CFB-then-ECB Mode-Based Image Encryption for an Efficient Correction of Noisy Encrypted Images

Pauline Puteaux, William Puech

CFB-then-ECB Mode-Based Image Encryption for an Efficient Correction of Noisy Encrypted Images

Pauline Puteaux, *Student Member, IEEE* and William Puech, *Senior Member, IEEE*

***Abstract*—During the last few decades, the transmission of images over secure networks has exponentially grown. Data security in certain applications such as secure storage, authentication or privacy protection on cloud platforms, require specific strategies for multimedia. Cryptography can be used for this purpose. Indeed, using a secret key, it is possible to make data unreadable in order to secure it. Although encryption approaches are effective to make the original data unreadable, they are also very sensitive to noise. Because of the introduction of noise into an encrypted image during its transmission or storage, the original data cannot be recovered. In this paper, we first describe a new encryption mode called CFB-then-ECB and based on a combination of the CFB mode and the ECB mode for AES encryption. Using this new encryption mode, if one encrypted pixel block is noised, this will result in two incorrectly reconstructed pixel blocks during the decryption (the current and the following pixel blocks). This noise spreading is then exploited in a new proposed approach of noisy encrypted image correction. It contains two main steps involving a classifier to discriminate clear and encrypted pixel blocks. After a direct decryption of a noisy encrypted image, the first step is to identify and localize the pixel blocks that are probably incorrectly decrypted. The second step of our proposed approach is to analyze and correct these pixel blocks. Experimental results show that the proposed method can be used to blindly correct noisy encrypted images, while preserving the image structure without increasing the original data size with additional information.**

***Index Terms*—Multimedia security, image encryption, image denoising, signal processing in the encrypted domain, convolutional neural network, statistical analysis.**

I. Introduction

With the constant evolution of the Internet and in particular cloud services, more and more multimedia data are exchanged over networks and need to be protected against illegal access and fraudulent usage. For the sake of information security and privacy protection, multimedia data are encrypted before being uploaded and transmitted. The aim of encryption methods are to guarantee data privacy by converting the content of original images into unintelligible ciphertext data [26]. These approaches can be divided into two groups: block cipher or stream cipher, depending on how the data is processed to be encrypted. Moreover, encryption can be selective [14], [22], when only a subset of data is encrypted or fully when the global meaning of the image is kept entirely secret [2], [31].

During transmission or storage of encrypted images, it is often necessary to analyze or process them, without knowing the original content or the secret key used during the encryption phase [7]. When an encrypted image is corrupted during its transmission, even if the secret key is known, it becomes difficult to reconstruct the original image without errors. Some methods propose to remove noise in noisy encrypted images by using classical error correcting codes (ECC) [16], [30]. However, they are not fully format compliant or they can increase the size of the encrypted content. Furthermore, some papers have focused on error correction during the AES encryption algorithm, by parity determination and modification during the input and output of each round [17], [32]. Privacy-preserving error correction schemes have also been proposed. Hu *et al.* suggested using a double cipher to perform non-local means denoising [11]. SaghaianNejadEsfahani *et al.* proposed to resort to secret sharing for wavelet denoising [25]. Recently, Pedrouzo-Ulloa *et al.* presented an error correction scheme based on 2-ring learning with errors, for homomorphically denoising images in the encrypted domain [19]. Note that most of the previously described methods do not allow us to localize corrupted parts in the noisy encrypted image and do not preserve the original image structure or size. Some recent methods consist of performing statistical analysis of each pixel block on the encrypted image during the decryption process to determine if it is correctly decrypted or not. In these kind of methods, the authors exploited the differences between clear and encrypted pixel blocks. Islam *et al.* explained how to correct noisy AES-encrypted images by calculating three statistical measurements: global variance method, mean local variance method and the sum of the squared derivative method [12]. Puteaux and Puech described an approach based on the local Shannon entropy measurement suitable for pixel blocks of very small sizes [23].

