Noname manuscript No.
(will be inserted by the editor)

Securing JPEG-2000 Images in Constrained Environments : a Dynamic Approach

Zeinab Fawaz · Hassan Noura · Ahmed Mostefaoui

Received: date / Accepted: date

Abstract This paper presents an efficient and lightweight format-compliant selective encryption algorithm for secure JPEG 2000 coding. The proposed encryption scheme is dynamic in nature, where the key is changed for every input image. Furthermore, an amount of 4% of bytes from each packet data is selected to follow the encryption process. Moreover, in order to achieve the desired security, two rounds of substitution-diffusion processes are applied to the selected bytes. Experimental analysis have proved that this amount of encrypted data ensures a hard image distortion while significantly preserve the communication bandwidth. In addition, compression analysis and extensive security tests have demonstrated: (1) the robustness of the proposed selective encryption approach against the most known types of attacks, (2) the preservation of the main compression properties (i.e., compression friendliness and format-compliant), and most importantly, (3) the efficiency in term of execution time compared to others similar JPEG-2000 images encryption schemes.

Keywords JPEG 2000 compression standard; Format Compliant; lightweight selective encryption; Dynamic substitution; Dynamic diffusion; security analysis

Zeinab Fawaz and Ahmed Mostefaoui
FEMTO-ST Institute Disc dep
University of Bourgogne-Franche Comte
Belfort 9000, France
E-mail: zeinab.fawaz@univ-fcomte.fr
E-mail: ahmed.mostefaoui@univ-fcomte.fr

Hassan Noura
Faculty of Engineering,
Lebanese University,
Hadath Campus, Lebanon
E-mail: hnouran@gmail.com

1 Introduction

Recent technology advances have witnessed the development of miniaturized devices with communication capabilities and high resolution multimedia acquisition features such as: smart-phones, image sensors, smart-glasses, etc. In this context, billions of high resolution images of different scope and importance, from personal and local use, to commercial and governmental need, are being managed and stored either in cloud-based storage systems or exchanged through various online social networks (i.e., Facebook, Twitter, Instagram . . .). This huge data volume raises nevertheless new research challenges for their storage and transmission, especially in limited-resources devices platforms (e.g., wireless multimedia sensor networks, WMSNs). Indeed, devices in such platforms have usually limited computing capabilities to process multimedia content and a tight energy constraint, as they are driven by batteries. Furthermore, they almost operate in open infrastructure characterized by wireless unreliable and unsecured communications.

Our aim in this paper is to secure the transmission of multimedia content while taking into consideration the realistic constraints of the underlying platforms. More specifically, the proposed approach has to fulfill, in this context, four main fundamental criterion: (a) reducing the amount of transmitted multimedia data to save the communication bandwidth, (b) securing the transmitted content with the objective of ensuring data confidentiality, (c) being compression format compliant in order to facilitate the software integration and (d) exhibiting low complexity and fastest execution time to efficiently respond to the real time delivery on one hand and to be supported by limited devices on the other hand.

Several compression techniques have been proposed to reduce the volume of images such as JPEG, JPEG 2000 [1] and JPEG XR [2]. JPEG 2000 provides better performance compared to other standards due to its main features: (1) low bit-rate performance, (2) lossless and lossy compression (3) random code-stream access and processing which allow devices to support some in-network processing such (i.e data aggregation in the code-stream) and (4) robustness to bit-errors which guarantees a safe data transmission in wireless environments. For these reasons, we have focused in our work on the standard JPEG-2000.

In this paper, we propose an efficient and lightweight format-compliant selective encryption algorithm for JPEG 2000 images. The proposed approach is based on selective encryption scheme, where an amount of 4% of bytes are selectively chosen from each packet data of the code-stream in order to be encrypted. Referred to Shannon theory [3], substitution-diffusion operations are combined together in order to provide a secure cipher algorithm. For this reason, two rounds of substitution-diffusion processes are performed to fulfill a high level of security against the most known types of attacks. In the substitution process, the values of bytes are changed non-linearly in order to break the high correlation between adjacent bytes and overcome the statistical-attacks. On the other hand, in the diffusion process, the values of bytes are changed linearly. Moreover, these two process are dynamic, since the used key is **dynamic** in nature, and changed for every input image. The main contributions of our approach can be summarized as follows:

- **High level of security**: The proposed encryption scheme ensures a sufficient security level (high visual degradation, good randomness degree, etc.) and demonstrates its robustness against the most known types of attacks (statistical, differential, brute force and chosen-plaintext attacks).
- **Significant data reduction**: An amount of 4% of data is selected from each packet data to follow the encryption process. Experiments results have proved that this amount is sufficient to achieve a good balance between the high security level and the image size. By that, a great data reduction is fulfilled by our algorithm compared to [4], where an amount of 20% of data is used to attain the required security level and compared to [5], where all bytes of all packet data are encrypted. Hence, the proposed algorithm significantly preserves the communication bandwidth.
- **Format-Compliant property**: The used substitution and diffusion processes in the proposed encryption scheme preserve in its intrinsic construction the format-compliant property, in a way that the characteristics of the compressed image remain unchangeable after encryption. We recall that this property is very useful in large scale platforms in order to ensure software compatibility and facilitate the software components deployment.
- **Fast Encryption**: The use of our proposed approach with only two rounds of substitution-diffusion processes allow to achieve a fast encryption speed compared to [6], where AES in counter mode (CTR) is used to achieve the encryption process and [5], where the encryption is achieved in a byte-by-byte manner for the whole packet data of the code-stream.

The rest of this paper is organized as follows. Section 2 provides the related work. Section 3 gives a background about JPEG 2000 compression standard. Section 4 discusses the proposed encryption algorithm with its encryption/decryption process. Cryptographic study of both substitution and diffusion layers is investigated in Section 5. Then after, extensive security analysis and compression evaluation are provided in Sections 6, 7 and 8 respectively. Section 9 ends the paper.

2 Related Work

Recently, several reaserch works have addressed the selective JPEG-2000 encryption issue for images as well as for video coding [7]. Due to the space limitation, we only introduce them under a generalized classification and discuss the most close work in details. A comprehensive survey about JPEG 2000 encryption can be found in [8]. According to [9], JPEG-2000 encryption algorithms can be categorized, according to the position where the encryption is introduced, in three main schemes: (1) transform-based schemes, (2) coding-based schemes and (3) package-based schemes.

Firstly, in transform-based schemes, a lightweight security is ensured using a secret wavelet transform. In [10], an encryption scheme for JPEG 2000 based on randomly generated wavelet packet decomposition has been proposed. Also, in [11], an image coding scheme has been proposed, based on the multiple description lattice vector quantization (MDLVQ). Another lightweight encryption approach has been discussed in [12], where an Anistropic Wavelet Packets (AWP) is used in the compression process. Unfortunately, many of the transform-based schemes are found to be inefficient and insecure [13]. Also, we note that, applying the encryption before the entropy coding may reduce the efficiency of coding, since the statistics of input data is modified.

Secondly, in coding-based schemes, the encryption and the entropy coding are fulfilled in one step. In [14], an encryption approach based on a randomization of the arithmetic coder has been proposed. The encryption is achieved by randomly swapping the most probable symbol (MSP) and the least probable symbol (LSP) intervals. However, since arithmetic coding is context based, any error will propagate to subsequent contexts and adversely will impact probabilities computations. Another encryption scheme has been proposed in [15], where a private initial table is generated based on a secret key and a mapping function. Then, this table is used to encrypt the selected discrete wavelet transformed (DWT) code-blocks in the entropy coding stage of JPEG-2000 coding scheme. Both approaches do not evaluate the chosen/plaintext attack which is considered as an effective attacks in multimedia cryptanalysis.

Thirdly, in package-based schemes, the encryption process is performed on the code-stream, which consists of a packet header and a packet body. Many approaches have been proposed in this category. In [16], a packet-level syntax compliant encryption scheme has been proposed, where a pseudo-random sequence is generated (0xFF bytes values are discarded). Then, the encryption is performed by adding each byte of the packet data whose value is not 0xFF and whose preceding value is not 0xFF to the corresponding byte of pseudo-random generator modulo 0xFF. As result, the format-compliant is always ensured. Another encryption scheme has been presented in [17], where each code-block contribution to a packet (CCP) is encrypted with a modular addition or a block cipher is repeated until the cipher-text is syntax compliant. Additionally, another encryption algorithm is proposed in [18], where the payload of each packet is divided into equally blocks. Then, a pseudo-random generator is used to select one byte from each block. After that, the value of each byte is checked before encryption. If the byte is blow 0xFF, its lower half is selected; otherwise the byte is skipped. All selected bytes are stored in a buffer. Then, they follow the encryption process using conventional cipher. The importance of code-stream based schemes appear by having a negligible influence on the compression rate, since they do not access the compression pipeline.

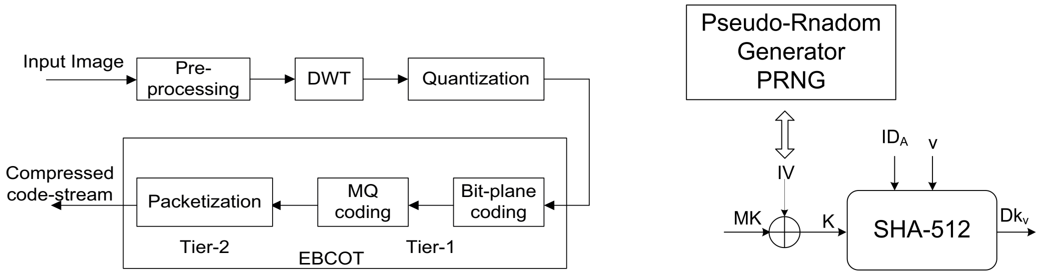
In [6], a selective image encryption has been presented based on the use of the standard AES-128 block cipher [19] in modified CTR mode of operation (using a conditional modular addition instead of XOR operation), in order to prevent the codewords within the interval [0xFF90, 0xFFFF] to be in the packet data. To achieve a full confidentiality, an amount of 5.43% of bytes are selected from each packet data to follow the encryption process. One positive feature of this approach is the use of CTR mode, which reduces the propagation of errors. However, the use of AES block cipher, with its multiple rounds renders this approach less efficient in term of execution time, especially when dealing with limited resources devices that require real-time delivery. The second approach is a chaotic-cipher based packet body encryption algorithm that has been proposed in [5]. In this approach, a piece-wise linear chaotic map (PWLCM) is used to generate the pseudo-random sequences. The encryption is performed on each 2-byte block of the packet body using a bitwise exclusive (OR) and some cyclic rotation operations in order to achieve a high randomness. Additionally, the format-compliant property is ensured by iterating the encryption process to prevent the undesirable marker to appear in the JPEG-2000 encrypted code-stream. Unfortunately, this approach preserves the feedback property, where the encryption of one block is dependent to the previous encrypted block. Thus, it makes the algorithm very prone to error propagation. Also, applying the encryption in all bytes of packet data (i.e huge amount of data to be encrypted) reduces the efficiency of this algorithm in terms of bandwidth and execution time.

In our proposed approach, instead of using the AES block cipher with multi-rounds, only two rounds of substitution-diffusion processes are preformed in CTR mode of operation. Both processes preserve in their intrinsic construction the format-compliant criterion. Additionally, the proposed approach significantly reduces the data bandwidth, by selecting only 4% of bytes from each packet data to follow the encryption process. Indeed, the proposed algorithm provides a sufficient security with high efficiency, especially in term of execution time compared to [6] and [5].

3 JPEG 2000 Compression Standard

JPEG 2000 is a still image compression standard created by Joint Photographic Experts Group Committee in 2000 [1]. It was designed to replace the widely used JPEG standard. The compression process in JPEG 2000 consists of four steps: (1) pre-processing, (2) Discrete Wavelet Transform (DWT) [20], (3) quantization and (4) block encoding using the Embedded Block Coding with Optimal Truncation (EBCOT) [21]. Figure 1-(a) represents the block diagram of JPEG-2000 coding procedure.

In the pre-processing step, the image is partitioned into rectangular and non-overlapping tiles of equally size. Each tile is compressed independently using its own set of specified compression parameters. Also, a color transformation is performed on the RGB image to



(a) (b)

Fig. 1: (a) JPEG 200 syntax format and (b) The generation of the dynamic key DK_v.

transform the R (Red), G (Green) and B (Blue) components into Y (Luminance), C_r (Chrominance) and C_b (Chrominance) components. Then, each tile of the image is wavelet transformed using the DWT which decomposes the image into high and low subbands. After that, in the quantization step, the DWT coefficients are quantized using dead-zone quantization, to reduce the precision of data and to make them more comprehensible. Before performing the coding process, the subbands of each tile are further divided into small code-blocks (32 × 32 or 64 × 64 block sizes). Then, each code-block is encoded independently using EBCOT in a bit-plane basis, to generate different code-streams. The coding procedure by the EBCOT involves two main stages: Tier-1 coding that essentially involves the entropy encoding technique and Tier-(2) that efficiently represents the layer and the block summary information of each code-block.

JPEG 2000 offers a flexible code-stream and behaves in four different directions: Quality (Q), Component (C), Resolution (R) and Precinct (P) (i.e another form of partitioning applied on the DWT coefficients by grouping the code-blocks belonging to the same spatial region). All these quality layers are ordered through the so called **packets**.

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. Hence, the total number of packets can be represented as $R \times L \times P \times C$.

