Lightweight Format-Compliant Encryption Algorithm for JPEG 2000 Images

Zeinab Fawaz, Hassan Noura, Ahmed Mostefaoui

Lightweight Format-Compliant Encryption Algorithm for JPEG 2000 Images

Z. Fawaz
FEMTO-St Institute DISC dep.
University of Franche Comte
Belfort, 90000, France
Email: Zeinab.fawaz@gmail.com

H. Noura
Faculty of Engineering
Lebanese University, Hadath Campus, Lebanon.
Email: hnnoura@gmail.com

A.Mostefaoui
FEMTO-St Institute DISC dep.
University of Franche Comte
Belfort, 90000, France
Email: ahmed.mostefaoui@univ-fcomte.fr

***Abstract*—The increasing usage of personal multimedia devices such as mobiles phones, smart glasses, etc. has pointed out the need for securing the captured/exchanged images content. Nevertheless, because of the limited resources of these novel platforms on one hand and the voluminous nature of multimedia content on the other hand, preserving multimedia content security remains a research challenge. In this paper, we tackle this issue by presenting a fast format-compliant selective encryption algorithm for JPEG 2000 images. It is based on selectively choosing data from the JPEG 2000 code-stream in a uniformly dynamic-key dependent manner to apply the proposed encryption algorithm. The encryption algorithm consists of two rounds of substitution-diffusion processes, based on a dynamic key, *that is changed for every input image*. Extensive security analysis has been conducted to evaluate the effectiveness of the proposed scheme. The obtained results have demonstrated the robustness of our algorithm against the most known types of attacks and have shown a significant improvement in term of execution time reduction compared to a similar existing JPEG 2000 images encryption scheme.**

***keywords:* JPEG 2000 compression standard; Format Compliant; selective encryption; substitution; diffusion; security analysis.**

I. Introduction

Nowadays, the widespread usage of multimedia devices such as smart-phones, image sensors, smart glasses, etc. has contributed in the production of billions of images that are daily exchanged throughout social networks and/or dedicated networks (e.g., Instagram). Furthermore, a large part of this huge amount of images is stored for later use either locally (personal devices) or transferred to cloud based storage systems. The management of such a huge amount of data raises new interesting research issues as efficient processing and storing, fast and friendly access, securing and protecting their content, etc. Within this work, we are particularly focusing on the latter issue, namely securing the content of exchanged/stored images whilst taking into account the hard constraints of the underlying platforms. This issue has in fact been pointed out recently through realistic examples such as the one related to celebrity photo hacking scandal. Another example can be found in Wireless Multimedia Sensor Networks (WMSNs) when the monitored area is very sensitive. In this category of applications, all sent images to the base station must be secure. Many other real examples can be cited in several domains as elder persons monitoring, medical applications, military applications, etc.

The common feature between all these applications is the usage of tiny multimedia devices, characterized by limited resources (CPU, memory and energy). Hence, securing image content within these devices require researchers to take into account the following, sometimes orthogonal, constraints: (a) multimedia data is very voluminous by nature in comparison to scalar data. Consequently, managing raw data is either not possible due to hard constraints in resources (i.e., memory limitation) or will break down all the devices resources (i.e., energy). Compression techniques are used in order to reduce the data volume. Even though after compression, the data remains voluminous. (b) The inherent features of images such as high redundancy and high pixels correlation render the traditional cryptographic mechanisms (i.e., AES, DES) not effective [1]. (c) Novel applications express the need of fast and lightweight cryptographic approaches that are able to provide sufficient security strength against most types of known attacks on one hand and that can be supported by these limited devices on the other hand. (d) For portability reasons between different platforms, the proposed approaches must be compression standard-Compliant (i.e., be able to be integrated into standard platforms without additional software installations).

In order to reduce the amount of transmitted data, especially in limited environments, many compression standards have been proposed [2], among them we note the standard JPEG 2000 [3] that provides better performance compared to other standards due to its main features such as: (a) low bit-rate performance, (b) lossless and lossy compression (c) random code-stream access and processing which allow devices to support some in-network processing (i.e data aggregation in the code-stream) and (d) robustness to bit-errors which guarantees a safe data transmission in wireless environments. These features motivate us to consider this compression standard within our work.

To secure compressed images, two main categories of approaches have been explored: (a) Selective Encryption approaches [4], where the encryption process is applied to a selective part of compressed data which is considered as important and (b) Joint Compression-Encryption approaches [5],

[6], where data encryption and compression are fulfilled into a single one step. Approaches in the first category must fulfill the **standard format-compliant**; i.e., the decoder is able to decode the encrypted data and to visualize the content. In other words, the decoder does not crash when decoding encrypted images. This important feature is fundamental to ensure the portability of the proposed encryption approaches. For this reason, we considered this feature as a requirement in our work.

In this paper, we propose a fast and format-compliant selective encryption algorithm for JPEG 2000 images. The proposed selective encryption scheme is applied to JPEG 2000 code-stream and requires two rounds of substitution-diffusion processes. To enforce its robustness, the used key is *dynamic* and changes for every input image. In the substitution process, bytes positions are changed non-linearly in order to effectively crack the correlation between adjacent bytes of the encrypted code-stream. While, in the diffusion process, bytes are changed sequentially (i.e., linear transformation) such that a small change in one byte can affect almost all bytes of the code-stream. The proposed approach is based on selectively choosing an amount of only 4% of data from each JPEG 2000 packet to follow the encryption process.