In this paper, we propose a new method of noisy encrypted image correction which is fully format compliant and preserves the size of the original data. Our first contribution is the design of a new CFB-then-ECB encryption mode, based on the combination of the CFB mode with the ECB mode for AES encryption. Using CFB-then-ECB mode-based image encryption method, an original image is encrypted and then transmitted across a network or stored on a cloud platform. As mentioned before, this encrypted image can be noised during its transmission or storage. In this case, after a direct decryption, some pixel blocks cannot be correctly decrypted. Moreover, due to the use of the proposed CFB-then-ECB encryption mode, in case of error, there is a noise spreading in the current and the following pixel blocks. This can be exploited in a perspective of noisy encrypted image correction. Consequently, our second contribution concerns the description of a new algorithm of pixel block analysis and noisy encrypted image

P. Puteaux and W. Puech are with the Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, Centre National de la Recherche Scientifique, Univ. Montpellier, Montpellier 34095, France (e-mail: pauline.puteaux@lirmm.fr; william.puech@lirmm.fr).

correction. This proposed method is based on two main steps, namely an initialization step and a correction step. Both of these steps involve a classifier to discriminate clear pixel blocks from probable incorrectly decrypted ones (*i.e* which have to be corrected). Indeed, clear and encrypted pixel blocks present different structures and statistical properties, even in the case of very small blocks of 4×4 pixels. By using these differences during the correction process, the content of the original image can be correctly reconstructed.

The remaining paper is organized as follows. Section II gives an overview of previous methods designed in the fields of image encryption and noisy encrypted image correction. Then, the proposed method of noisy encrypted images correction relying on CFB-then-ECB mode-based image encryption is described in Section III. Experimental results are provided in Section IV and finally, the conclusion is drawn and future work is proposed in Section V.

II. Related work

In this section, we first give a brief overview of the classical encryption techniques and then we present current state-of-the-art image encryption methods. We also describe some work dealing with error correction for noisy encrypted images.

Encryption methods are used to encode a message in such a way that only authorized parties can access it. Security is then ensured by randomizing the original information by using a secret key. Cryptosystems can be symmetric, when the same key is used during the encryption and the decryption phases, like in AES [6] or DES, or asymmetric, with public and private keys, like in RSA [24] or in the Paillier cryptosystem [18]. Moreover, in symmetric cryptography, data can be encrypted independently of the last operation or by utilizing previously encrypted content [26]. The Advanced Encryption Standard (AES) algorithm was designed in 1999 by Joan Daemen and Vincent Rijmen [6]. It consists of a set of different processing operations, which are repeated for a given number of iterations. This number of rounds depends on the key size: 10 cycles of repetition for 128-bit keys, 12 cycles of repetition for 192-bit keys or 14 cycles of repetition for 256-bit keys. Moreover, the AES algorithm can support different block encryption modes such as ECB (Electronic Code Book), CBC (Cipher Block Chaining), CFB (Cipher Feedback), OFB (Output Feedback) or CTR (Counter) for example.

The aim of image encryption is to guarantee data privacy and visual confidentiality of the original content. Moreover, encryption can be *full*, when no information at all about the original image content is available after encryption, *selective*, when only a subset of the data is encrypted, or *partial*, when only a specific area of the image is encrypted and the content outside this area remains clear. Special requirements are also needed: 1) the encrypted version of the original image in a specific format has to be in the same format (*format compliance* property) and 2) the size of the original image and its encrypted version have to be identical (*size preservation* property). Therefore, standard encryption algorithms described previously cannot be used on their own and must be adapted to meet these specifications. Many other methods have also been specifically developed for image encryption in order to take into account image properties. They are then divided into three main categories, depending if they are based on substitution [20], [36], [37], pixel scrambling [9], [13], [35], or pixel block scrambling [4], [28], [29]. Substitution-based approaches often consist in performing an exclusive-or (XOR) operation between a pseudo-randomly generated binary sequence and the content of an original image in clear [20]. Note that the encryption can be achieved at pixel level (for example, for each pixel in the scanline order, from the least significant bit to the most significant bit) or at bit-plane level (for example, from the least significant bit-plane to the most significant bit-plane). Scrambling techniques are efficient and easy to implement. Their objective is to produce a non-intelligible image, by permuting the position of the pixels or of the pixel blocks. Usman *et al.* suggested randomly permuting the rows and the columns of an image in order to break the correlation of the edges [27]. In [21], Premaratne *et al.* proposed a similar approach. Wright *et al.* proposed two scrambling techniques [31]. The first one consists of permuting the locations of the pixels within the blocks. In the second one, sub-blocks within the blocks are permuted and, after that, pixels in sub-blocks are shuffled. Moreover, along with the rapid development of theory and the application of chaos, a lot of image encryption schemes based on chaos theory have been presented [9]. In most cases, in addition to a scrambling operation, the pixel values are substituted. Chen *et al.* employed a three-dimensional (3D) Arnold cat map [2] and Mao *et al.* used a 3D baker map [15] to shuffle the pixel positions during the substitution phase. Guan *et al.* applied the Arnold cat map to shuffle the positions of the image pixels in the spatial-domain and then, used the chaotic system of Chen and Ueta in [3] to modify the pixel values [10]. In order to reduce execution time, Xiang *et al.* suggested encrypting only the four most significant bits of each pixel in their scheme described in [34]. Thus, this method is selective: the four last bits of each pixel remain clear. In their paper [38], Zhu *et al.* proposed an image cryptosystem where the Arnold cat map is used for bit-level permutation, this results in both pixel position and pixel value modifications. Finally, a logistic map is employed for diffusion.