In order to ensure format-compliance, the syntax of code-stream requires that the packets carry the content bit-streams whose code-words (i.e., two contiguous bytes) are not in the interval [0xFF90, 0xFFFF] [6].

4 Proposed Encryption/Decryption scheme

In this section, we present our proposed cipher algorithm. First, we introduce the main elements that constitute the diffusion/confusion process. Then after, we discuss the encryption/decryption fundamental 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: the substitution process and the diffusion process. Moreover, the proposed cipher is applied in CTR mode of operation due to the following reasons: (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 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 [22].

4.1 Preliminaries

The proposed selective encryption algorithm consists of two main fundamentals layers: substitution and diffusion layers. Also, as mentioned before, to enhance the robustness of the proposed approach, the used key is dynamic and changes for every input image. In this section, we first provide the description about the generation of the dynamic key DK_v. Then after, we discuss separately the substitution/diffusion processes with the explanation of their main elements.

4.1.1 Dynamic Key Generation

First, the two parts of communication (the transmitter and the receiver) follow the shared key schemes. In other words, before establishing any communication, a secret Master Key, denoted as MK (128 bits), is shared secretly between the two intelligible parties. In fact, exploiting the shared key schemes rather than the public

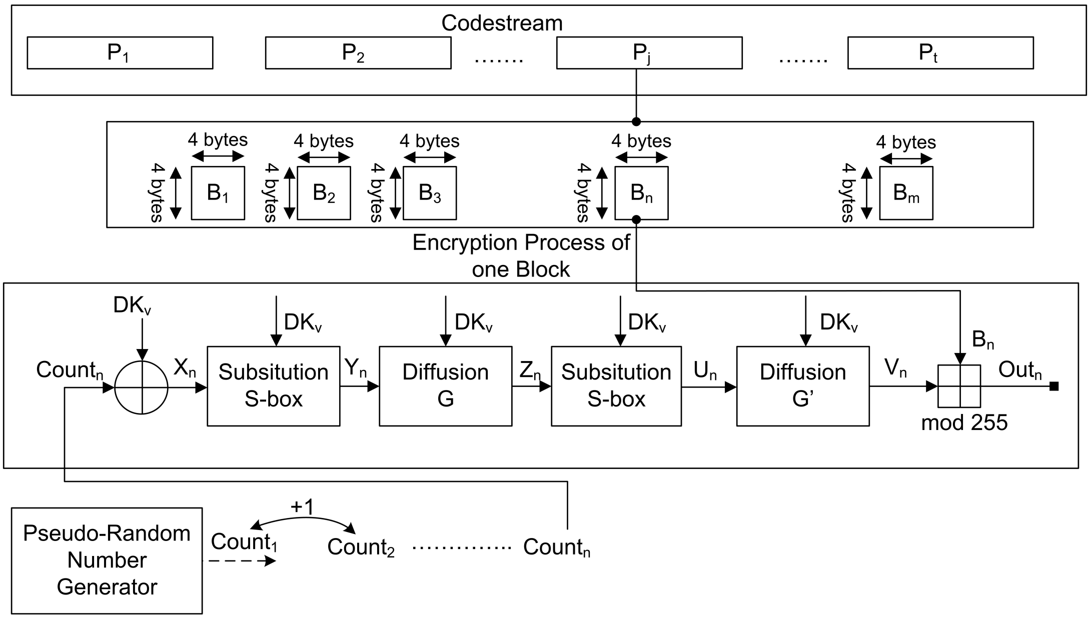


Fig. 2: The proposed selective encryption scheme for JPEG 2000 images.

key ones exhibits low complexity overhead, which can be greatly integrated with tiny, limited-resources devices (i.e., WMSNs). We note that this issue has been discussed in [23] and one of the proposed solutions can be integrated within our approach.

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-(b) and achieved following these steps:

- First, MK is Xored with an Initialization Vector IV (128-bits) to produce an output K. IV must be unpredictable and used only for one time within every input image. To ensure the unpredictability of each input IV, one of the Pseudo-Random-Generators PRNGs that are approved by the National Institute of Standards and Technology NIST in [24] can be used to generate a random and unpredictable IV sequence.
- Then, the output K is concatenated with ID_A (identity of the transmitter) and the counter v. After that, the concatenated form is hashed using SHA-512 hash function to produce the dynamic key DK_v (64 bytes).
- In the final step, the 64 bytes (i.e., $DK_v^1, \dots, DK_v^{64}$) of the dynamic key DK_v sequence are reshaped into a matrix $temp$ with size of 8×8, defined as follows:

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

This matrix contributes in the generation of the two main components of substitution and diffusion processes: (a) In the substitution process, the substitution key K_s is derived from DK_v and used later to construct the nonlinear S-box (b) While, in the diffusion process, the diffusion matrix G is constructed directly from DK_v. The generation of both components is described in sections below.

4.1.2 Substitution Box (S-box)

First, a XOR operation is used to combine elements of each column of *temp* matrix. Thus, it produces an output column $K_s = \{K_{s_1}, K_{s_2}, \dots, K_{s_8}\}$ composed of eight elements and denoted as substitution key. Then, two control parameters vectors 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:

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

Where the first input $L_0[k] = k$, $(k = 0, 1, \dots, 255)$. 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 to the result as expressed in this equation:

$$L_i = RightShift(L_i, 3). \tag{3}$$

Algorithm 1 The proposed Selective Encryption algorithm

```
Input an original JPEG 2000 code-stream, 128-bits dynamic key $DK_v$, lookup table of S-box, diffusion matrix G, a pseudo-random sequence $Count$ that is incremented by one to encrypt each selected block.
for $j \leftarrow 1$ to $t$ do
    Choose 4% of bytes from the packet data body $P_j$, using the byte's selection method (refer to pseudo-code 2).
    Re-order the selected $N_j$ bytes of the packet $P_j$ into blocks, each of size $4 \times 4$ bytes.
    if $N_j$ is not multiple of 16 then
        Padd the last block with 0's to complete the block elements.
    end if
    The selected $m$ blocks from one Packet $P_j$ follow the encryption process
    for $n \leftarrow 1$ to $m$ do
        To encrypt one block $B_n$, the counter $Count_n$ is XORed with a dynamic key $DK_v$ to produce $X_n$.
        The output $X_n$ follows the two rounds of substitution-diffusion processes to produce the output $V_n$
        The addition modulo 255 is applied between $V_n$ and the block $B_n$ to give at the end the corresponding block cipher $Out_n$ as expressed in Equation 8
        Increment the counter by one to encrypt the subsequent block.
    end for
    if $N_j$ is not multiple of 16 then
        Remove the previous padded bytes from the last cipher block $Out_m$
    end if
    Return back the encrypted $N_j$ bytes to their initial positions in the packet $P_j$
end for
Output The encrypted JPEG-2000 image.
```

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. By that, we guarantee that the produced S-box does not contain any element whose value is equal to 0xFF. We note that the generation of the inverse S-box is not required here, since the encryption is realized in CTR mode.

4.1.3 Diffusion matrix G

The diffusion matrix G consists of integer numbers instead of floating ones to avoid the complex floating operations. G matrix of size 4×4 is constructed based on M matrix as expressed in the following equations:

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

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

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

Since the diffusion process is based on a matrix multiplication, so it requires more execution time compared to the substitution process. An optimization of the matrix G is one the main goals in our extended future work.

4.2 Encryption Scheme

As mentioned before, the proposed selective encryption scheme deals with images after applying the JPEG 2000 compression. We highlight here some points:

- The JPEG-2000 code-stream can be viewed as a set of $R \times L \times P \times C$ packets. The number of packets contained in the image varies from one image to another depending on the characteristics of each image (i.e., size, color, details, region of interest). Indeed, the JPEG-2000 code-stream can be represented as a set of packets $\{P_1, P_2, \ldots, P_j, \ldots, P_t\}$, where t is the total number of packets in one codestream.
- The encryption process is applied to a selective number of bytes chosen from each packet P_j of JPEG-2000 code-stream (i.e., packets in JPEG-2000 code-stream do not necessary have the same number of bytes).
- The percentage of selected data chosen from each packet to contribute in the encryption process is set to $perc = 4\%$. This value has been justified based on experimental results as illustrated in Fig. 6.3.1 and discussed in Section 6.3.1. It represents the minimum allowable percentage that can ensure a good compromise between the high visual distortion and the low computation complexity. Also, the positions of bytes that are selected to follow the encryption process are directly dependent to S-box as exposed below.
- In order to prevent an attacker to break the algorithm by discovering the positions of encrypted bytes, we propose the following idea: we make the

Algorithm 2 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.
```

positions of these bytes related to the nonlinear S-box, which is in turn dependent on the dynamic key. In other words, N_j represents the amount of 4% of bytes that is selected from each packet data of length l. If the length l of packet P_j is less than 256 (i.e., size of S-box), then the positions of bytes are equivalent to the N_j elements in S-box (see pseudo-code 2). Otherwise, if the length l is greater than 256, each 256 elements are selected similarly to the method discussed above. By doing so, we ensure that the selected positions are dynamic and directly dependent on the dynamic key. Thus, it makes the algorithm more secure, since the positions of selected bytes become directly related to the dynamic key that is changed for every new input image. This procedure is explained in pseudo-code 2.

- The selected data, corresponding to N_j bytes are then represented as blocks of 4×4 bytes (i.e., each of 16 bytes). If N_j selected from a packet P_j is not a multiple of 16, a padding with 0's is performed on the last block to complete the block elements. The number of blocks in one packet can be represented as $\{B_1, B_2, \ldots, B_n, \ldots, B_m\}$, 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 encrypt, which are stored into two blocks: the first block consists of 16 bytes, while the second block consists of 4 bytes and 12 bytes that are zero padded).
- For the encryption process, a pseudo-random generator PRGN is used to generate first 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. This output follows the two rounds of substitution-diffusion processes to produce V_n. Finally, the addition modulo operation is applied to V_n and B_n as expressed in Equation 8 to give at the end its corresponding output Out_n.
- After encrypting all selected blocks, if N_j was not a multiple of 16, then the padded bytes must be removed from the last block, before putting the encrypted bytes in their initial positions in the corresponding packet P_j.

After highlighting these key points, we now discuss the encryption mechanism that is performed in a block-by-block manner. Below, we describe the encryption process of selected blocks contained in one packet P_j, $j \in \{1, 2, \ldots, t\}$. More explanation about the encryption scheme can be found in pseudo-code 1.

1. First, for each selected block B_n, $n \in \{1, 2, \ldots, m\}$ of one packet P_j, $j \in \{1, 2, \ldots, t\}$, the corresponding counter $Count_n$ (128 bits) is XORed with the dynamic key DK_v (128 bits) to form an output X_n (128 bits).
2. After that, X_n follows the encryption process beginning by the first substitution process through the use of the nonlinear S-box as follows:

$$Y_n = S(X_n) \quad n \in \{1, 2, \ldots, m\} \tag{6}$$

3. Then, the diffusion process is applied to each substituted block Y_n. Thus, it achieves through multiplying the components of Y_n with the dynamic diffusion matrix G modulo 255, resulting on a output denoted by Z_n. This step is expressed as follows:

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

4. After that, the second substitution process is applied to the output block Z_n to produce a block U_n, as follows:

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

5. 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:

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

6. In the final step, the resultant components of V_n are mixed with the components of the initial block B_n, by employing the arithmetic addition (modulo 255), to produce at the end the cipher block Out_n as follows:

$$Out_n = (V_n + B_n) \; mod \; 255 \tag{8}$$

7. 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.

4.3 Decryption Process

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. In order to fulfill the decryption process at the receiver side, the following steps are achieved:

1. Since the transmitter and the receiver share the same MK, hence the receiver can re-generate the same dynamic key DK_v as well as the same diffusion matrix G and the same S-box.
2. Upon receiving the JPEG-2000 codestream, the receiver chooses 4% of bytes to be decrypted from each packet P_j, $j \in \{1, 2, \ldots, t\}$, where t is the total number of packets contained in one code-stream. The receiver applies the same procedure discussed in pseudo-code 2 to choose the bytes to be decrypted from each packet.
3. For a packet P_j, the chosen bytes are organized into blocks, each of size 4×4, with $n \in \{1, 2, \ldots, m\}$, where m is the total number of blocks chosen from one packet to follow the decryption process.
4. In order to decrypt one block Out_n, the same counter $Count_n$ of that used in the encryption process is XORed with DK_v, to produce the output X_n.
5. Then X_n follows the two rounds of substitution-diffusion processes to produce at the end the output V_n.
6. To get the original block B_n using the output block Out_n, the following equation is applied:

$$B_n = (Out_n - V_n) \; mod \; 255 \tag{9}$$

5 Cryptographic Strength

In this section, we demonstrate the robustness of the proposed encryption scheme. We first begin by discussing the reason of selection of two rounds of encryption. Then, we study the cryptographic strength of the substitution layer and that of the diffusion layer.

5.1 Number of encryption rounds

The main goal of the proposed approach is to attain a high level of security. In other words, one-bit change in the plaintext must produce a completely different cipher-text of at least 50%. This property is defined by Shannon in its famous paper [3] and known as Avalanche Effect (AE).

Algorithm 3 The proposed Selective Decryption algorithm