The main contributions of our work can be summarized as follows:

- High level of security: The proposed cipher algorithm fulfills the main security requirements (key sensitivity, randomness and uniformity) with only two rounds of encryption, rather than using multi encryption rounds as in the case in other encryption algorithms [7], [8].
- High data reduction: An amount of 4% of data is selected from each JPEG 2000 packet to contribute in the encryption process. This amount is sufficient to achieve a high security level compared to other approaches as [9], where an amount of 20% of data is needed to attain the required security level. By that, the proposed algorithm preserves significantly the communication bandwidth.
- Format-compliant property: Our proposed encryption scheme respects in its intrinsic construction the format compliant property.
- Fast Encryption: The use of our proposed approach with only two rounds of substitution-diffusion processes allows to achieve a fast encryption speed compared to [10], where AES in counter mode (CTR) is used to achieve the encryption process.

The rest of this paper is organized as follows. Section II provides the related work. Section III discusses the proposed encryption algorithm. Extensive security analysis is provided in Section IV. Additionally, compression analysis is investigated in Section V. The performance of the proposed algorithm is term of execution time is studied in Section VI. Finally, Section VII concludes the paper.

II. Related Work

Several works have been proposed for JPEG 2000 encryption schemes. According to [11], the position where the encryption is introduced can be used as a primary parameter to classify the JPEG 2000 encryption approaches. This classification produces three main schemes: (1) transform-based schemes, (2) coding-based schemes and (3) package-based schemes. In transform-based schemes, the encryption is relied on the use of a secret wavelet transform to provide lightweight security as in [12], [13]. Unfortunately, many of the transform-based schemes are found to be inefficient and insecure [14]. In coding-based schemes, the encryption and the entropy coding are fulfilled by one step as in [15], [16]. Algorithms from this class necessitate a modification of both encoder and decoder which may require additional compression overhead.

In the package-based schemes, the encryption process is performed on the code-stream, which consists of multiple packets (i.e., each packets consists of one packet header and one packet body). Many approaches have been proposed under this class as in [17], [18]. The main objective of these approaches is to achieve the encryption while respecting the format-compliant property. The mostly related work to this paper is Massoudi et al. approach [10], where a selective image encryption has been proposed based on selectively choosing an amount of 5.43% from each packet data. Then, bytes are encrypted using the standard AES-128 block cipher [7] with a modified CTR mode of operation (using additional modular operation instead of XOR operation), in order to achieve the format-compliant property and ensure the full confidentiality.

Instead of using the AES block cipher with multi-rounds, our proposed encryption algorithm achieves in its intrinsic construction (substitution and diffusion processes) the format-compliant criterion using only two rounds of encryption within a significantly shorter execution time and by selectively encrypting a small amount of data. Additionally, since the format-compliant property is ensured by the algorithm itself, then it can be used with any mode of operation without introducing any additional overhead.

III. The Proposed Encryption scheme

In this section, we present the proposed cipher algorithm. First, we introduce the main elements of our algorithm. Then after, we discuss the encryption/decryption fundamentals processes. In order to achieve a sufficient level of security while preserving the format-compliant property, our proposed encryption scheme is realized, with two rounds of encryption. In each round, two main processes are applied: (1) the substitution process and (2) the diffusion process.

Moreover, the proposed cipher is applied in counter mode of operation (CTR). This is motivated by the fact that: (a) the encryption process in CTR mode is applied to each block independently from other blocks, which makes the algorithm fully parallelizable and fast in hardware implementation (b) it allows the random access of encryption/decryption process, (c) the algorithm is simple in software and hardware implementations, since the decryption process is achieved similarly to the encryption without the need to reconstruct the inverse counterparts (d) and most importantly, CTR mode assures the robustness against error propagation [19].

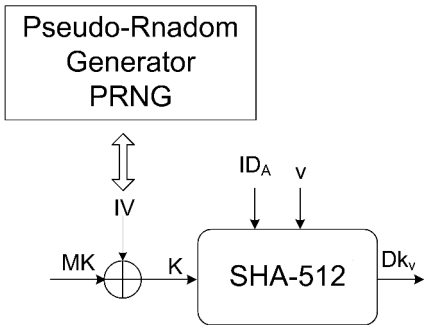

Fig. 1: The generation of the dynamic key DK_v

A. Preliminaries

The encryption/decryption process consists of two fundamental layers: substitution and diffusion layers. In this Section, we begin by explaining the generation of the dynamic key DK_v. Then, we provide a discussion about the main elements that constitute the substitution and the diffusion layers (non-linear S-box and the diffusion matrix G) which are derived from DK_v.

1) Dynamic Key Generation: First, the generation of the dynamic key follows the shared key schemes, where the two intelligible parts of communication (transmitter and receiver) share a secret key called *Master key (MK)*. This security mechanism exhibits low complexity compared to the public key schemes. Therefore, it can be well integrated when dealing with constrained devices.