Very few previously existing methods propose to correct noisy encrypted images. Most of them propose to remove noise by using error correcting codes (ECC) [16], [30]. ECC-based approaches consist of introducing redundancy in digital data. Check bits, computed from data using specific algorithms, are added to the original bitstream. At the recipient side, check bits are derived from received data and compared with the received ones. If they are the same, no error is found, however, if an error is detected, an error correcting method needs to be applied. Error correction can be carried out using automatic repeat request (ARQ) or forward error correction (FEC). The ARQ technique consists of repeating the request for retransmission of corrupted data until all the data is verified. The FEC approach is based on encoding data using error correcting code before transmission. Some papers have also focused on error correction during the AES encryption algorithm execution [5],

[17], [32]. However, this work focuses on the specifications of encryption algorithm, whereas no importance was given to the inherent statistical properties in the data. Other error correction schemes have been specifically designed for homomorphically encrypted images [11], [19], [25]. The main drawback of these kind of methods is that they do not respect the size preservation property. Indeed, using homomorphic encryption, there is always a size expansion in the encrypted domain. Let's note nevertheless, that some relevant methods consist of performing statistical analysis of each block on the encrypted image during the decryption process to determine if it is correctly decrypted or not. Islam *et al.* explained how to correct noisy AES-encrypted images by calculating three statistical measurements: global variance, mean local variance and the sum of the squared derivative [12]. Puteaux and Puech described an approach based on the local Shannon entropy measurement suitable for pixel blocks of very small sizes [23]. They exploited differences between clear and encrypted pixel blocks to perform a significant entropy measurement and then correct a noisy encrypted image.

III. The proposed method of noisy encrypted image correction

Although encryption algorithms are useful to preserve content confidentiality of an original image, they are also extremely sensitive to noise. In the case of a noisy encrypted image, the knowledge of the secret key used during the encryption phase is not sufficient enough to reconstruct the original content without error. Indeed, even if only one bit of the encrypted image is altered, the reconstructed image can be quite different from the original one.

In this section, we present a new method of noisy encrypted image correction. In Section III-A, we first present a new encryption mode, based on a combination of the CFB mode and ECB mode. Using this CFB-then-ECB encryption mode, an original image is encrypted and then transmitted across a network or stored on a cloud platform. If this encrypted image is noised during its transmission or storage, some pixel blocks cannot be directly correctly decrypted. Due to the use of the proposed CFB-then-ECB mode-based image encryption method, in case of error, there is a noise spreading in the current and following pixel blocks which can be exploited for the analysis and correction steps. Then, in Section III-B, we describe our proposed method of pixel block analysis and noisy encrypted image correction. This method is based on two main steps namely an initialization step and a correction step. Both of these steps involve a classifier to discriminate clear pixel blocks from probable incorrectly decrypted ones.