```
**Input** an cipher JPEG 2000 image, 128-bits dynamic key $DK_v$, lookup table of S-box, diffusion matrix G, the same pseudo-random sequence $Count$ that is incremented by one to encrypt each selected block is used in the decryption.
**for** $j \leftarrow 1$ **to** $t$ **do**
    The receiver use the pseudo-code 2 to select 4% of bytes (corresponding to $N_j$ bytes) from each packet to follow the decryption process and organized them into blocks, each of size $4 \times 4$.
    **if** $N_j$ is not multiple of 16 **then**
        Padd the last cipher block with 0's to complete the block elements.
    **end if**
    The selected $m$ blocks for one packet follow the decryption process
    **for** $n \leftarrow 1$ **to** $m$ **do**
        In order to decrypt the block $Out_n$, the corresponding counter $Count_n$ is XORed with the dynamic key $DK_v$, to produce the output $X_n$
        The output $X_n$ follows the two rounds of substitution-diffusion processes to produce the output block $V_n$
        Compute the corresponding initial block $B_n$ as expressed in Equation 9
        Increment the counter by one to decrypt the subsequent block.
    **end for**
    **if** $N_j$ is not multiple of 16 **then**
        Remove the previous padded bytes from the last initial block $B_m$
    **end if**
**end for**
**Output** The original JPEG-2000 image.
```

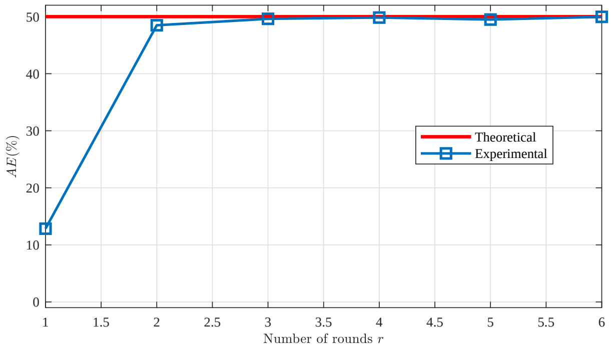


Fig. 3: The Avalanche Effect test AE (%) versus different number of rounds.

In order to specify the number of iterations that must be used in our approach and satisfy the AE property, we have studied the Avalanche Effect AE for different iterations $(1, 2, 3, \ldots 6)$. For each round, we have compared the experimental percentage value with the theoretical value (50%). Results are shown in Figure 3.

As we see in Figure 3, from the second round, the proposed approach satisfies the avalanche effect property and the obtained value (49.98%) is very close to the theoretical one. In addition, with the increase of the number of rounds, this value remains approximately the same. For this reason, two rounds of substitution-diffusion processes are selected and applied in the proposed approach to attain the required security level.

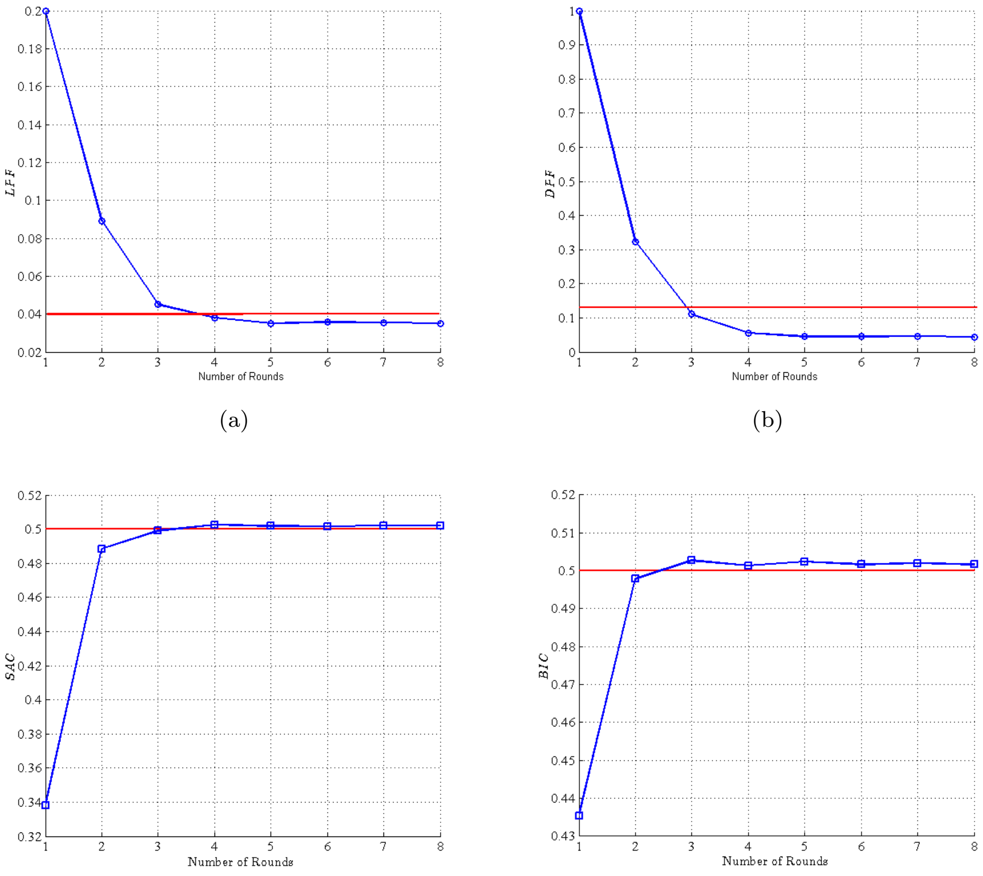
5.2 Substitution Layer

A Substitution layer is said to be robust if it demonstrates its strength under the following tests [25, 26, 27]: (a) Linear Probability approximation boolean Function (LP_F), (b) Differential Probability approximation Function (DP_F), (c) Strict Avalanche Criterion (SAC) and (d) output Bits Independence Criterion (BIC).

- LP_F: It was first introduced in [26], in the proposition of a linear cryptanalysis for DES block cipher. The idea is to find a linear approximation that connects some bits of the plain-text $\{p_1, p_2, p_3, \ldots, p_b\}$ with its corresponding cipher-text $\{c_1, c_2, c_3, \ldots, c_b\}$ (b is the number of bits). The objective behind is to guess the corresponding key value $\{k_1, k_2, k_3, \ldots, k_b\}$.

  Definition 1 *For a substitution layer F: $[0, 2^n - 1] \rightarrow [0, 2^n - 1]$, the linear probability boolean function LP_F is defined as follows:*

$$LP_F = Max_{\alpha,\beta\neq0}[\frac{card\{i/i \odot \alpha = F(i) \odot \beta\} - 2^{n-1}}{2^{n-1}}]^2 \quad (10)$$



(a)

(b)

(c)

(d)

Fig. 4: (a), (b), (c) and (d) Probability of LP_F, DP_F, SAC and BIC versus the number of iterations.

Where $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ and $\beta = \{\beta_1, \beta_2, \ldots, \beta_n\}$, $\alpha, \beta \in [1, 2^n - 1]$, card represents the cardinal and $F(i) \odot \beta = f(i_1) \bigwedge \beta_1 \oplus f(i_2) \bigwedge \beta_2 \oplus \ldots \oplus f(i_n) \bigwedge \beta_n$ and $i \odot \alpha = i_1 \bigwedge \alpha_1 \oplus i_2 \bigwedge \alpha_2 \oplus \ldots \oplus i_n \bigwedge \alpha_n$. Equation 10 can be expressed also as follows:

$$LP_F(\alpha, \beta) \neq \frac{1}{2^n - 1} \tag{11}$$

Otherwise $\sum_{\alpha=1}^{2^n-1} LP_F(\alpha, \beta) = 1 \ \forall \beta$ *and* $\sum_{\beta=1}^{2^n-1} LP_F(\alpha, \beta) = 1 \ \forall \alpha$.

This means that the immunity of the substitution layer against linear attacks is directly dependent to the uniformity of $LP_F(\alpha, \beta)$. Additionally, the lower the LP_F value will be, the higher the complexity of linear attacks and vice versa (For example, in AES block cipher, the LP_F is equal to $2^{-6} = 0.015625$.

In our algorithm, a testing of LP_F must result to a low probability as discussed in [26]. In order to evaluate the required number of necessary iterations to attain a low LP_F value, we have plotted in Figure 4-(a) the LP_F values *versus* different number of iterations. For each iteration, the computed number corresponds to the mean of 1000 tested blocks. Results show that the graph becomes steady with lower LP_F values (approximately equal to 0.04) when the number of iterations equals to 4. Hence, the substitution process becomes immune against linear attack after 4 iterations.

– DP_F: This criterion studies the effect of a slight change in plaintext pairs on the corresponding ciphertexts pairs. The cryptanalyst in this attack tries to exploit the high probability of occurrence that appears in the difference of two plaintexts. The substitution layer must have differential uniformity. Strictly speaking, a difference between two plaintexts $\triangle_{ik}$ must produce a unique difference in the output ciphertexts denoted as $\triangle f_k = F(i) \oplus F(i + \triangle_{ik})$.

Definition 2 *The differential probability approximation function DP_F is defined as follows:*

$$DP_F = Max_{\triangle_i \neq 0, \triangle_F}[DP_F(\triangle_i, \triangle_F)] \tag{12}$$

Where

$$DP_F(\triangle_i, \triangle_F) = \frac{card\{i/F(i) \oplus F(i \oplus \triangle_i) = \triangle_f\}}{2^n} \tag{13}$$

$\triangle_i \in [1, 2^n - 1]$ *and* $\triangle_f \in [0, 2^n - 1]$

In Figure 4-(b), the variation of DP_F values *versus* different number of iterations is evaluated. For each iteration, the computed number corresponds to the mean of 1000 tested blocks. Results show that after the fourth iteration, a lower probability of DP_F is attained and remains steady (< 0.1), and hence the proposed cipher can resist to differential attacks after 4 rounds.

- *SAC*: Referred to Shannon [27], an efficient cryptosystem must ensure a good substitution and diffusion properties. In other words, one-bit change in the plaintext must produce a completely different cipher-text of at least 50%. This property is also known as the Strict Avalanche Criterion (*SAC*).

Definition 3 *Assume that a plaintext with n-bits, with* $i \in [0, 2^n - 1]$ *is substituted using non-linear Function* $F(i) \in [0, 2^{n-1}]$. *In order to measure the SAC criterion, for each input, we apply the following steps:*

1. *The plaintext elements are arranged as one vector* $i = \{i_1, i_2, \ldots, i_n\}$ *and* $ik = \{i_1, i_2, \bar{i_k}, \ldots, i_n\}$, *where i and ik have only one-bit difference (the* k^{th} *bit).*
2. *The non-linear function is applied on all elements of i vector to produce:* $F=[F(1), \ldots, F(k), \ldots, F(n)]$.
3. *Another vector V is defined as* $V=[V(1), V(2), \ldots, V(k), \ldots, V(n)]$, *with* $V(k)=F(i) \oplus F(ik)$.
4. *The following relation is applied:* $a_{j,k} = a_{j,k} + v_{j,k} \quad j,k = \{1, 2, \ldots, n\}$. *Where* $v_{j,k}$ *is the* j^{th} *bit of the vector V in its binary form and* $a_{j,k}$ *is the* j^{th} *element of the matrix of dependence A of size* $n \times n$ *(A is initially defined with all zero elements).* $a_{j,k}$ *represents the relation between the bit k of the plaintext and its corresponding substituted bit j.*
5. *The SAC matrix is obtained by dividing each element of A matrix by* 2^n.
6. *Indeed, a substitution matrix achieves the SAC criterion, if the mean of matrix A is close to* 0.5.

We have studied the variation of the probability of *SAC* with regards to different number of iterations and using 1000 different blocks. The obtained results, plotted in Figure 4-(c), indicate that from the third iteration, SAC values become very close to 0.5.

- *BIC*: This measure has been defined by Webster and Tavers [27], to measure the level of dependence between the output bits after performing the substitution process.

Definition 4 *The measure of boolean independence criterion function BIC is realized as follows:*

1. *The elements of a plaintext i are arranged in a vector* $i= \{i_1, i_2, \ldots, i_n\}$ *and* $ik =\{i_1, i_2, \ldots, \bar{i_k}, \ldots, i_n\}$ *which differ of one-bit.*
2. *The nonlinear function F is applied, where* $F= [F(1), F(2), \ldots, F(k), \ldots, F(n)]$.
3. *A vector V is defined with* $V=[V(1,1), V(2,2), \ldots, V(j,k), \ldots, V(n,n)]$ *with* $V(j,k)=F(i)\oplus F(k)$ *and* $j,k= \{1, 2, \ldots, n\}$ *and* $j \neq k$.
4. *The following relation is defined* $b_{j,k}= b_{j,k} + d_{j,k}$ *where* $d_{j,k}$ *is the hamming distance of* $V(j,k)$ *in bits and* $b_{j,k}$ *is one element in the matrix of dependence B of size* $n \times n$ *(i.e., B is initially defined with zeros elements) and it represents the relation between the substituted bit j and the substituted bit k.*
5. *The BIC matrix is obtained by dividing each element of B matrix by* 2^n.
6. *Indeed, a substitution matrix achieves the BIC criterion, if the mean of matrix B is close to* 0.5 *value.*

The variation of *BIC* value for different number of iterations and using 1000 different blocks is illustrated in Fig. 4-(d). It is clearly shown that from the third iteration, *BIC* values become approximately equal to the optimal value 0.5. Hence, under this value, the proposed substitution layer becomes immune against chosen plain-text/cipher-text attacks.