The generation of one dynamic key DK_v (v is a counter that is incremented for every new image) from the shared key MK is illustrated in Figure 1 and achieved following these steps:

- First, MK is XORed with an Initialization Vector IV (128 bits) to produce an output K. In order to ensure the unpredictability of IV, a pseudo-random generator proposed in [20] is used to generate the IV sequence.
- Then, K is concatenated with ID_A (identity of the transmitter) and v. The concatenated form is hashed using SHA-512 hash function to produce the dynamic key DK_v (64 bytes).
- The dynamic key DK_v (64 bytes) is reshaped into a matrix $temp$ with size of 8×8 elements as follows:

$$temp = \begin{bmatrix} DK_v^1 & DK_v^2 & \cdots & DK_v^8 \\ DK_v^9 & DK_v^{10} & \cdots & DK_v^{16} \\ \vdots & \vdots & \vdots & \vdots \\ DK_v^{57} & DK_v^{58} & \cdots & DK_v^{64} \end{bmatrix} \quad (1)$$

This matrix contributes in the generation of the two main components: Substitution key (K_s) that is used later to construct the non-linear S-box of the substitution process and the diffusion matrix (G) for the diffusion process.

2) Substitution Box (S-box): First, K_s is realized by Xoring the elements of each column of $temp$, to produce $K_s = \{K_{s_1}, K_{s_2}, \ldots, K_{s_8}\}$. Then, two control parameters r and t are generated from K_s. To assure the bijectivity property, r is chosen to be the even components of K_s, while t corresponds to its odd components. After that, a nonlinear transformation f is iterated four times to produce the substitution S-box as follows:

$$L_i = f(L_{i-1}) = (L_{i-1} \times (r_i \times L_{i-1} + t_i)) \ mod \ 2^8 \quad (2)$$

Where, the first input $L_0[k] = k$, $(k = 0, 1, \ldots, 255)$. r_i, t_i are the corresponding control parameters for the i^{th} value $(i = 1, 2, 3, 4)$. Then, for each iteration, a bitwise right shift by 3 is applied on the result as expressed in this equation:

$$L_i = RightShift(L_i, 3). \quad (3)$$

S-box is equal to the output of Equation 3 after four iterations. In other words, $S = L_4$. After that, values corresponding to 255 are eliminated from the lookup table of S-box. Therefore, the produced S-box does not contain any element whose value is equal to 0xFF. Let's note that the generation of the inverse S-box is not required here, since the encryption is realized in CTR mode.

3) Diffusion matrix G: The diffusion matrix G deals with integer numbers instead of floating ones to avoid the complex floating operations. G matrix of size 4×4 is constructed based on M that is derived from the dynamic key components as expressed in the following equations:

$$M = \begin{bmatrix} DK_v^1 & DK_v^2 \\ DK_v^9 & DK_v^{10} \end{bmatrix} \quad (4)$$

$$G = \begin{bmatrix} M & M + I_m \\ M - I_m & M \end{bmatrix} \quad (5)$$

I_m is the identity matrix of size 2×2 and all elements in G matrix are belong to $\{0, 255\}$. I_m is expressed as follows:

$$I_m = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (6)$$

B. Encryption/Decryption Scheme

The cipher scheme deals with images after applying the JPEG 2000 compression. Hence, a packet is an elementary unit that constitutes the JPEG 2000 code-stream. It transports a compressed data format under certain resolution R, certain layer L, certain precinct P and certain component C. Indeed, the image can be viewed as a set of $R \times L \times P \times C$ packets, where the encryption process is applied to each packet P_j contained in the code-stream $P_j \in \{P_1, P_2, \ldots, P_t\}$ (i.e., all packets do not necessary have the same number of bytes and t is the total number of packets contained in one code-stream).

Then, an amount of 4% of bytes is selected from each packet P_j to follow the encryption process. Elements in the produced S-box (i.e., with size equal to 256) are used to specify the positions of bytes to be encrypted. The procedure of byte's selection is summarized in pseudo-code 1.

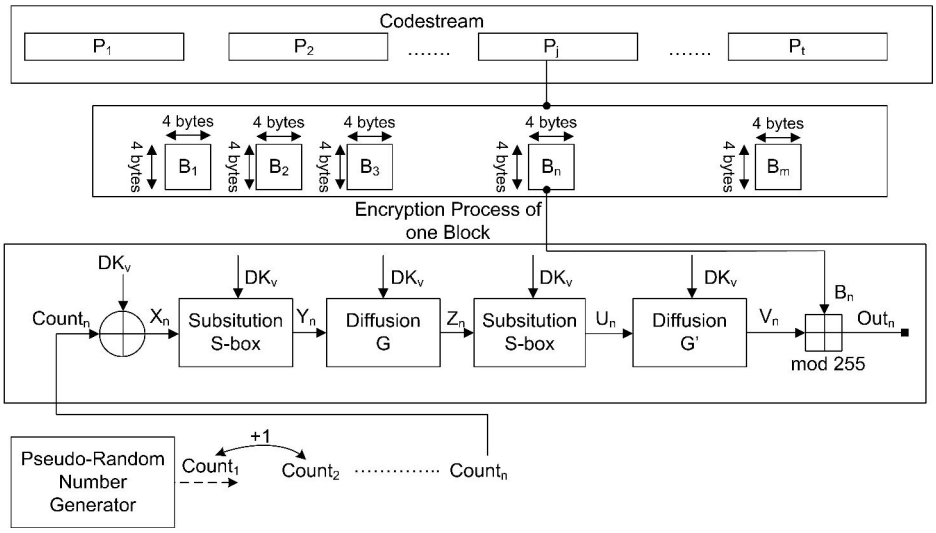

Fig. 2: The proposed selective encryption scheme for JPEG 2000 images

Algorithm 1 The selected byte's positions of each packet chosen in a dynamic manner according to the variable $Selected_{positions}$