A. The proposed image encryption scheme

1) The proposed CFB-then-ECB encryption mode: The simplest encryption mode is ECB (Electronic Code Book). Each block of 4×4 pixels (16 bytes) is encrypted independently from each others. Let us consider a pixel block $b_{clear}(i)$ from a clear image $I_{clear} = \{b_{clear}(i)\},\ 0 \leq i < \#blocks$ and $\mathcal{E}_K(\cdot)$ the AES encryption function using a key K. The encrypted version $b_{enc}(i)$ of $b_{clear}(i)$ is:

$$b_{enc}(i) = \mathcal{E}_K(b_{clear}(i)). \tag{1}$$

During the decryption phase, the clear version $b_{clear}(i)$ of $b_{enc}(i)$ can be recovered by applying $\mathcal{D}_K(\cdot)$, the AES encryption function using the key K:

$$b_{clear}(i) = \mathcal{D}_K(b_{enc}(i)). \tag{2}$$

With ECB, if we have two identical pixel blocks in the clear domain, then they are encrypted in the same way. Therefore, they remain identical in the encrypted domain. Moreover, some information about the original image content can be extracted from the encrypted image. However, it is not recommended to use this mode for multimedia applications in cryptography.

A safer encryption mode is CFB (Cipher FeedBack). With this mode, the encryption process is close to a self-synchronizing stream cipher. For the encryption, a bitstream is generated to be applied to a block of 4×4 pixels. This bitstream is obtained by encrypting the previous encrypted image block $b_{enc}(i-1)$ with the AES encryption function $\mathcal{E}_K(\cdot)$. Let us consider a pixel block $b_{clear}(i)$ from a clear image I_{clear}, its encrypted version $b_{enc}(i)$ is obtained by a simple eXclusive OR (XOR) operation with the generated bitstream:

$$b_{enc}(i) = \mathcal{E}_K(b_{enc}(i-1)) \oplus b_{clear}(i). \tag{3}$$

For the decryption, due to the symmetry of the XOR operation, the clear version $b_{clear}(i)$ of $b_{enc}(i)$ is recovered:

$$b_{clear}(i) = \mathcal{E}_K(b_{enc}(i-1)) \oplus b_{enc}(i). \tag{4}$$

Note that with the CFB encryption mode, the AES decryption function $\mathcal{D}_K(\cdot)$ is not involved. In fact, only the encryption function is required in the whole encryption/decryption process.

In this paper, for the encryption process, we propose to combine the CFB mode with the ECB mode to design a new encryption mode, called CFB-then-ECB mode. As shown in Fig. 1, in order to encrypt each pixel block $b_{clear}(i)$ of an original image, we first use the CFB mode and then the ECB mode. As a result, we obtain an encrypted pixel block $b_{enc}(i)$:

$$b_{enc}(i) = \mathcal{E}_K(\mathcal{E}_K(b_{enc}(i-1)) \oplus b_{clear}(i)). \tag{5}$$

Then, to achieve the decryption of $b_{enc}(i)$ and recover its associated clear version $b_{clear}(i)$, as with ECB, the AES decryption function $\mathcal{D}_K(\cdot)$ has to be applied first. After that, as with CFB, the block $b_{clear}(i)$ is obtained by performing a XOR operation with an encrypted version using $\mathcal{E}_K(\cdot)$ of the previous encrypted pixel block $b_{enc}(i-1)$:

$$b_{clear}(i) = \mathcal{D}_K(\mathcal{E}_K(b_{enc}(i-1)) \oplus b_{enc}(i)). \tag{6}$$

After encrypting all the pixel blocks of the image I_{clear} using the CFB-then-ECB encryption mode, an encrypted image $I_{enc} = \{b_{enc}(i)\},\ 0 \leq i < \#blocks$ is obtained. This image is then transmitted across a network and/or stored onto a cloud platform.