From the above-mentioned tests, we can conclude that the proposed substitution process requires 4 iterations to achieve the secure cryptographic properties. For this reason, the S-box necessitates in its construction 4 iterations of the non-linear function f (as discussed in Section 4.1.2. Then, the produced S-box is applied one time on each sub-matrix for the two encryption/decryption rounds.

5.3 Diffusion layer

In order to demonstrate the strength of the proposed diffusion layer against linear and differential cryptanalysis, we have investigated the **Branch Number (BN)** test.

Definition 5 *The Branch Number BN of a matrix M of order k over the finite field* $GF(2^n)$ *is represented by* *the minimum number of non-zero elements in the input vector v and the output vector $u = M \times v$ [28]. Thus, it can be expressed as follows:*

$$\beta_M = min_{v \neq 0}\{wt(v) + wt\{M \times v\}\} \tag{14}$$

where $wt(v)$ represents the byte weight of the vector v (number of non zero bytes). A non zero byte is called also active byte.

From this definition, we can deduce that the sum of non-zero components is bounded by the branch number. Indeed, having a branch number reflects the fact that a small change in the input will produce a great change in the output.

Definition 6 *A maximum Distance Seperable (MDS) matrix of order k is the one that attains the optimal branch number, denoted by $k + 1$ [28].*

This definition can be explained by the fact that a little single change in the input vector produce a great diffusion effect, if it propagates to change all the k components of the output vector. Indeed, the largest branch number that can be achieved is equal to $k + 1$.

Returning back to our diffusion matrix, the diffusion matrix is performed with order $k = 4$ (each block consists of 4×4 bytes). In this context, if the matrix multiplication is applied with a single active byte, the output can have at most 4 active bytes, since columns in the block are treated independently. Indeed, the branch number BN is equal to 5. Hence, it is MDS. Indeed, our diffusion process is secure against differential and linear attacks.

6 Experimental Results

In this section, we first define the main settings used to evaluate the efficiency of the proposed approach. Then, we based on experimental study to set the percentage that must be selected from each packet data to produce a high image distortion. After that, the encryption strength of: (1) our proposed approach, (2) Masoudi et al. [6] appraoch and (3) Guosheng et al. [5] approach are demonstrated by testing the three main visual metrics: (1) Structural Similarity Index (SSIM), (2) Luminance Similarity Score (LSS) and (3) Edge Similarity Score (ESS).

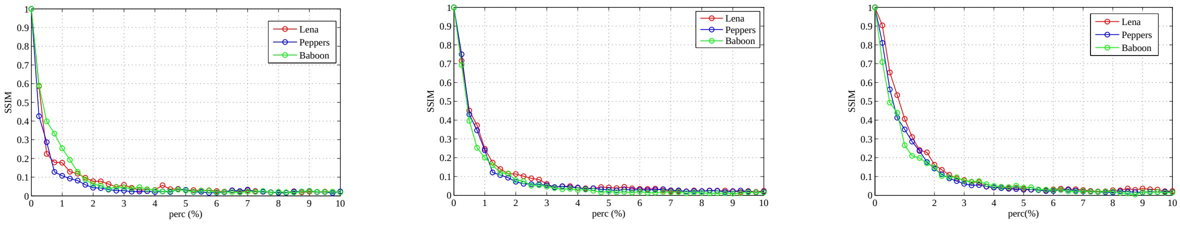
6.1 Settings

Several experiments are employed to demonstrate the efficiency of the proposed selective encryption algorithm. In this context, different images from the USC-SIPI database image [29] are utilized to perform all the experiments. In some tests, only the results for the three standards images: Lena, Peppers and Baboon are plotted due to the space limitation. Additionally, all experiments 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 [30]. The JPEG 2000 compression is performed using the standard lossy mode with: four tiles, 3 resolutions, three compression rates: $1, 0.5$ and 0.25, respectively for each resolution, one quality layer, one precinct and using the LRCP progression mode. Moreover, the same experiments are investigated for Masoudi et al. [6] and Guosheng et al. approaches [5] under the same environments. The interpretation of the results is discussed in Sections 6.3.1, 6.3.2 and 6.3.3.

6.2 Study of the percentage of data selection

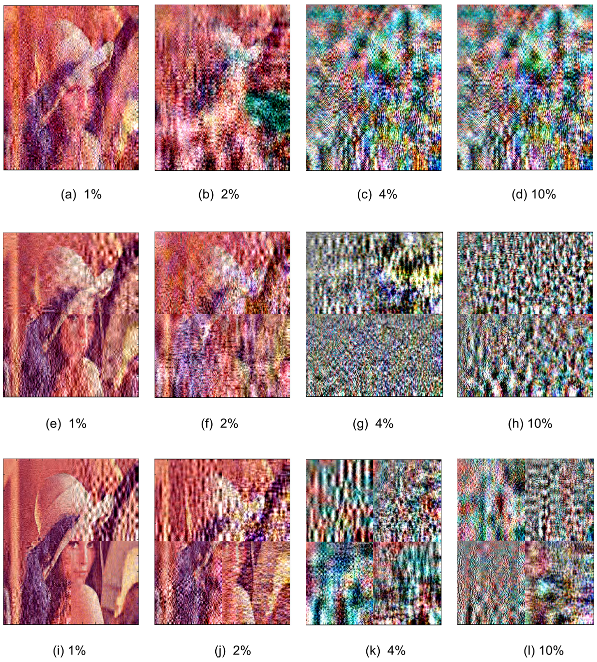
In order to fix the percentage of bytes *perc* % that must be selected from each packet body of the JPEG-2000 code-stream to follow the encryption process, the following experiment is provided: First, three standards images: Lena, Peppers and Baboon, each of size $512 \times 512 \times 3$ are used as inputs of the experiment. Then, the percentage *perc* is changed between 0 and 10%. Besides, for each percentage, the compression criterion are changed and different tiles have been investigated. In each case, the SSIM index between the original and the cipher JPEG-2000 encrypted code-stream is computed and the results are illustrated in Figure 5-(a),(b) and (c) respectively. Additionally, results for the JPEG-2000 encrypted Lena image with different *perc* and for different tiles: one, two and four tiles are illustrated in Figures 6-(a),(b) and (c) respectively.

Results in Figures 5 and 6 show that the encryption strength is highly dependent on *perc* as well as on the compression criterion. For all tested images, the higher the *perc* value, the lower the SSIM index is, and the more the encrypted code-stream is distorted. However, with $perc = 4\%$, a high visual degradation is attained (SSIM value less than 0.1). After $perc = 4\%$, the SSIM index between the original and the cipher image does not change significantly, and the curve tends to be steady with lower SSIM values. Additionally, for all tiles, at $perc = 4\%$, the image content becomes highly distorted. Hence, at $perc = 4\%$, a good compromise between the image distortion and its size is achieved. Thus, it explains the choice of $perc = 4\%$ in the proposed selective encryption approach, where a significant



(a) (b) (c)

Fig. 5: (a), (b), (c) the variation of SSIM index for three standards encrypted images: Lena, Peppers and Baboon using 500 different dynamic keys for one, two and four tiles respectively.



(a) 1% (b) 2% (c) 4% (d) 10%

(e) 1% (f) 2% (g) 4% (h) 10%

(i) 1% (j) 2% (k) 4% (l) 10%

Fig. 6: (a), (b), (c) and (d) JPEG-2000 encrypted Lena image using different amounts of percentage *perc* and with **one tile** of compression. (e), (f), (g) and (h) JPEG-2000 encrypted Lena image using different amounts of percentage *perc* and with **two tiles** of compression.(i), (j), (k) and (l) JPEG-2000 encrypted Lena image using different amounts of percentage *perc* and with **four tiles** of compression.

data reduction with a high security level is ensured for all types of tested images.

Let's not that with a percentage of 3%, a low SSIM value is achieved also and the image is degraded. However, for some images, a little content remain visible. Indeed, this percentage can be used in some applications, where the user must pay to get access to the full content as in video on demand applications.

6.3 Encryption Strength

In order to evaluate the strength of the proposed selective encryption algorithm, several quantitative metrics are employed in this section to measure the level of similarity between the plain and the JPEG-2000 encrypted images.

6.3.1 Structural Similarity Index SSIM

Due to the fact that the Human Visual System (HVS) has evolved to extract the structural information from the scene, a new metric named, Structural Similarity Index SSIM, has been presented in [31]. This index is used to measure the loss of structural information between images. The SSIM index between an original image x and its cipher image y is measured as follows:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (15)$$

Images			Encryption Strength (Proposed)			Masoudi et.al [6]			Guosheng.al [5]		
Image Number	Image Name	Image size	SSIM	LSS	ESS	SSIM	LSS	ESS	SSIM	LSS	ESS
4.2.04	Lena	512 × 512 × 3	0.0662	-16.9990	0.4104	0.0533	-16.9514	0.4156	0.0710	-17.5281	0.4104
4.2.07	Peppers	512 × 512 × 3	0.0421	-10.5130	0.4650	0.0442	-19.6736	0.4654	0.0340	-21.8186	0.4653
4.2.03	Baboon	512 × 512 × 3	0.0454	-15.1445	0.4740	0.0457	-15.1759	0.4741	0.0365	-16.0230	0.4740
4.2.06	Lack	512 × 512 × 3	0.0442	-19.7467	0.4639	0.0403	-19.8617	0.4639	0.0408	-20.8318	0.4640
4.2.02	Tiffany	512 × 512 × 3	0.0668	-22.9245	0.4252	0.0570	-23.1799	0.4250	0.0537	-26.2290	0.4253
4.1.02	Elaine	256 × 256 × 3	0.0273	-29.2905	0.4973	0.0304	-29.4161	0.4968	0.0264	-33.0274	0.4980
0.1.06	Tree	256 × 256 × 3	0.0520	-18.5563	0.5015	0.0518	-18.3447	0.5007	0.0315	-19.9927	0.5011
4.1.05	House	256 × 256 × 3	0.0801	-15.8224	0.4738	0.0412	-15.6643	0.4738	0.0478	-18.1466	0.4742
4.2.05	Airplane	512 × 512 × 3	0.0697	-20.5606	0.4622	0.0614	-20.7992	0.4624	0.0450	-22.4439	0.4626
House	House2	512 × 512 × 3	0.0801	-17.1715	0.4818	0.0555	-17.3119	0.4817	0.0516	-18.8228	0.4821
Boat.512	Boat	512 × 512 × 1	0.0711	-12.5243	0.4646	0.0679	-12.5365	0.4643	0.0643	-14.0140	0.4643
Elaine.512	Elaine2	512 × 512 × 1	0.0937	-13.6839	0.5016	0.0494	-12.4440	0.4377	0.0416	-14.1996	0.5010
4.1.04	Girl	256 × 256 × 3	0.0831	-16.0961	0.4793	0.0869	-16.3004	0.4796	0.0418	-16.4603	0.4790
4.1.01	Girl	256 × 256 × 3	0.0486	-23.5914	0.4725	0.0386	-23.4146	0.4730	0.0317	-25.1022	0.4726
4.1.03	Girl	256 × 256 × 3	0.0890	-11.2968	0.4018	0.0786	-11.4235	0.4023	0.0422	-13.2586	0.4014
4.1.07	Jelly Beans	256 × 256 × 3	0.0486	-17.7992	0.3105	0.0656	-17.7796	0.3107	0.0667	-19.3131	0.3109
4.1.08	Jelly Beans	256 × 256 × 3	0.0699	-17.6772	0.3205	0.0625	-17.5357	0.3204	0.0661	-18.6268	0.3205
5.1.09	Moon Surface	256 × 256 × 1	0.0926	-8.9420	0.4601	0.0865	-9.0352	0.4615	0.0496	-14.1417	0.4598
5.2.08	Couple	512 × 512 × 1	0.0521	-12.3771	0.4373	0.0869	-13.8539	0.5017	0.0998	-14.0450	0.4373
5.1.03	Man	1024 × 1024 × 1	0.0426	-19.8124	0.4694	0.0408	-19.9257	0.4699	0.0720	-21.8790	0.4698
-	average	-	0.0628	-17.3982	0.4402	0.0543	-17.5314	0.4476	0.0508	-19.7952	0.4487

(a)

Table 1: Encryption Strength metrics for: the proposed approach, Masoudi et al. [6] approach and Guoasheng et al. [5] approach.