Input The selected set of packets of a compressed JPEG 2000 code-stream, produced dynamic lookup table of S-box and diffusion matrix G.
perc=4/100;
for $j = 1 \rightarrow t$ **do**
  $l \leftarrow length(packets\{j\})$
  $Selected_{length} \leftarrow \lceil percent \times l \rceil$
  $w \leftarrow 1$
  if $(l \leq 256)$ **then**
    for $i \leftarrow 1$ **to** $Selected_{length}$ **do**
      if $S-box(i) \leq l$ **then**
        $Selected_{positions}(w) \leftarrow S-box(i)$
        $w \leftarrow w+1$
      end if
    end for
  else if $(l > 256)$ **then**
    $nb \leftarrow \lfloor l/255 \rfloor$
    for $co \leftarrow 1$ **to** nb **do**
      for $i \leftarrow 1$ **to** $\lceil Selected_{length}/nb \rceil$ **do**
        $Selected_{positions}(w) \leftarrow S-box\ (i)+(co*255)$
        $w \leftarrow w+1$
      end for
    end for
  end if
  $encryptData \leftarrow$ **Encr**($packets\{j,\ Selected_{positions}\}$, S-box, G)
  $packets\{j,\ Selected_{positions}\} \leftarrow encryptData$
end for
Output Encrypted selected packets of the code-stream.

Let's note that the percentage 4% is chosen in a way that a balance between the high visual degradation and the low computation complexity is fulfilled (see Section IV-E).

The selected data, corresponding to N_j number of bytes is then represented as blocks, each of 16 bytes. If N_j is not a multiple of 16, a padding with 0's is performed to complete the block elements. The number of blocks in one packet can be represented as $\{B_1, B_2, \ldots, B_m\}$, where m depends directly on N_j (i.e., if the packet P_1 consists of 500 bytes, only $N_1 = 20$ bytes of P_1 are selected to follow the encryption process, which are stored in two blocks: the first block consists of 16 bytes, while the second block consists of the four remaining bytes and twelve bytes that are padded). Therefore, the encryption process is performed in block-by-block manner using the following steps:

- First, a pseudo-random generator PRGN is used to generate a pseudo-random, unpredictable and nonce sequence denoted by $Count_1$. For each block encryption, this counter is incremented by one. Indeed, in order to encrypt the selected block B_n, the counter $Count_n$ (i.e., which is equal to $Count_1$+ (n-1)) is XORed with the dynamic key DK_v to produce the output X_n.
- Then, X_n follows the encryption process, beginning by the first substitution process through the use of the non linear S-box as follows:

$$Y_n = S(X_n) \quad n \in \{1, 2, \ldots, m\} \tag{7}$$

- Then, the diffusion process is applied on each substituted block Y_n by multiplying its components with the dynamic diffusion matrix G modulo 255, to prevent values of 0xFF to appear in the sequence. Thus, it produces an output denoted by Z_n, as expressed in the following Equation:

$$\begin{bmatrix} z_{n_{1,1}} \cdots z_{n_{1,4}} \\ z_{n_{2,1}} \cdots z_{n_{2,4}} \\ \vdots \ \ddots \ \vdots \\ z_{n_{4,1}} \cdots z_{n_{4,4}} \end{bmatrix} = \begin{bmatrix} g_{1,1} \cdots g_{1,4} \\ g_{2,1} \cdots g_{2,4} \\ \vdots \ \ddots \ \vdots \\ g_{4,1} \cdots g_{4,4} \end{bmatrix} \cdot \begin{bmatrix} y_{n_{1,1}} \cdots y_{n_{1,4}} \\ y_{n_{2,1}} \cdots y_{n_{2,4}} \\ \vdots \ \ddots \ \vdots \\ y_{n_{4,1}} \cdots y_{n_{4,4}} \end{bmatrix} \mod 255$$

- After that, the output block Z_n is used as an input to the second substitution process to produce a new block U_n, as expressed in the following equation:

$$U_n = S(Z_n) \quad n \in \{1, 2, \ldots, m\} \tag{8}$$

- Then after, the second diffusion process is applied to the substituted block U_n. In this step, elements of U_n are multiplied by G' matrix (the transpose of G matrix), to produce the corresponding block V_n as follows:

$$\begin{bmatrix} v_{n_{1,1}} \cdots v_{n_{1,4}} \\ v_{n_{2,1}} \cdots v_{n_{2,4}} \\ \vdots \ \ddots \ \vdots \\ v_{n_{4,1}} \cdots v_{n_{4,4}} \end{bmatrix} = \begin{bmatrix} g'_{1,1} \cdots g'_{1,4} \\ g'_{2,1} \cdots g'_{2,4} \\ \vdots \ \ddots \ \vdots \\ g'_{4,1} \cdots g'_{4,4} \end{bmatrix} \cdot \begin{bmatrix} u_{n_{1,1}} \cdots u_{n_{1,4}} \\ u_{n_{2,1}} \cdots u_{n_{2,4}} \\ \vdots \ \ddots \ \vdots \\ u_{n_{4,1}} \cdots u_{n_{4,4}} \end{bmatrix} \mod 255$$