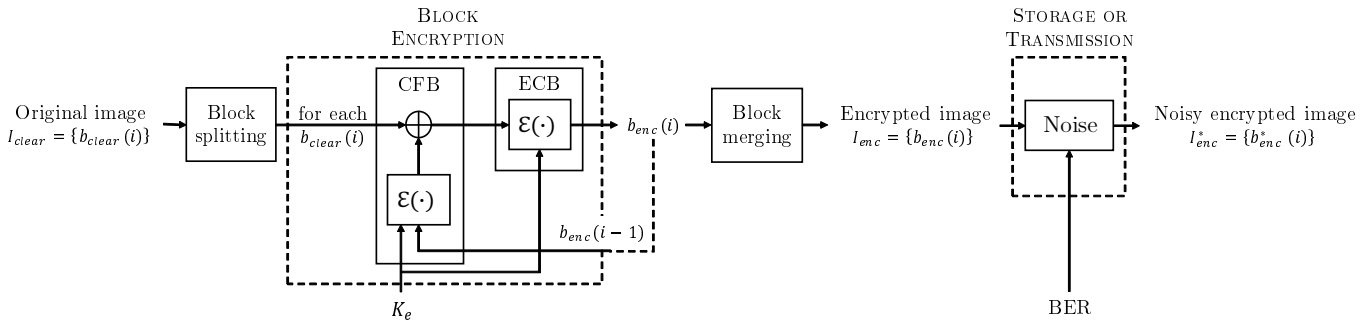


Fig. 1: Overview of the proposed CFB-then-ECB mode-based image encryption method.

2) Noise corruption during transmission: Starting from an encrypted image using the proposed CFB-then-ECB encryption mode, as illustrated in Fig. 1, an encrypted image I_{enc} can be corrupted during its transmission or storage due to channel noise. As a result, a noisy encrypted image $I^*_{enc} = \{b^*_{enc}(i)\},\ 0 \leq i < \#blocks$ is obtained. Therefore, during the CFB-then-ECB mode-based image decryption method, even knowing the key K used during the encryption step, it is not possible to correctly decrypt I^*_{enc}. Indeed, all of the encrypted blocks that are noisy cannot be decrypted at all because at least one bit has been flipped. Moreover, the previous encrypted pixel block is required to decrypt the current encrypted pixel block. Thereby, if the previous encrypted block is noisy, even if the current encrypted block value is not altered, it cannot be correctly decrypted.

Let us consider $b_{enc}(i)$ as a pixel block of the encrypted image I_{enc}. After a noise corruption, its noisy version $b^*_{enc}(i)$ corresponds to:

$$b^*_{enc}(i) = b_{enc}(i) + N(i), \tag{7}$$

where $N(i)$ is the noise associated to the pixel block $b_{enc}(i)$. The noise impact $N(i)$ can be characterized by a Bit-Error-Rate (BER), which expresses the number of bit errors divided by the total number of transferred bits. As shown in Table I, this rate is often quite low whatever the transmission type.

Wireless	Twisted cable	Coaxial cable	Optical fiber
10^{-4}	10^{-6}	10^{-9}	10^{-12}

TABLE I: BER as a function of the transmission type.

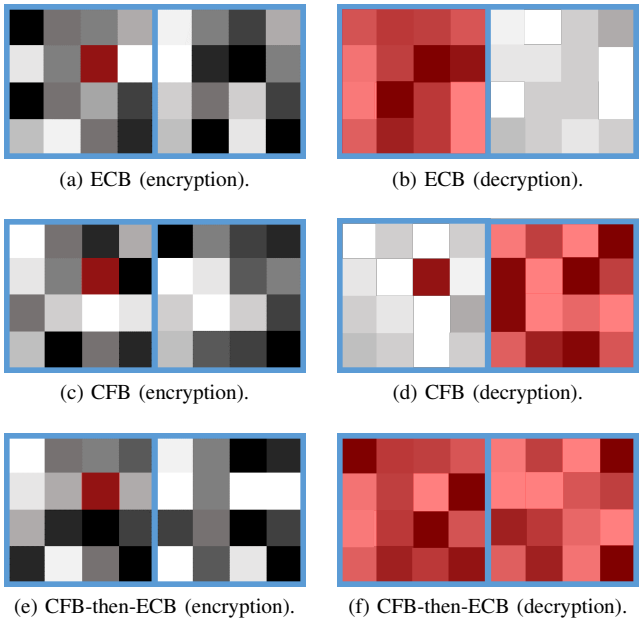
Regardless of the BER, we are interested in having the smallest possible pixel block size in order to ensure as much as possible that one bit at most per pixel block has been flipped. For example, using our proposed image encryption method, the block size is equal to 4×4 pixels. Then, if an encrypted image is noised with a BER of 10^{-3}, this randomly corrupts on average one bit every six pixel blocks.