μ_x, μ_y denotes the mean of the original and distorted images respectively. σ_x, σ_y refers to the standard deviation of the original and distorted images and σ_{xy} represents the covariance of both images. C_1, C_2 and C_3 are three constants that are introduced to deal with situations where the donminators are close to zero. For an 8-bit grayscale image composed of $L = 2^8 = 256$ gray-levels, $C_1 = (K_1L)^2$, $C_2 = (K_2L)^2$ and $C_3 = \frac{C_2}{2}$, where $K_1 = 0.01$ and $K_2 = 0.03$ [32]. SSIM values range in the interval [0,1]. A value of 0 means that there is no correlation between the original image and its corresponding cipher image, while a value close to 1 means that both images are nearly the same.

In this context, we have measured the SSIM metric between the previously defined original images and its corresponding JPEG-2000 encrypted images after encryption using 500 different dynamic keys for the proposed, Masoudi et al. [6] and Guosheng et al. [5] approaches. Results are presented in Table 1. In Figure 7, we have plotted the variation of SSIM index between original and JPEG-2000 encrypted images of three standard images of the proposed approach: Lena, Peppers and Baboon. As shown in the results, for all encrypted code-streams, a low SSIM value is attained (average equal to 0.0628). Thus, it means that the proposed selective encryption algorithm performs a high visual distortion in a way that no useful information about the original image can be extracted from the encrypted image.

Moreover, reffered to Table 1, the average SSIM value achieved by Masoudi et al. [6] for all tested images is equal to 0.0543 and that of Guosheng et al. [5] is equal to 0.0508. These values prove the high visual distortion achieved by both encryption schemes.

6.3.2 Luminance Similarity Score LSS

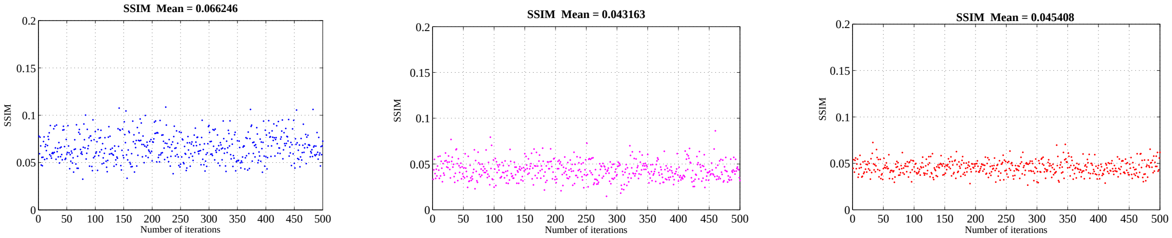
The luminance of the color space information is considered as one of the most important factors that can be extracted by an observer of a given image as suggested in the studies related to human visual system [33]. For this reason, Luminance Similarity Score (LSS) metric is involved to quantitatively measure the luminance similarity between two images. First, both images x_1 and x_2 are partitioned into blocks of size 8×8, then the average luminance of the i^{th} block (x_{1i}, x_{2i}) is computed as follows:

$$LSS = \frac{1}{n}\sum_{i=1}^{n} f(x_{1i}, x_{2i}) \tag{16}$$

Where the function $f(x_1, x_2)$ of each pair of average luminance values is defined as follows:

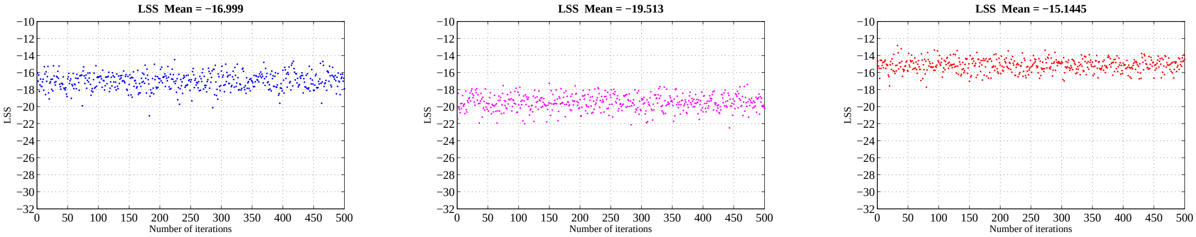
$$f(x_1, x_2) = \begin{cases} 1 & \text{if‘ } |x_1 - x_2| \leq \frac{\beta}{2} \\ -\alpha\ round\ (\frac{|x_1 - x_2|}{\beta}) & \text{otherwise} \end{cases}$$

n is the total number of blocks, α and β are two parameters, used to control the sensitivity of the score. α factor ranges between 0 and 1, and β factor is used specifically to resist to minor perturbations and noise.



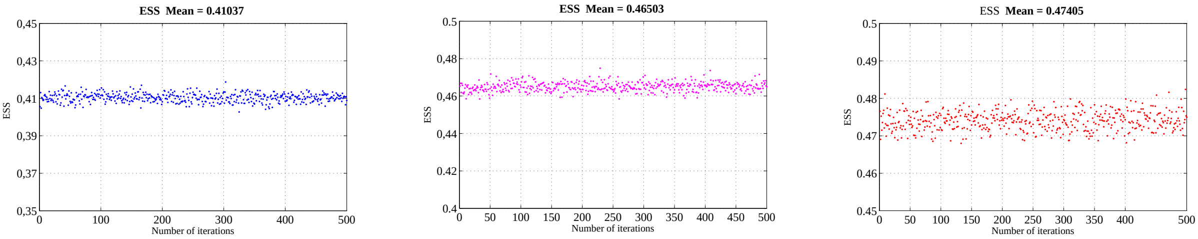
(a) (b) (c)

Fig. 7: (a), (b), (c) the variation of SSIM index for three standards encrypted images: Lena, Peppers and Baboon respectively using 500 different dynamic keys.



(a) (b) (c)

Fig. 8: (a), (b), (c) the variation of LSS index for three standards encrypted images:Lena, Peppers and Baboon respectively using 500 different dynamic keys.



(a) (b) (c)

Fig. 9: (a), (b), (c) The variation of ESS index for three standards encrypted images: Lena, Peppers and Baboon using 500 different dynamic keys.

In our experiments, α and β are set to 0.1 and 3 respectively [33]. A negative LSS value reflects a substantial dissimilarity in luminance between both images. In this context, LSS value between the original and the encrypted images is tested for different types of images and for different approaches. Results are shown in Table 1 and Figure 8. Referred to this table, the mean value of LSS for all tested images for our proposed encryption scheme is equal to -17.3982. Thus, it means that no luminance similarity is achieved between the original and the encrypted image, which in turn demonstrates the effectiveness of the proposed approach.

Moreover, reffered to Tbale 1, the average value of LSS for all tested images in Masoudi et al. approach [6] is equal to -17.5314 and that of Guosheng et al. [5] is equal to -19.7952 respectively. These values are very similar to that achieved by our proposed approach and assure the high encryption strength of these approaches.

6.3.3 Edge Similarity Score ESS

The Edge Similarity Score (ESS) measures the level of similarity in term of edge and contour information (i.e shape of object) between two images. In order to compute the ESS value, the original image p and the cipher image c are first divided into blocks, each of size 8×8 [33]. Then, the edge detection is measured for each block separately. We note that the dominant edge direction is extracted by Sobel operator and quantized into one of the eight representative directions. The eight representative edge directions are equally separated by space of 22.5 degrees in a polar coordinate system. To represent these eight directions, indices from 1 to 8 are used, where each index represents one direction. Index 0 is used to represent a non-edge block. e_{1i}, e_{2i} denotes the edge direction indices for the i^{th} block of the original and the encrypted image respectively. Then, ESS is defined as follows:

$$ESS = \frac{\sum_{i=1}^{n} w(e_{1i}, e_{2i})}{\sum_{i=1}^{n} c(e_{1i}, e_{2i})} \tag{17}$$

Where, $w(e_{1i}, e_{2i})$ is a weighting function that is defined as follows:

$$w(e_1, e_2) = \begin{cases} 0 & \text{if } e_1 = 0 \ or \ e_2 = 0 \\ |cos(\phi(e_1) - cos(\phi(e_2))| & \text{otherwise} \end{cases}$$

Where, $\phi(e)$ is the representative edge angle for an index e, and $c(e_1, e_2)$ is the indicator function defined as follows:

$$c(e_1, e_2) = \begin{cases} 0 & \text{if } e_1 = e_2 = 0 \\ 1 & \text{otherwise} \end{cases}$$

ESS value ranges between 0 and 1. A value of 0 means that there is no edge similarity between both images, then the original and the encrypted images are highly distinct. While, a value of 1 means that the edge of both images are totally matched. In our experiments, the ESS value between the original image and the encrypted one is computed for different types of images using 500 different dynamic keys. The mean of ESS value for those images are given in Table 1 for the proposed, Masoudi et al. and Guosheng approaches. Also, ESS index for the three standards encrypted images of the proposed scheme: Lena, Peppers and Baboon are plotted in Figure 9. Results show that for all tested images of the proposed approach, the average of ESS value is equal to 0.4402, which means that no edge similarity is detected between original and cipher images. Thus, it assures the high visual degradation achieved by the proposed encryption technique.

In addition, reffered to Tbale 1, the average ESS value for Masoudi et al. approach [6] is equal to 0.4476 and that of Guosheng et al. [5] is equal to 0.4487 which demonstrate that both encryption schemes ensure a high visual distortion.

7 Security Analysis

A selective encryption algorithm is said to be efficient if it has enough strength against the well known types of attacks such as statistical, differential, chosen/known plain-text, brute-force and averaging attacks [34]. Extensive experiments are performed in this section using the same parameters defined in Sub-Section 6.1 to evaluate the robustness of the proposed approach and demonstrate its efficiency against these attacks.

7.1 Statistical analysis

On of the main requirement to resist the statistical attacks, is that the cipher image must support high level of randomness [35]. To this end, several statistical tests have been performed in this section, involving the following tests: (a) Histogram analysis, (b) Entropy analysis and (c) Correlation between plain and encrypted images.

7.1.1 Histogram analysis

In statistical analysis, histogram is used to display the frequency of pixel values. After applying the encryption algorithm, a uniform behavior of the frequency counts means that all pixel values are effectively masked and no information about the original image can be extracted from the cipher one. Indeed, we can say that the proposed algorithm is robust against statistical attacks. In Fig. 10, histograms of several images of USC-SIPI image database and their corresponding cipher ones are illustrated. Results show that histograms of the encrypted images follow a uniform distribution, which is quietly different from that of the plain images.

Moreover, in order to quantitatively evaluate the level of uniformity, the **Chi-square test** is applied as expressed in Equation 18:

$$\chi^2(\gamma, interval) = \sum_{k=1}^{256} \frac{(v_k - 256)^2}{256} \tag{18}$$

Where k is the number of gray levels (for gray scale image, $k = 256$), and v_k is the observed occurrence frequencies of each gray level (0-255).

This statistical test is used to compare the observed data with that we would expect according to a specific hypothesis [36]. The hypothesis, also called ”level of significance” γ is normally set to 0.05 (or 5%) [37]. Indeed, with a significant level of 0.05 and a number of intervals equal to 256, the maximal value attained by the chi-square test is equal to 293 [38]. All values lower than this value reflect a uniform histogram behavior. To his end, the chi-square test is performed to all images in USC-SIPI database image after applying the proposed encryption algorithm using 500 different dynamic keys. Results are given in Table 3. Additionally, the chi-square test of the encrypted code-streams: Lena,

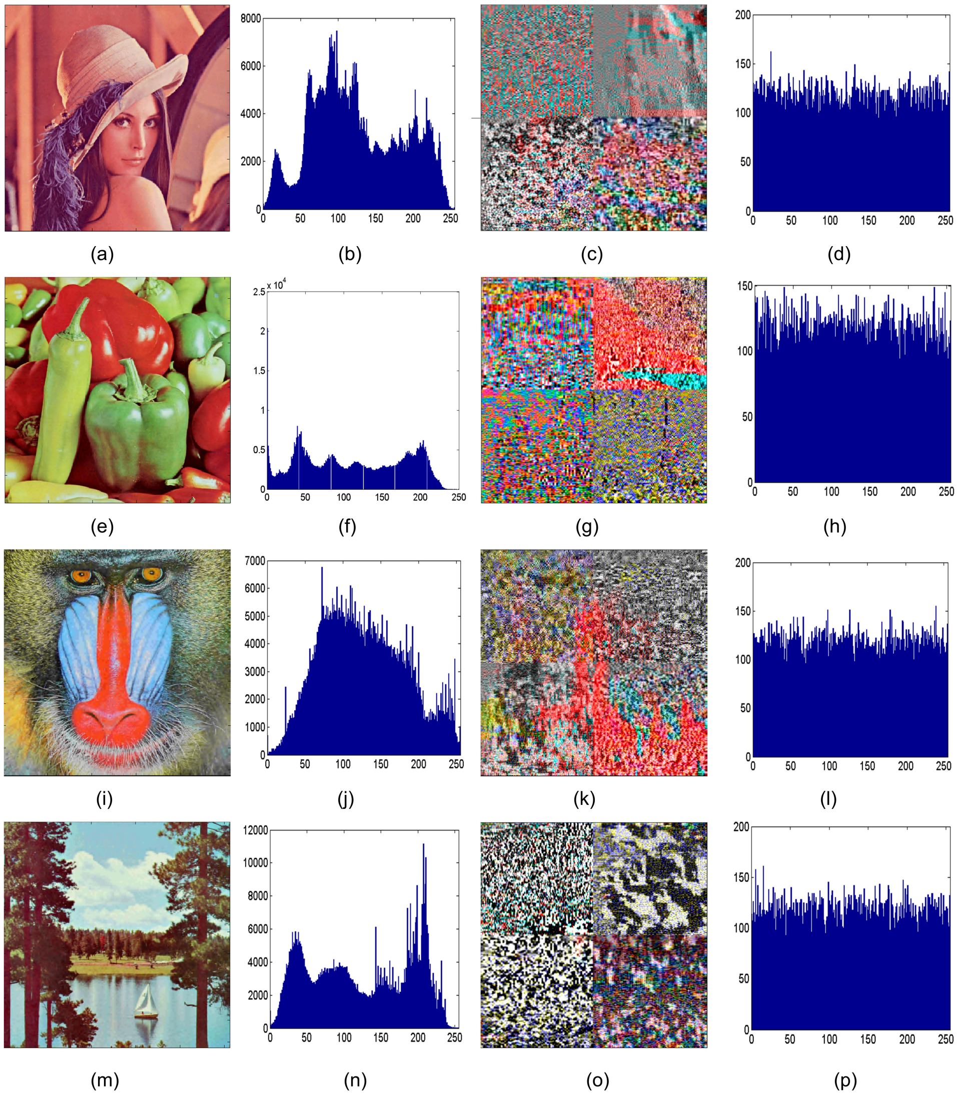


Fig. 10: (a), (e), (i), (m) Original Lena, Peppers, Baboon and Lack images respectively. (b), (f), (j), (n) Histogram of Lena, Pepper, Baboon and Lack images respectively. (c), (g), (k), (o) JPEG-2000 encrypted Lena, Pepper, Baboon and Lack images respectively. (d), (h), (l), (p) Histogram of cipher Lena, Pepper, Baboon and Lack respectively.