- As final step, the resultant output block V_n is XORed (modulo 255) with their corresponding initial block B_n to produce at the end the cipher block C_n as follows:

$$C_n = (V_n + B_n) \ mod \ 255 \quad n \in \{1, 2, \ldots, m\} \tag{9}$$

- Finally, after encrypting all corresponding blocks of one packet P_j, only N_j bytes are taken sequentially from the successive encrypted blocks and located back to their initial positions in P_j.

By these two rounds, the encryption scheme is completed and the code-stream becomes ready to be transmitted to the receiver. Let's note that, since we are dealing with CTR mode, the decryption process is realized similarly to the encryption one, using the same substitution S-box as well as the same diffusion G matrix. Also, due to the fact that the same S-box is produced, the receiver succeeds to retrieve the positions of the encrypted bytes from each packet of the code-stream to apply the decryption process. The only change is that in Equation 9, the addition operation is replaced by a subtraction operation as follows:

$$B_n = (C_n - V_n) \ mod \ 255 \quad n \in \{1, 2, \ldots, m\} \tag{10}$$

IV. Security Analysis

In this section, several security tests are conducted to evaluate the efficiency and the cryptographic performance of the proposed encryption algorithm. The three standards images: Lena, Pepper and Baboon, each of size $512 \times 512 \times 3$ are taken as source images to perform our experiments. In some security tests, only results for Lena image are represented due to the space limitation. Additionally, security analysis are performed under the following software and hardware environments: GCC, micro-computer Intel Core i7, 5600U CPU at 2.6 GHz with 16 GB RAM Intel, Windows 7, MATLAB R2014a framework and OpenJPEG codec. The JPEG 2000 compression is performed using the standard lossy mode with rate 0.25, 4 tiles, 3 resolutions, one quality layer, one precinct and performed in LRCP progressive mode.

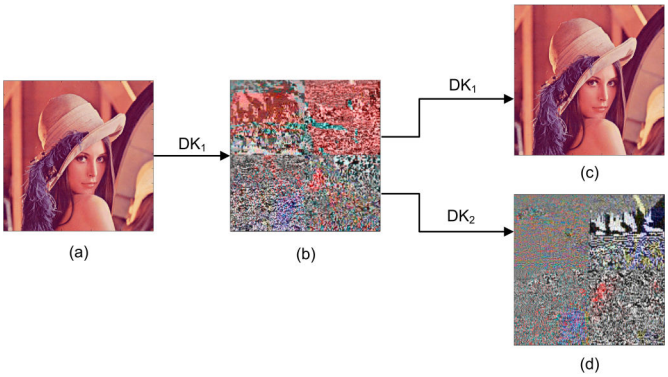

Fig. 3: (a) Lena plain image, (b) Encrypted Lena image using DK_1, (c),(d) Decrypted Lena image using DK_1 and DK_2 respectively (DK_2 differs from DK_1 with only one-bit).

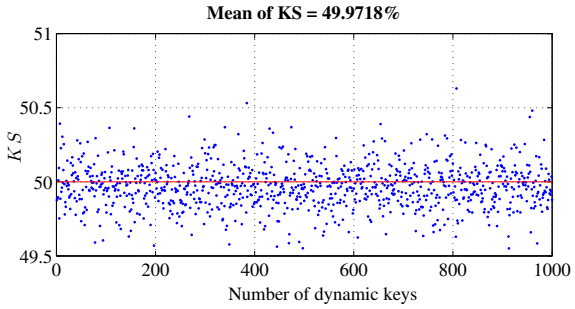

Fig. 4: The key sensitivity test using 1000 different dynamic keys.

A. Key Sensitivity

Key sensitivity is of primary importance to resist to chosen-plaintext and linear attacks. Thus, it means that the cipher is highly depends on the key. In other words, one bit change on the key must produce a totally different cipher-text. To test the sensitivity of the proposed encryption scheme, the following scenario is performed: First, a dynamic key DK_1 is used to perform the encryption algorithm on the compressed Lena image. The decryption process succeeds to recover the compressed image as illustrated in Figure 3-(c). On the other side, a modified dynamic key DK_2 with one bit difference from DK_1 is used to decrypt the same image. As shown in Figure 3-(d), with a little change on the dynamic key, the decryption process fails to reconstruct the original Lena image.

