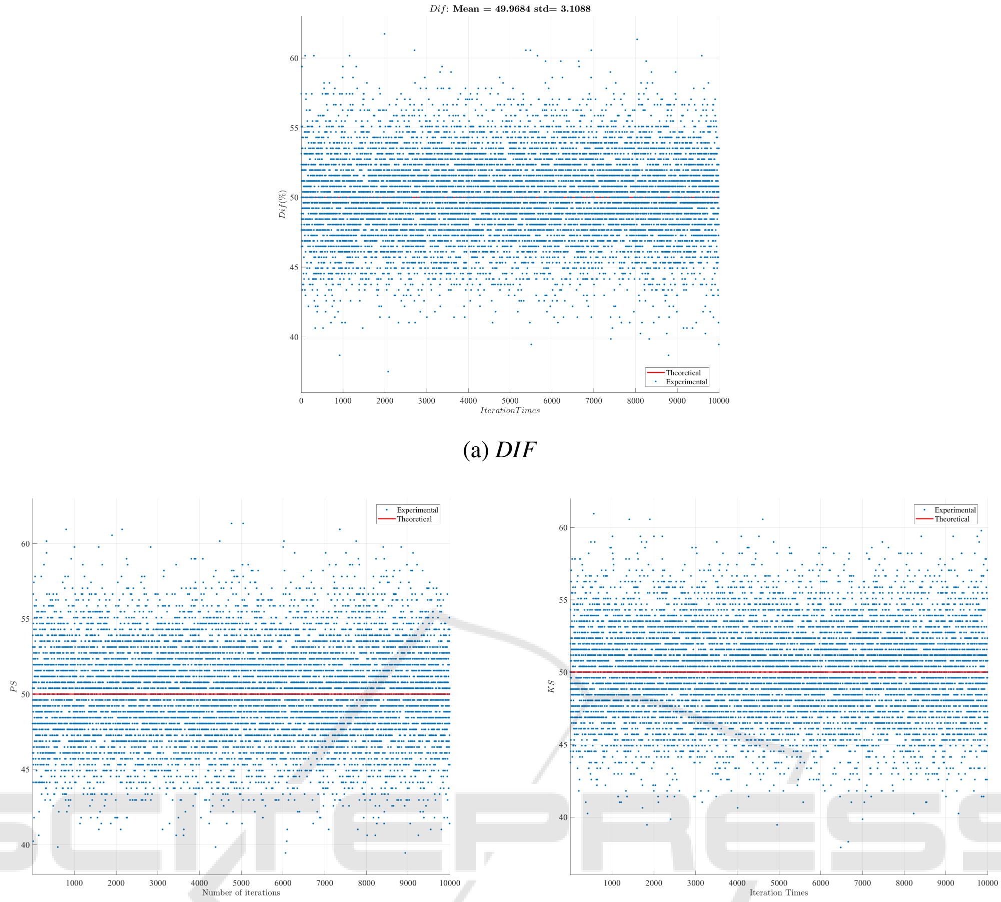
Furthermore, the proposed cipher approach renders differential and linear attacks ineffective and infeasible since the avalanche effect at the block level is attained along with a high key sensitivity, in a dynamic manner. In fact, any change(s) in any bit of the secret key or Nonce can cause a significant difference in the encrypted messages as seen in Figure 6. Moreover, the key space of the secret key can be 2^{128}, 2^{256} or 2^{512}, which is sufficiently large to render brute-force attacks infeasible. Additionally, the key space of the dynamic key is 2^{512}, which can also be considered large enough to overcome brute-force attacks. Hence, a secret key and a dynamic key are employed in the proposed cipher approach to make the cipher-text-only attack impossible and there is no way to retrieve any useful information from the encrypted messages. Therefore, the proposed cipher can guard against any cipher-text attack. Besides, the use of a dynamic key-dependent cryptographic primitives limits the ability of the attackers to break the proposed cipher scheme, especially when conducting side channel attacks.

4 PERFORMANCE ANALYSIS: COMPUTATIONAL DELAY

In this section, the computational delay of the proposed cipher is evaluated. The main objective of the proposed cipher approach is to achieve a high level of security with the minimum number of operations and round iterations to reduce the computational complexity and consequently, the latency and resource requirements.