Peppers, Baboon images are illustrated in Figure 11-(a), (b) and (c) respectively. Referred to Table 3, the mean value of the chi-square test for all tested images of the proposed approach is equal to $257.4154 \leq 293$, which confirms the uniformity distribution of the encrypted code-stream. Indeed, we can conclude that the proposed selective encryption algorithm efficiently resists the statistical attacks.

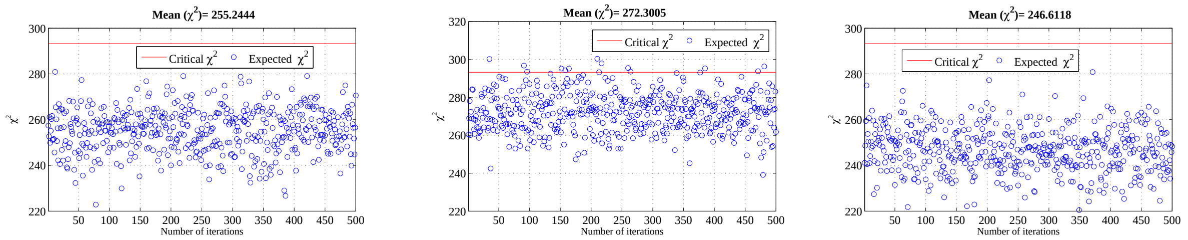
In addition, reffered to Table 3, the mean value of chi-square test for Masoudi et al. approach [6] is equal to 261.6981 and that of Guosheng et al. [5] is equal to 258.04421 which demonstrate the uniformity distribution of both approaches.

7.1.2 Information Entropy Analysis

As Shanon theory [3], the information entropy of a source message m is a metric that measures the level of uncertainty in a random variable [39], and it is defined using the following equation:

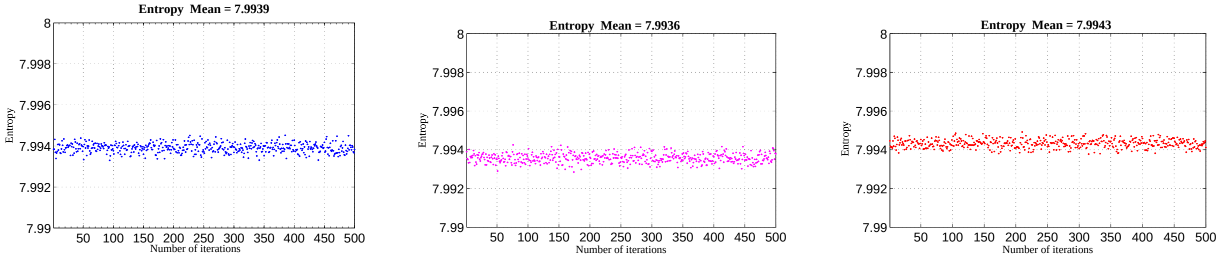
$$H(m) = \sum_{i=0}^{2^M-1} p(m_i) \log_2 \frac{1}{p(m_i)} \qquad (19)$$

Where $p(m_i)$ represents the probability of occurrence of the symbol m_i and 2^M is the total states of the information source. The entropy is presented in bits and an ideal random source has an entropy value equal



(a) (b) (c)

Fig. 11: (a), (b), (c) Chi-square test of the three standards JPEG-2000 encrypted code-streams: Lena, Peppers and Baboon respectively, using 500 different dynamic keys.



(a) (b) (c)

Fig. 12: (a), (b), (c) The variation of Entropy index for three standards JPEG-2000 encrypted code-stream: Lena, Peppers and Baboon respectively, using 500 different dynamic keys.

Images			Statistical Tests Proposed		Statistical Tests Masoudi et al. [6]		Statistical Tests Guosheng et al. [5]	
File	Description	size	chi-square	Entropy	chi-square	Entropy	chi-square	Entropy
4.2.04	Lena	512 × 512 × 3	255.2444	7.9939	274.0174	7.6982	244.2431	7.6270
4.2.07	Peppers	512 × 512 × 3	272.3005	7.9936	272.4886	7.4255	281.1109	6.7554
4.2.03	Baboon	512 × 512 × 3	246.6118	7.9943	247.9832	7.5855	255.4759	6.803
4.2.06	Lack	512 × 512 × 3	268.5226	7.9937	268.8596	7.6682	252.0583	7.0841
4.2.02	Tiffany	512 × 512 × 3	270.6755	7.9938	275.7587	7.5873	283.7495	7.1671
4.1.02	Elaine	256 × 256 × 3	275.9325	7.9709	275.6189	7.4320	264.5501	6.8322
0.1.06	Tree	256 × 256 × 3	236.5797	7.9762	237.0996	7.5397	248.9887	6.5292
4.1.05	House	256 × 256 × 3	255.0504	7.9751	256.6247	7.4714	255.3483	6.9993
4.2.05	Airplane	512 × 512 × 3	271.2779	7.9935	271.3408	7.6396	257.0244	7.1058
House	House2	512 × 512 × 3	290.2564	7.9932	288.9163	7.7058	253.6217	7.1985
Boat.512	Boat	512 × 512 × 1	236.1480	7.9944	236.0833	7.5053	255.6003	7.2661
Elaine.512	Elaine2	512 × 512 × 1	234.8080	7.9945	252.7662	7.5069	246.5866	7.6814
4.1.04	Girl	256 × 256 × 3	261.1250	7.9738	261.1335	7.3870	270.1746	7.1052
4.1.01	Girl	256 × 256 × 3	244.8036	7.9753	245.4590	7.4865	256.0663	7.3420
4.1.03	Jelly Beans	256 × 256 × 3	267.5710	7.9724	255.4828	7.5916	243.4446	7.6059
4.1.07	Girl	256 × 256 × 3	255.4849	7.9741	267.5374	7.3338	260.1384	6.9358

Table 2: Some statistical tests for: the proposed selective encryption approach, Masoudi et al. [6] approach and Guosheng et al. [23] approach

to 8 [40, 39]. A statistically strong encryption algorithm must have an entropy for their cipher information very close to the perfect value 8. In this context, Entropy is measured for different JPEG-2000 encrypted code-streams after applying the proposed selective encryption scheme and using 500 different dynamic keys and for different approaches. Results for all tested images are given in Table 2. Also, Figure 12 illustrates the variation of entropy value for the encrypted code-stream of the proposed approach: Lena, Peppers and Baboon respectively. Reffered to Table 2, the mean value of entropy for all tested images of the proposed approache is equal to 7.9853, which is very close to the theoretical value 8. Thus, it ensures the robustness of the proposed approach against entropy attacks.

In addition, reffered to Table 3, the mean value of entropy for Masoudi et al. approach [6] is equal to 7.5353 and that of Guosheng et al. [5] is equal to 7.1274 which not too close to 8. Both approaches do not achieve a high security against entropy attacks.

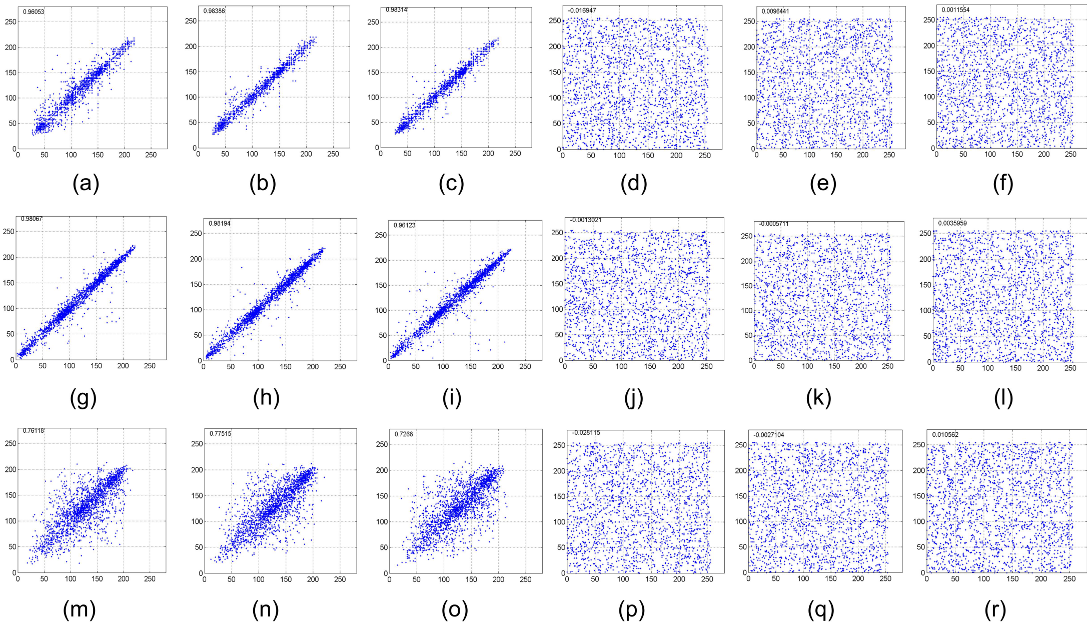


Fig. 13: (a),(b),(c), (g),(h), (i), (m), (n), (o) The horizontal,vertical and diagonal correlations of the original: Lena, Peppers and Baboon images respectively. (d),(e),(f),(j),(k),(l),(p),(q),(f) The horizontal,vertical and diagonal correlations of the JPEG-2000 encrypted code-streams: Lena, Peppers and Baboon respectively.

7.1.3 Correlation between Original and Cipher images

The main characteristic of an image is the high correlation between its adjacent pixels (correlation coefficient close to 1). During the encryption process, an encryption algorithm is said to be efficient and secure, if it succeeds to remove the spatial redundancy between pixels and produce a cipher image that is independent from the original image. In other words, an attacker must not succeed to find any clue that can help him/her to reveal any information about the original image from the cipher one. For this reason, the correlation coefficient in horizontal (HP), vertical (VP) and diagonal (DP) directions of the original images, as well as those of the JPEG-2000 encrypted code-streams, denoted as: (HE), (VE) and (DE) respectively are computed for all USC-SIPI images, after applying the proposed selective encryption approach and using 500 different dynamic keys. Results are given in Table 3. Also, results of the correlations in the three directions for the three standards JPEG-2000 encrypted code-streams: Lena, Peppers and Baboon are illustrated in Figure 13.

As shown in Table3, the horizontal, vertical and diagonal coefficients of all original images attain a high value (mean of HP=0.9504, mean of VP=0.9512, mean of DP =0.9270). While, the horizontal, vertical and diagonal correlation coefficients of their corresponding JPEG-2000 encrypted code-streams attain a very low value (mean of HE=−0.0046, mean of VE =−0.0138, mean of DE=-6.15×10^{-4}). Indeed, the proposed encryption succeeds to remove the high correlation existed between pixels of the original image, and prevent an attacker to reveal any information about the original image from the cipher one.

Moreover, the horizontal, vertical and diagonal correlation coefficients of the encrypted images using Masoudi et al. [6] approach attain a very low value (mean of HE=−0.0020, mean of VE =−0.0059, mean of DE=−0.0015). Similarly of that of Guosheng et al. [5] approach (mean of HE=5.4850×10^{-4}, mean of VE =−0.0058, mean of DE=−0.0026). These values indicate that both approach break the correlation between adjacent pixels while encryption.

7.2 Sensitivity Tests

Sensitivity test relies on how much a slight change on the key will affect the resultant security of the proposed cipher. Higher change reflects a better sensitivity of the encryption scheme. Below we analyze this type of sensitivity.

7.2.1 Key Sensitivity

In order to have enough strength against chosen-plaintext and linear attacks, a selective encryption algorithm must support also a high sensitivity against any little change in their secret key. This means that a tiny change in

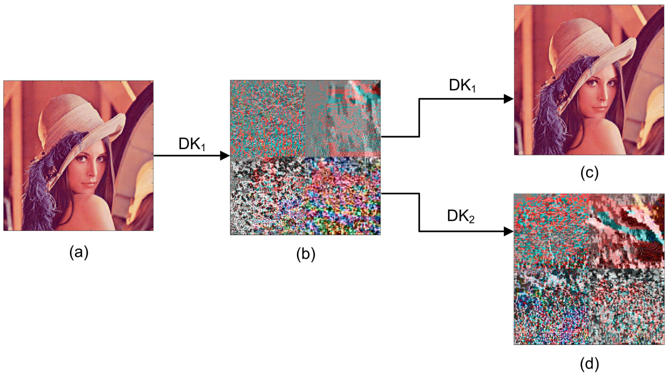


Fig. 14: (a) Lena Plain image, (b) Encrypted Lena image using DK_1, (c),(d) Decrypted Lena image using DK_1 and DK_2 respectively.

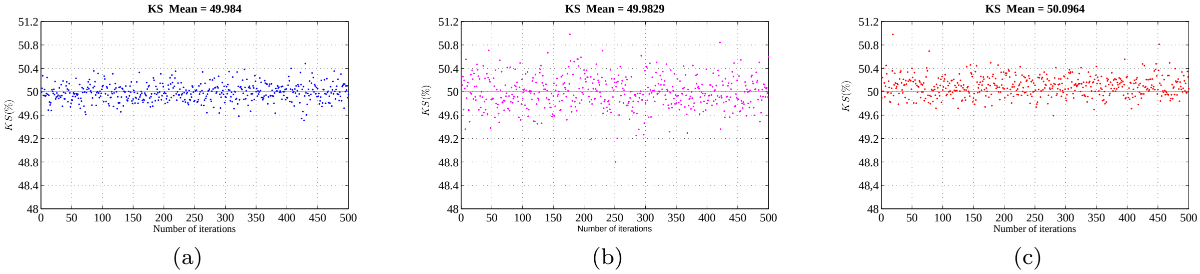


Fig. 15: (a), (b), (c) Key Sensitivity test for the three encrypted images: Lena, Peppers and Baboon respectively, using 500 different dynamic keys.

Images			Original Correlation Coefficient			Proposed approach				Masoudi et al. [6]				Guosheng et al. [5]			
File	Description	size	HP	VP	DP	HE	VE	DE	KS	HE	VE	DE	KS	HE	VE	DE	KS
4.2.04	Lena	512 × 512 × 3	0.9831	0.9839	0.9605	-5.7110×10^{-4}	0.0012	0.0096	49.9840	0.0147	-0.0319	0.0205	49.9676	0.0133	-0.0026	0.0298	50.0016
4.2.07	Peppers	512 × 512 × 3	0.9757	0.9779	0.9789	0.0043	-0.0371	0.0158	49.9829	0.0274	-0.0242	-0.0117	49.9695	0.0572	0.01005	-0.0367	50.0041
4.2.03	Baboon	512 × 512 × 3	0.7612	0.7751	0.7268	-0.0281	-0.0027	0.0106	50.09641	-0.0337	-0.0084	0.01178	50.0982	0.0006	-0.0205	-0.0185	50.004
4.2.06	Lack	512 × 512 × 3	0.9700	0.9764	0.9538	2.2627×10^{-4}	-0.0326	0.0209	49.8979	-0.0006	-0.0367	0.0022	49.8947	0.0186	-0.0024	-0.0208	49.9981
4.2.02	Tiffany	512 × 512 × 3	0.9614	0.9661	0.9303	-0.0495	-0.0375	-0.0105	49.9836	-0.0271	0.0275	-0.0273	49.9796	-0.0098	0.0069	0.0341	49.9942
4.1.02	Elaine	256 × 256 × 3	0.9481	0.9528	0.9110	0.0286	-0.0238	0.0116	49.9809	0.0294	0.0051	-0.0359	49.8765	-0.0166	-0.0069	0.0068	49.9928
0.1.06	Tree	256 × 256 × 3	0.9497	0.9478	0.9399	-0.237	-0.0221	-0.0048	49.9320	0.0045	-0.0053	0.0171	49.9072	0.0448	-0.0097	-0.0402	49.9982
4.1.05	House	256 × 256 × 3	0.9548	0.9447	0.9324	-0.0184	-0.0196	-0.0573	49.8872	-0.0053	-0.0204	0.0134	49.8731	0.0049	-0.0173	0.0053	50.0049
4.2.05	Airplane	512 × 512 × 3	0.9689	0.9738	0.9496	-0.0432	-0.0166	0.0160	49.8852	0.0047	-0.0234	-0.0084	49.9120	-0.0409	-0.0012	-0.0088	49.9994
House	House2	512 × 512 × 3	0.9670	0.9661	0.9199	0.0478	-0.0326	-0.0174	49.6217	-0.0330	0.0255	-0.0168	49.6822	-0.0094	-0.0229	0.0125	50.0037
Boat.512	Boat	512 × 512 × 1	0.9655	0.9722	0.9099	4.7184×10^{-4}	-0.0293	-0.0109	49.9379	0.0274	-0.0223	0.0154	50.177	-0.0005	-0.0173	-0.0208	50.0095
Elaine.512	Elaine2	512 × 512 × 1	0.9732	0.9703	0.9706	-0.0034	-0.005	-0.0392	49.4920	-0.0285	-0.00187	-0.0057	50.1681	-0.0087	-0.0482	-0.0165	50.0074
4.1.04	Girl	256 × 256 × 3	0.9888	0.9879	0.9575	-0.0193	0.0090	-0.0580	49.9802	-0.0534	0.0209	-0.0301	49.8955	0.0008	-0.0034	-0.0022	50.000
4.1.01	Girl	256 × 256 × 3	0.9738	0.9641	0.9447	-0.0284	0.0073	0.0238	49.8921	0.0092	-0.0101	0.0134	49.8353	-0.0102	-0.0066	-0.0084	50.005
4.1.03	Jelly Beans	256 × 256 × 3	0.9020	0.9092	0.8911	0.0070	0.0301	0.0123	49.9426	0.0491	0.0082	-0.0017	50.1006	-0.0401	-0.0201	-0.0098	50.0049