To further demonstrate the key sensitivity, the above scenario is repeated using 1000 randomly selected keys. First, the compressed Lena image followed the selective encryption

process using the correct dynamic key DK_v to produce the corresponding cipher code-stream C. The reconstruction of the original compressed Lena image L succeeds using DK_v. On the other part, C is decrypted using 1000 different dynamic keys, that have only one-bit difference (Least Significant Bit LSB) to produce, at each iteration, a new image denoted by L'. After that, the hamming distance (i.e. the difference in bits) between L and L' is computed as expressed in Equation 11 and illustrated in Figure 4,respectively.

$$KS = \frac{\sum_{k=1}^{T} L \oplus L'}{T} \times 100\% \tag{11}$$

$$= \frac{\sum_{k=1}^{T} D_{DK_v}(C) \oplus D_{DK'_v}(C)}{T} \times 100\%$$

where T is the length in bits of the plain-text and cipher-text images.

Results show that the majority of values are close to the optimal value (50 %) [8], which indicates that the proposed selective encryption algorithm has a great sensitivity against any little change on the dynamic key.

B. Key Space analysis

Key space represents the total number of different keys that can be used in the encryption process. In order to ensure the robustness of the encryption algorithm against the brute-force attacks, the size of key space must not be smaller than 2^{128} [21]. For this reason, the initialization vector IV, the counter sequence as well as the dynamic key DK_v used in the proposed approach, each consists of 128-bits. Indeed, the proposed encryption algorithm has 2^{128} large key space. Thus, it makes the brute-force attacks unfeasible.

C. Histogram analysis

In statistical analysis, histogram is used to display the frequency of pixel values. If the frequency counts of the encrypted data behaves as a uniform distribution, then pixel values are effectively masked.

In Fig. 5, histograms of the original Lena, Peppers and Baboon images and their corresponding cipher ones are illustrated. Results show that the histograms of the JPEG-2000 encrypted data are uniformly distributed, which are very different compared to those of the original images. Hence, the proposed encryption algorithm significantly eliminates the statistical information of the original image and supports a high strength against statistical attacks.

D. Correlation between Original and Cipher images

Images are characterized by the high correlation between its adjacent pixels (close to 1). An encryption algorithm is said to be efficient, if it removes the spatial redundancy and hides all the attributes of the plain-image, to provide at the end a cipher image that does not support any information about the original image. Indeed after encryption, the correlation between the original and the encrypted images must be negligible (close to zero). To study the correlation effect of the proposed encryption algorithm, the correlation coefficient r between

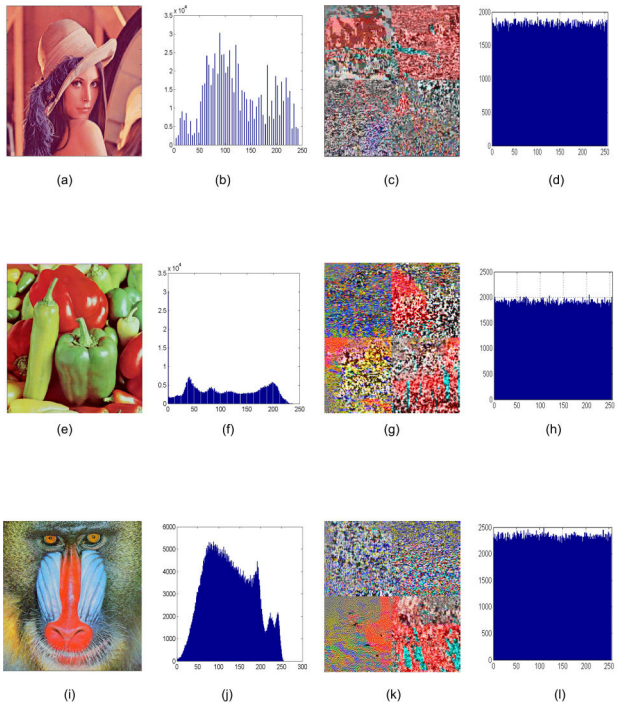

Fig. 5: (a), (e), (i) Original Lena, Peppers and Baboon images. (b), (f), (j) Histogram of Lena, Pepper and Baboon images. (c), (g), (k) Cipher Lena, Pepper and Baboon images. (d), (h), (l) Histogram of cipher Lena, Pepper and Baboon images.

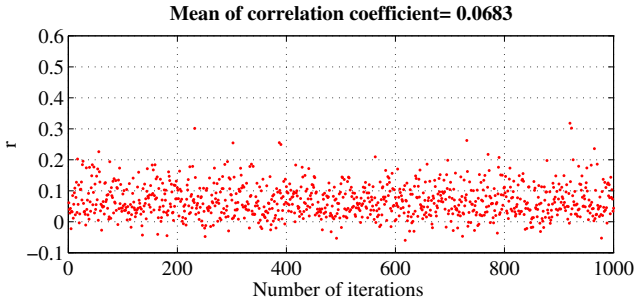

Fig. 6: Correlation analysis between Plain Lena image and its corresponding JPEG-2000 encrypted images using 1000 different keys.

pixels of the original Lena image of size $S = 512 \times 512 \times 3$ and its corresponding JPEG-2000 encrypted image is measured after applying the proposed algorithm using 1000 different random keys. Results in Figure 6 show that pixels of the plain Lena image have negligible correlation with the pixels of the cipher image (mean equal to 0.0683). Thus, it demonstrates that the proposed encryption scheme succeeds to crack the correlation between pixels and have a high strength against statistical attacks.