To asses the total associated delay of the proposed scheme, we define the following delay components:

1. T_S denotes the required substitution execution time for a block of h bytes.
2. T_D denotes the required diffusion execution time for a block of h bytes.
3. T_{xor} denotes the required logical ”exclusive or” execution time between two blocks of h bytes.
4. T_{add} denotes the required arithmetic ”addition modulo 256” execution time between two blocks of h bytes.



(a) *DIF*

(b) *PS*

(c) *KS*

Figure 6: Independence tests (a), plain-block sensitivity (b) and key sensitivity (c) against 1,000 random keys for the proposed cipher.

5. T_π denotes the required time to permute an input block.

Therefore, the total Computational Delay (*CD*) of the proposed scheme to encrypt two blocks is:

$$CD_{D-ECB} = 2 \times T_S + 2 \times T_D + \times T_{xor} + \times T_{add} + T_\pi \quad (6)$$

while the total computation delay of the standard AES in (Daemen and Rijmen, 2013) to encrypt one block is:

$$CD_{AES} = rT_S + (r+1)T_{xor} + (r-1)T_D + rT_{SR} \quad (7)$$

where T_D represents the required delay for the AES diffusion mix-column operations (for all 4 columns), representing the highest delay compared to the other AES operations. T_{SR} represents the required delay for the AES permutation "Shift-rows" operation and r represents the number of rounds. The minimum value of r is 10 for 128 bits secret key. Hence, the minimum AES computation delay is given by:

$$CD_{AES(r=10)} = 10T_S + 11T_{xor} + 9T_D + 10T_{SR} \quad (8)$$

Consequently, the AES computational delay is larger compared to the proposed one. Moreover, the proposed solution uses an optimized binary diffusion operations minimize the computational complexity compared to mix-columns of AES, which reduces further the required diffusion delay.

Accordingly, the proposed scheme requires less computational complexity compared to the AES standard cipher for 128-bit length secret key.

5 CONCLUSION

In this paper, a novel cipher scheme is presented and analyzed based on different criteria such as cryptographic robustness and performance. The obtained results show that the proposed solution provides a high level of security with low resources and latency requirements. This is achieved due to the low required number of operations and rounds. In addition, the proposed cipher scheme is based on a combination of static (that can ensure maximum cryptographic performance) and dynamic cryptographic primitives. Additionally, all of its associated operations can be realized in parallel. The diffusion operation is flexible and depends on the block size, which can be chosen according to the device limitations. Moreover, its permutation operation is performed at the block level and not at the byte level, which reduces the overhead of permutation in terms of memory consumption and latency.

REFERENCES

Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J., and Tokita, T. (2000). Camellia: A 128-bit block cipher suitable for multiple platforms—design andanalysis. In *International Workshop on Selected Areas in Cryptography*, pages 39–56. Springer.

Daemen, J. and Rijmen, V. (2013). *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media.

Koo, B., Jang, H., and Song, J. (2006). On constructing of a 32× 32 binary matrix as a diffusion layer for a 256-bit block cipher. *Information Security and Cryptology–ICISC 2006*, pages 51–64.

Koo, B. W., Jang, H. S., and Song, J. H. (2003). Constructing and cryptanalysis of a 16× 16 binary matrix as a diffusion layer. In *International Workshop on Information Security Applications*, pages 489–503. Springer.

McKay, K. A., Bassham, L., Turan, M. S., and Mouha, N. (2016). Report on lightweight cryptography. *NIST DRAFT NISTIR*, 8114.

Melki, R., Noura, H. N., Mansour, M. M., and Chehab, A. (2018). An efficient ofdm-based encryption scheme using a dynamic key approach. *IEEE Internet of Things Journal*.

Noura, H., Chehab, A., Sleem, L., Noura, M., Couturier, R., and Mansour, M. M. (2018). One round cipher algorithm for multimedia iot devices. *Multimedia tools and applications*, 77(14):18383–18413.

Noura, H., Sleem, L., Noura, M., Mansour, M. M., Chehab, A., and Couturier, R. (2017). A new efficient lightweight and secure image cipher scheme. *Multimedia Tools and Applications*.

Noura, H. N., Chehab, A., and Couturier, R. (2019). Efficient & secure cipher scheme with dynamic key-dependent mode of operation. *Signal Processing: Image Communication*, 78:448–464.

Poschmann, A. Y. (2009). Lightweight cryptography: cryptographic engineering for a pervasive world. In *PH. D. THESIS*. Citeseer.