4.1.07	Girl	256 × 256 × 3	0.9821	0.9849	0.9672	-0.0711	-0.0506	0.0117	50.0935	0.0147	-0.0155	0.0491	49.9416	0.0568	0.0415	0.0254	50.0014
4.1.08	Jelly Beans	256 × 256 × 1	0.9768	0.9785	0.9546	0.0338	0.0158	0.0063	50.0997	0.0146	0.0203	0.0077	50.0478	0.0002	-0.0311	0.0183	49.9866
5.1.09	Moon Surface	256 × 256 × 3	0.9419	0.9375	0.9077	0.0204	0.0057	-0.0015	49.9429	0.0018	-0.0169	0.0024	50.1945	-0.0249	0.0133	0.0001	50.0041
5.2.08	Couple	512 × 512 × 1	0.8893	0.8763	0.8715	-0.0244	-0.0111	0.0288	49.9429	0.0029	0.0053	-0.0174	50.1680	-0.0094	-0.0056	0.0207	50.0061
5.1.03	Man	1024 × 1024 × 1	0.9756	0.9780	0.9629	0.0204	-0.024	0.0199	49.8608	0.0208	-0.0027	0.0322	50.3568	-0.0157	-0.0396	-0.0122	50.004
-	average	-	0.9504	0.9512	0.9270	-0.0046	-0.0138	-6.15×10^{-4}	49.9398	0.0020	-0.0059	-0.0015	50.0022	5.4850×10^{-4}	-0.0058	-0.0026	50.0016

Table 3: Correlation and Sensitivity Analysis for: the proposed selective encryption approach, Masoudi et al. [6] approach and Guosheng et al. [23] approach.

the key will make the decrypted image random and no precious information about the original image can be extracted from it. To test the key sensitivity of the proposed selective encryption scheme, the following scenario is performed: First, a dynamic key DK_1 is used to encrypt the JPEG-2000 Lena image. Since, DK_1 is the true key, then the decryption succeeds to recover the original plain image as illustrated in Figure 14-(c). After that, another dynamic key DK_2 built from DK_1 with only one bit difference (the Least Significant Bit LSB of the first byte) is used to decrypt the same image. However, due to this minor change on the dynamic key, the decryption process is totally failed to reconstruct the original image, instead a like-random image is produced as shown in Figure 14-(d).

Moreover, for each tested image, the following scenario is repeated to prove the high key sensitivity of the proposed selective encryption scheme. First, the proposed encryption is applied to each tested image using one correct key DK_v to produce the corresponding ci-

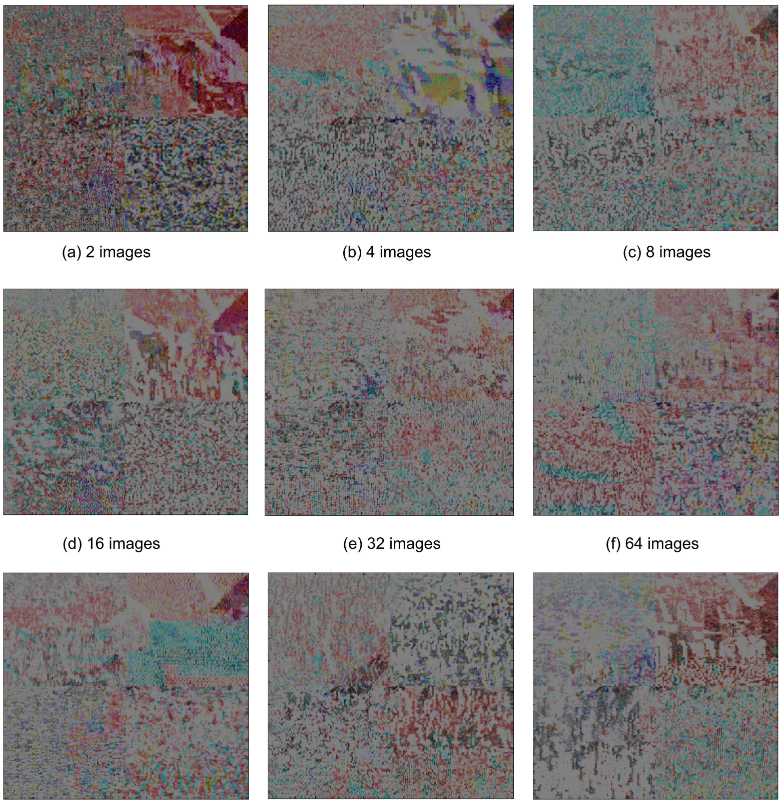


Fig. 16: Averaging of different numbers of JPEG-2000 Lena encrypted images.

$Number$	2	4	8	16	32	64	128	256	512
$SSIM$	0.1173	0.1594	0.1825	0.1838	0.1997	0.1968	0.1977	0.1988	0.1981

Table 4: Averaging attack of Lena images with different number of encrypted images.

pher image C_1. Then, the same image is decrypted using 500 different dynamic keys, each has one-bit difference from DK_v. At each iteration, a new cipher image denoted by C_2 is produced. After that, the hamming distance between the two cipher images C_1 and C_2 is computed using Equation 20. The key sensitivity (KS) for encrypted: Lena, Peppers and Baboon images are illustrated in Figure 15-(a),(b),(c) respectively.

$$KS = \frac{\sum_{k=1}^{T} C_1 \oplus C_2}{T} \times 100\% \tag{20}$$

$$= \frac{\sum_{k=1}^{T} E_{DK_v}(P_1) \oplus E_{DK'_v}(P_1)}{T} \times 100\%$$

where T is the length in bits of the plain-text image.

Additionally, referred to Table 3, the mean value of KS is equal to 49.9398, which is very close to the optimal value (50 %) [5]. Thus, it demonstrates that the proposed selective encryption algorithm is highly sensitive against any little change in its key.

Moreover, the mean value of KS for Masoudi et al [6] approach is equal to 50.0022 and that of Guosheng et al. [5] is equal to 50.0016 which demonstrate the sensitivity of these approaches against any slight change on the key.

7.3 Key Space analysis

From cryptography point of view, key space refers to the number of all possible combination of keys used in encryption algorithm. This key space must be no smaller than 2^{128} [35] to resist against brute-force attacks. For this reason, the initialization vector IV as well as the dynamic key DK_v used in the proposed selective encryption approach consist of 128 bits. This is a fairly large key space to make brute-force attacks unfeasible.

Images		size		Compression Time(sec)	Encryption Time (sec)		
File	Description	in pixels	in JPEG2000	OpenJPEG	Proposed	Massoudi et al. [6]	Guosheng et al. [5]
4.2.04	Lena	$512 \times 512 \times 3$	31kb	0.1777	0.0748	0.1323	3.0802
4.2.07	Peppers	$512 \times 512 \times 3$	31kb	0.1927	0.0873	0.9215	3.0831
4.2.03	Baboon	$512 \times 512 \times 3$	31kb	0.2104	0.0700	0.9299	3.0850
4.2.06	Lack	$512 \times 512 \times 3$	31kb	0.1984	0.0799	0.9095	3.0458
4.2.02	Tiffany	$512 \times 512 \times 3$	31kb	0.1783	0.0848	0.9815	3.1427
4.1.02	Elaine	$256 \times 256 \times 3$	8kb	0.0399	0.0387	0.2805	0.7463
0.1.06	Tree	$256 \times 256 \times 3$	8kb	0.0478	0.0348	0.3041	0.7357
4.1.05	House	$256 \times 256 \times 3$	8kb	0.0420	0.0348	0.3474	0.7357
4.2.05	Airplane	$512 \times 512 \times 3$	31kb	0.1540	0.0857	0.9760	3.0443
House	House2	$512 \times 512 \times 3$	31kb	0.1823	0.0802	0.9302	3.0581
Boat.512	Boat	$512 \times 512 \times 1$	31kb	0.0734	0.0574	0.6272	3.1093
Elaine.512	Elaine2	$512 \times 512 \times 1$	31kb	0.0715	0.0564	0.6128	3.1205
4.1.04	Girl	$256 \times 256 \times 3$	8kb	0.0400	0.0396	0.3121	0.7355
4.1.01	Girl	$256 \times 256 \times 3$	8kb	0.0510	0.0377	0.2808	0.7360
4.1.03	Jelly Beans	$256 \times 256 \times 3$	8kb	0.0385	0.0385	0.3053	0.7513
4.1.07	Girl	$256 \times 256 \times 3$	8kb	0.0321	0.0364	0.2773	0.7328
4.1.08	Jelly Beans	$256 \times 256 \times 1$	8kb	0.0352	0.0350	0.3867	0.7568
5.1.09	Moon Surface	$256 \times 256 \times 3$	8kb	0.0214	0.0280	0.1781	0.7210
5.2.08	Couple	$512 \times 512 \times 1$	8kb	0.0829	0.0727	0.3053	3.1126
5.1.03	Man	$1024 \times 1024 \times 1$	123kb	0.4084	0.1883	2.6028	13.5791
-	average	-	-	0.1069	0.0634	0.6629	2.5188

Table 5: Time Analysis for the proposed approach, Masoudi et al. approach [6] and Guosheng et al. approach [5]

7.4 Averaging Attack

On the aim of enhancing the image quality in the presence of noise, especially if the noise is independently distributed and their average is equal to 0, an averaging of multiple images can be provided. However, an attacker can make an analogy to this scenario on the encryption process. Indeed, he/she can consider the encryption process as a noise addition process. Then, he/she tries to combine multiple decrypted images with incorrect keys to see if any important information about the original image can be found. If he/she succeeds to get any information, then the proposed encryption scheme is said to be highly prone to averaging attack.

To this end, we take a number of decrypted Lena image with incorrect keys to average their pixel values. The SSIM metric is measured for each set of averaging images and results are given in Table 4 and shown in Figure 16. First, an increase of SSIM value is shown (for 2 to 16 averaging images, the SSIM index increases from 0.1173 to 0.1838) which refers to the uniformly distribution of luminance and not a similarity to the plain image. Then, the SSIM value becomes stable with low value (equal to 0.19). Indeed, no intelligible information can be extracted from the averaged images about the original image. Thus, it ensures the robustness of the proposed scheme against averaging attack.

7.5 Resistance to Chosen-plaintext attack

A chosen-plaintext attack is one of the cryptanalysis model that is based on the assumption that an attacker has the capability to arbitrary choose plain-texts and obtain correctly its corresponding cipher-texts. An efficient cipher algorithm must effectively be immune against this kind of attack. Unfortunately, permutation-only image encryption schemes are not able to resist this type of attack due to the absence of the diffusion process [41, 42]. However, in the proposed image encryption scheme, the IV is generated from a pseudo-random generator and refreshed for every new encryption process. Also, the used key DK_v derived from IV is dynamic and changed for every input image. Moreover, the sensitivity test demonstrates the high level of sensitivity of the proposed algorithm against any little change in the dynamic key. Hence, the proposed encryption scheme highly immune to chosen-plain-text attack.

7.6 Execution Time

One of the most important factor for any cipher algorithm, especially the ones dedicated to deal with constrained devices is its execution time. This is due to the fact that a secure and fast cipher algorithm, reduces the complexity of the algorithm and renders it less energy consuming. In this context, the encryption time of all previous tested images of the USC-SIPI database image is computed for: (1) our proposed approach, (2) Massoudi et al. algorithm [6] as well as (3) Guosheng et al. algorithm [5]. First, the compression time necessary to transform the plain image into JPEG 2000 format using the OpenJPEG tool [30] is computed for each plain image. Then, the encryption time required to encrypt the JPEG-2000 image, is calculated for the three approaches as shown in Table 5. Results of the encryption time show that the proposed algorithm is faster than Massoudi et al. algorithm by approximately an average of 10.45 times and by an average of 39.72

times compared to Guosheng.et al approach [5]. Results are relevant to what we have already explained about each algorithm: Guosheng et al. approach is based on encrypting the whole packet data byte-by-byte, for this reason it has the latest execution time. Moreover, the encryption process in Massoudi et al. algorithm is performed using the AES block cipher (muti-encryption rounds) in CTR mode of operation after selecting 5.43% of each packet data. For that, its execution time is slower than our proposed approach that is based on only two rounds of substitution-diffusion processes and that used less amount of data to encrypt. Indeed, the proposed encryption scheme achieves a highly increase in throughput. Then, it can effectively deal with delays-sensitive communications.