Image Name	Image Size in pixels	Image Size in JPEG 2000	Execution Time (sec) Compression	Execution Time (sec) Proposed	Execution Time (sec) Massoudi et al.
Lena	$256 \times 256 \times 3$	8 KB	0.0433	0.0325	0.2216
Lena	$512 \times 512 \times 3$	31 KB	0.1825	0.0749	0.9150
Lena	$1024 \times 1024 \times 3$	128 KB	0.4568	0.2533	3.4668
Peppers	$256 \times 256 \times 3$	8 KB	0.0499	0.0300	0.2890
Peppers	$512 \times 512 \times 3$	31 KB	0.1999	0.0861	0.9676
Peppers	$1024 \times 1024 \times 3$	128 KB	0.5816	0.2309	3.6580
Baboon	$256 \times 256 \times 3$	8 KB	0.0561	0.0276	0.2098
Baboon	$512 \times 512 \times 3$	31 KB	0.2177	0.0727	0.9223
Baboon	$1024 \times 1024 \times 3$	123 KB	0.5084	0.1911	3.4398

TABLE I: Performance of the proposed Approach and Massoudi et al. Approach for the three standards images: Lena, Peppers and Baboon with different sizes.

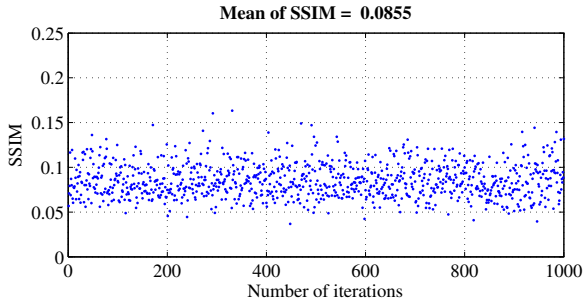

Fig. 7: SSIM index between original and JPEG-2000 encrypted Lena image for 1000 different random keys.

E. Visual Degradation

This metric measures the degradation operated on the original image using the encryption algorithm in a way that its visual content, present in the cipher image, is no more comprehensible. In order to evaluate the visual degradation, an amount of 4% of bytes is chosen from each packet data of Lena code-stream to follow the encryption approach. This procedure is repeated using 1000 random keys to produce different cipher images. At each iteration, the Structural Similarity Index ($SSIM$) [22] is used to measure the level of similarity between the original image and its corresponding JPEG-2000 encrypted image and results are illustrated in Figure 7. $SSIM$ index ranges between 0 and 1. A value of 1 means that the two measured images are identical and a value close to 0 indicates that the two measured images are totally different. Results show that for all ciphered images, the $SSIM$ index does not exceed the value of 0.17, which means that a hard and sufficient visual distortion is achieved using the proposed approach. Additionally, the amount of 4% that is selected from each packet data of the code-stream to follow the encryption process is sufficient to achieve a good balance between hard visual distortion and low computational complexity.

V. Compression Analysis

Besides ensuring a high security level, the proposed selective encryption scheme must be compression friendly. For this reason, two main metrics related to the compression aspect are evaluated in this section, including (1) the Code-stream Compliant analysis as well as (2) the Compression Friendliness evaluation.

A. Code-stream Compliant analysis

Ensuring the format-compliant property is of paramount importance, especially when dealing with selective JPEG 2000 encryption scheme. Since, compliance allows preserving the main characteristics of the original compression coding. Thus, it enables the decoder to decode the encrypted code-stream before decryption. In the proposed selective encryption scheme, both substitution and diffusion processes in their intrinsic structures are format-compliant. First, in the substitution process, any value that is corresponding to 0xFF is eliminated from the look-up table. Additionally, in the diffusion process, the matrix multiplication is provided with addition modulo 255. Indeed, preventing 0xFF marker to appear in the code-stream, means that both code-words 0xFF90 and 0xFFFF will not appear in the packet body. Therefore, the encrypted code-stream is compliant to the format of the original unencrypted code-stream and preserves all its characteristics.

B. Compression Friendliness

Most joint compression-encryption algorithms decrease the compression performance, since the encryption is performed before the quantization process or within the encoding process. However, using the code-stream-oriented encryption schemes, data modification is applied to the code-stream. Thus, it has no influence on the compression performance. Additionally, the proposed encryption method preserves the format-compliant property. By that, the encrypted image is compliant to the format structure of the unencrypted image. In order to fulfill the encryption algorithm, such operations are based on addition modulo, which neither add nor subtract a bit from the code-stream. Therefore, the proposed algorithm does not affect the compression performance and satisfies the compression friendliness property.

VI. Performance Evaluation

In order to study the performance of the proposed algorithm, the execution time is evaluated due to the fact that less computational time means low computation complexity,

and consequently minimum resource requirements for the encryption/decryption processes. In this contexct, Colors Lena, Peppers and Baboon images, with three different sizes : $256 \times 256 \times 3$, $512 \times 512 \times 3$ and $1024 \times 1024 \times 3$ are used to demonstrate the practical performance. First, the compression time for transforming each image into JPEG 2000 format is measured. Then, the encryption time for our proposed encryption scheme as well as for Massoudi et al. approach [10] are tested as shown in Table I. Results of the encryption time show that the proposed algorithm is faster than Massoudi et al. algorithm by approximately 8 times for small image sizes and by approximately 15.6354 times for large image sizes. Thus, it results to a highly increase in throughput.