8 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.

8.1 Code-stream Compliant analysis

As mentioned earlier, format-compliant property is one of the main characteristics that must be taken into consideration when dealing with selective JPEG-2000 encryption. This is due to the fact that compliance allows to preserve the main characteristics of the original compression coding and hence it increases the robustness of image codec. By that, the decoder can correctly decode the encrypted code-stream before decryption without any risk to crash.

Returning to the proposed selective encryption approach, both substitution and diffusion processes achieve the format-compliant intrinsically in their structures. For the substitution process, all values corresponding to 0xFF are eliminated from the look-up table. Additionally, the matrix multiplication of the diffusion process is provided with modulo 255, then all 0xFF values are forbidden from the packet data. Indeed, eliminating the 0xFF marker from the encrypted code-stream, ensures that both code-words 0xFF90 and 0xFFFF will not appear in the encrypted packet body. Therefore, the encrypted code-stream is compliant to the format of the original unencrypted code-stream and preserves all its characteristics and functionality.

8.2 Compression Friendliness

In order to make the selective encryption approach meaningful, combining compression with encryption must not influence the compression performance significantly. In fact, most joint compression-encryption algorithms decrease the compression performance, since the encryption is applied before the quantization process or within the encoding process. However, using the code-stream-oriented encryption schemes, encryption is applied to the compressed data. Thus, it provides no influence on the compression performance.

Additionally, as mentioned before, the proposed encryption method scheme is format-compliant. Then, the encrypted image is compliant to the format structure of the unencrypted image. Moreover, all operations that are used in the substitution/diffusion process are based on the addition modulo operation, which neither add nor subtract a bit from the code-stream. Therefore, the proposed algorithm does not affect the compression performance significantly and satisfies the compression friendless property.

9 Conclusion

In this paper, a secure lightweight format compliant cipher algorithm for protecting the transmission of JPEG 2000 images over unreliable networks has been proposed. This algorithm is based on selective encryption, where a dynamic key is generated first, and changed for every input image. Then, the positions of selected bytes that are chosen to follow the encryption process are directly dependent on the dynamic key. Also, the encryption process consists of two main processes: substitution and diffusion process. Combining these two processes, allow to achieve a high level of security, whilst preserving the format-compliant property.

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 Massoudi et al. approach [6] as well as Guosheng et al. approach [5]. Indeed, all these features open the door to efficiently integrate the proposed JPEG-2000 images encryption scheme to deal with the transmission of images over tiny and constrained devices.

References

1. C. Christopoulos, A. Skodras, T. Ebrahimi, The jpeg2000 still image coding system: an overview,

Consumer Electronics, IEEE Transactions on 46 (4) (2000) 1103–1127.

2. F. Dufaux, G. Sullivan, T. Ebrahimi, The jpeg xr image coding standard, IEEE Signal Processing Magazine 26 (MMSPL-ARTICLE-2009-004) (2009) 195–199.
3. C. E. Shannon, Communication theory of secrecy systems*, Bell system technical journal 28 (4) (1949) 656–715.
4. R. Norcen, 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.
5. G. Gu, J. Ling, G. Xie, Z. Li, A chaotic-cipher-based packet body encryption algorithm for jpeg2000 images, Signal Processing: Image Communication 40 (2016) 52–64.
6. A. Massoudi, F. Lefebvre, C. D. Vleeschouwer, 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.
7. H. Bai, W. Lin, M. Zhang, A. Wang, Y. Zhao, Multiple description video coding based on human visual system characteristics, IEEE Transactions on Circuits and Systems for Video Technology 24 (8) (2014) 1390–1394.
8. D. Engel, T. Stütz, A. Uhl, A survey on jpeg2000 encryption, Multimedia Systems 15 (4) (2009) 243–270.
9. Assessing jpeg2000 encryption with key-dependent wavelet packets.
10. D. Engel, A. Uhl, Secret wavelet packet decompositions for jpeg 2000 lightweight encryption, in: Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on, Vol. 5, IEEE, 2006, pp. V–V.
11. H. Bai, C. Zhu, Y. Zhao, Optimized multiple description lattice vector quantization for wavelet image coding, IEEE Transactions on Circuits and Systems for Video Technology 17 (7) (2007) 912–917.
12. D. Engel, A. Uhl, An evaluation of lightweight jpeg2000 encryption with anisotropic wavelet packets, in: Electronic Imaging 2007, International Society for Optics and Photonics, 2007, pp. 65051S–65051S.
13. D. Engel, T. Stütz, A. Uhl, Assessing jpeg2000 encryption with key-dependent wavelet packets, EURASIP Journal on Information Security 2012 (1) (2012) 1–16.
14. M. Grangetto, E. Magli, G. Olmo, Multimedia selective encryption by means of randomized arithmetic coding, Multimedia, IEEE Transactions on 8 (5) (2006) 905–917.
15. J.-L. Liu, Efficient selective encryption for jpeg 2000 images using private initial table, Pattern Recognition 39 (8) (2006) 1509–1517.
16. H. Wu, D. Ma, Efficient and secure encryption schemes for jpeg2000, in: Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on, Vol. 5, IEEE, 2004, pp. V–869.
17. Y. Wu, R. H. Deng, Compliant encryption of jpeg2000 codestreams, in: Image Processing, 2004. ICIP'04. 2004 International Conference on, Vol. 5, IEEE, 2004, pp. 3439–3442.
18. O. Watanabe, A. Nakazaki, H. Kiya, A scalable encryption method allowing backward compatibility with jpeg2000 images, in: Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on, IEEE, 2005, pp. 6324–6327.
19. J. Daemen, V. Rijmen, The design of Rijndael: AES-the advanced encryption standard, Springer Science & Business Media, 2013.
20. C. Lian Jr, K.-F. Chen, H.-H. Chen, L.-G. Chen, Lifting based discrete wavelet transform architecture for jpeg2000, in: Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on, Vol. 2, IEEE, 2001, pp. 445–448.
21. D. Taubman, High performance scalable image compression with ebcot, Image Processing, IEEE transactions on 9 (7) (2000) 1158–1170.
22. K. Burda, Error propagation in various cipher block modes, IJCSNS 6 (11) (2006) 235.
23. D. d. O. Gonçalves, D. G. Costa, A survey of image security in wireless sensor networks, Journal of Imaging 1 (1) (2015) 4–30.
24. E. B. Barker, J. M. Kelsey, Recommendation for random number generation using deterministic random bit generators (revised), US Department of Commerce, Technology Administration, National Institute of Standards and Technology, Computer Security Division, Information Technology Laboratory, 2007.
25. E. Biham, A. Shamir, Differential cryptanalysis of the data encryption standard, Springer Science & Business Media, 2012.
26. M. Matsui, Linear cryptanalysis method for des cipher, in: Advances in Cryptology—EUROCRYPT'93, Springer, 1994, pp. 386–397.
27. I. Hussain, T. Shah, M. Afzal, H. Mahmood, Comparative analysis of s-boxes based on graphical sac, Analysis 2 (5).
28. T. Peyrin, Fast Software Encryption: 23rd International Conference, FSE 2016, Bochum, Germany,

March 20-23, 2016, Revised Selected Papers, Vol. 9783, Springer, 2016.

29. U. Signal, Image processing institute,“the usc-sipi image database.”.
30. M. Savinaud, M. Malaterre, J. Malik, M. Grizonnet, J. Michel, A. Descampe, Openjpeg a free and open-source solution to access the new jpeg2000 geospatial products, in: ESA Special Publication, Vol. 722, 2013, p. 302.
31. S. Li, X. Zheng, Cryptanalysis of a chaotic image encryption method, in: Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on, Vol. 2, IEEE, 2002, pp. II–708.
32. Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, Image Processing, IEEE Transactions on 13 (4) (2004) 600–612.
33. Y. Mao, M. Wu, Security evaluation for communication-friendly encryption of multimedia, in: Image Processing, 2004. ICIP’04. 2004 International Conference on, Vol. 1, IEEE, 2004, pp. 569–572.
34. X. Wang, L. Teng, X. Qin, A novel colour image encryption algorithm based on chaos, Signal Processing 92 (4) (2012) 1101–1108.
35. J. A. E. Fouda, J. Y. Effa, S. L. Sabat, M. Ali, A fast chaotic block cipher for image encryption, Communications in Nonlinear Science and Numerical Simulation 19 (3) (2014) 578–588.
36. J.-B. du Prel, G. Hommel, B. Röhrig, M. Blettner, Confidence interval or p-value?: part 4 of a series on evaluation of scientific publications., Deutsches Ärzteblatt International 106 (19) (2009) 335–9.
37. C. R. W. VanVoorhis, B. L. Morgan, Understanding power and rules of thumb for determining sample sizes, Tutorials in Quantitative Methods for Psychology 3 (2) (2007) 43–50.
38. J.-x. Chen, Z.-l. Zhu, C. Fu, L.-b. Zhang, Y. Zhang, An efficient image encryption scheme using lookup table-based confusion and diffusion, Nonlinear Dynamics (2015) 1–16.
39. G. Zhang, Q. Liu, A novel image encryption method based on total shuffling scheme, Optics Communications 284 (12) (2011) 2775–2780.
40. B. Norouzi, S. M. Seyedzadeh, S. Mirzakuchaki, M. R. Mosavi, A novel image encryption based on hash function with only two-round diffusion process, Multimedia systems 20 (1) (2014) 45–64.
41. S. Li, C. Li, G. Chen, N. G. Bourbakis, K.-T. Lo, A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks, Signal Processing: Image Communication 23 (3) (2008) 212–223.
42. C. Li, K.-T. Lo, Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks, Signal processing 91 (4) (2011) 949–954.