VII. Conclusion

In this paper, we have proposed a secure format compliant lightweight cipher for protecting JPEG 2000 images. It is consisted of two rounds of: substitution and diffusion process. Format-compliant property is preserved intrinsically in the construction of both process, in order to preserve the characteristics of the original unencrypted code-stream.

Moreover, extensive experiments have been conducted to prove the high level of security and the robustness of the proposed algorithm against the most known types of attacks and its effectiveness in term of execution time compared to the approach proposed in [10]. Indeed, all these features consider our proposed JPEG 2000 images encryption scheme as a good candidate to deal efficiently with tiny constrained devices.

References

[1] Q. Zhang, L. Guo, and X. Wei, "Image encryption using dna addition combining with chaotic maps," *Mathematical and Computer Modelling*, vol. 52, no. 11, pp. 2028–2035, 2010.

[2] F. Dufaux, G. Sullivan, and T. Ebrahimi, "The jpeg xr image coding standard," *IEEE Signal Processing Magazine*, vol. 26, no. MMSPL-ARTICLE-2009-004, pp. 195–199, 2009.

[3] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The jpeg2000 still image coding system: an overview," *Consumer Electronics, IEEE Transactions on*, vol. 46, no. 4, pp. 1103–1127, 2000.

[4] H. Cheng and X. Li, "Partial encryption of compressed images and videos," *Signal Processing, IEEE Transactions on*, vol. 48, no. 8, pp. 2439–2451, 2000.

[5] M. Grangetto, E. Magli, and G. Olmo, "Multimedia selective encryption by means of randomized arithmetic coding," *Multimedia, IEEE Transactions on*, vol. 8, no. 5, pp. 905–917, 2006.

[6] J.-L. Liu, "Efficient selective encryption for jpeg 2000 images using private initial table," *Pattern Recognition*, vol. 39, no. 8, pp. 1509–1517, 2006.

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

[8] T. Xiang, J. Qu, and D. Xiao, "Joint spiht compression and selective encryption," *Applied Soft Computing*, vol. 21, pp. 159–170, 2014.

[9] R. Norcen and A. Uhl, "Selective encryption of the jpeg2000 bitstream," in *Communications and Multimedia Security. Advanced Techniques for Network and Data Protection*. Springer, 2003, pp. 194–204.

[10] A. Massoudi, F. Lefebvre, C. D. Vleeschouwer, and F.-O. Devaux, "Secure and low cost selective encryption for jpeg2000," in *Multimedia, 2008. ISM 2008. Tenth IEEE International Symposium on*. IEEE, 2008, pp. 31–38.

[11] T. Xiang, C. Yu, and F. Chen, "Secure mq coder: An efficient way to protect jpeg 2000 images in wireless multimedia sensor networks," *Signal Processing: Image Communication*, vol. 29, no. 9, pp. 1015–1027, 2014.

[12] T. Stütz and A. Uhl, "Efficient wavelet packet basis selection in jpeg2000," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 317–320.

[13] T. Stütz, B. Mühlbacher, and A. Uhl, "Best wavelet packet bases in a jpeg2000 rate-distortion sense: The impact of header data," in *Multimedia and Expo (ICME), 2010 IEEE International Conference on*. IEEE, 2010, pp. 19–24.

[14] D. Engel, T. Stütz, and A. Uhl, "Assessing jpeg2000 encryption with key-dependent wavelet packets," *EURASIP Journal on Information Security*, vol. 2012, no. 1, pp. 1–16, 2012.

[15] C.-H. Yuen and K.-W. Wong, "A chaos-based joint image compression and encryption scheme using dct and sha-1," *Applied Soft Computing*, vol. 11, no. 8, pp. 5092–5098, 2011.

[16] A. Pande, P. Mohapatra, and J. Zambreno, "Securing multimedia content using joint compression and encryption," *IEEE MultiMedia*, vol. 20, no. 4, pp. 50–61, 2013.

[17] Z. Brahimi, H. Bessalah, A. Tarabet, M. Kholladi *et al.*, "Selective encryption techniques of jpeg2000 codestream for medical images transmission," *WSEAS Transactions on Circuits and Systems*, vol. 7, no. 7, pp. 718–727, 2008.

[18] S. Lian and X. Chen, "On the design of partial encryption scheme for multimedia content," *Mathematical and Computer Modelling*, vol. 57, no. 11, pp. 2613–2624, 2013.

[19] K. Burda, "Error propagation in various cipher block modes," *IJCSNS*, vol. 6, no. 11, p. 235, 2006.

[20] M. Dworkin, "Recommendation for block cipher modes of operation. methods and techniques," DTIC Document, Tech. Rep., 2001.

[21] J.-S. Cho, S.-S. Yeo, and S. K. Kim, "Securing against brute-force attack: A hash-based rfid mutual authentication protocol using a secret value," *Computer Communications*, vol. 34, no. 3, pp. 391–397, 2011.

[22] C. Li and A. C. Bovik, "Content-partitioned structural similarity index for image quality assessment," *Signal Processing: Image Communication*, vol. 25, no. 7, pp. 517–526, 2010.