Let $b^*_{enc}(i)$ be the a noisy encrypted pixel block from I^*_{enc}. By using Eq. (6), we obtain its associated decrypted version $b_{dec}(i)$:

$$b_{dec}(i) = \mathcal{D}_K(\mathcal{E}_K(b^*_{enc}(i-1)) \oplus b^*_{enc}(i)). \tag{8}$$

Depending on $N(i)$, $b_{dec}(i)$ is probably an incorrectly decrypted pixel block. Indeed, there are different possible cases. If the associated current encrypted pixel block or the previous encrypted pixel block is noisy ($N(i-1) \neq 0$ or $N(i) \neq 0$), then $b_{dec}(i)$ is totally different from the expected value of the original pixel block value ($b_{clear}(i)$). Conversely, if $N(i) = N(i-1) = 0$, then $b_{dec}(i)$ corresponds to the original pixel block value in clear $b_{clear}(i)$, because both the current and the previous block are not noisy:

$$b_{dec}(i) = b_{clear}(i), \quad \text{if and only if} \begin{cases} b^*_{enc}(i-1) = b_{enc}(i-1) \\ \text{and} \\ b^*_{enc}(i) = b_{enc}(i) \end{cases}. \tag{9}$$



(a) ECB (encryption). (b) ECB (decryption).

(c) CFB (encryption). (d) CFB (decryption).

(e) CFB-then-ECB (encryption). (f) CFB-then-ECB (decryption).

Fig. 2: Three different encryption modes using two neighboring pixel blocks: left column) Noisy encrypted pixel blocks: one bit of one pixel (in red) of the first block is corrupted due to noise introduction, right column) Decrypted blocks (incorrectly decrypted parts are shown in red).

In fact, it is very difficult to make the distinction between a correctly decrypted pixel block and an incorrectly decrypted one and, in particular, when the block size is very small. This

is one of the main motivations behind using the CFB-then-ECB encryption mode, instead of a standard mode during the encryption process. In Fig. 2, we illustrate the difference between these three encryption modes (ECB, CFB and CFB-then-ECB) in case of noise introduction. We consider two neighboring encrypted blocks. During the transmission or the storage of these two blocks, if one bit of the first block is corrupted, then the spreading of the noise in the decrypted blocks is very different as a function of the used mode (in red in Fig. 2.a, Fig. 2.c and Fig. 2.e). If the ECB encryption mode is used, after the decryption, the whole first pixel block (*i.e.* the block containing the noisy bit) is incorrectly decrypted. Indeed, the AES decryption function is applied to the wrong encrypted block value. Moreover, the second pixel block is perfectly reconstructed in clear (Fig. 2.b), because the first block is not involved in the decryption process of the second one. If the CFB encryption mode is used, all the pixels of the first block are correctly decrypted, except the noisy one in the encrypted domain. Indeed, because of the XOR operation, only the corrupted part (the noisy bit) cannot be recovered. On the other hand, the second pixel block cannot be reconstructed at all. Indeed, the AES encryption function is applied to the noisy encrypted first block. Then, the wrong bitstream is obtained and the XOR operation with the encrypted second block does not allow us to retrieve any of the original pixel block value in clear (Fig. 2.d). If the proposed CFB-then-ECB encryption mode is applied, then the decryption of the two pixel blocks results in two badly reconstructed pixel blocks (Fig. 2.f). In conclusion, in order to help the correction of a noisy encrypted image, it is very interesting that the noise is spread as much as possible into the two blocks. We then suggest exploiting the noise spreading by using the CFB-then-ECB encryption mode. Indeed, instead of detecting only an incorrectly decrypted pixel block or only one incorrectly decrypted bit, we investigate an effective way to highlight two incorrectly decrypted neighboring pixel blocks.

B. The proposed pixel block analysis and noisy encrypted image correction

In this paper, we propose a new method to effectively correct a noisy encrypted image during the decryption step. Our proposed algorithm, illustrated in Fig. 3, is based on two main steps in order to make the clearest possible distinction between clear and encrypted pixel blocks. By considering the directly decrypted pixel blocks, we first perform an initialization step to separate the correctly decrypted ones from those that are probably incorrectly decrypted. Then, during a second step, for each probable incorrectly decrypted pixel block, we analyze the associated possible different configurations. Finally, this allows us to perform their correction, and then to reconstruct the original image without error.

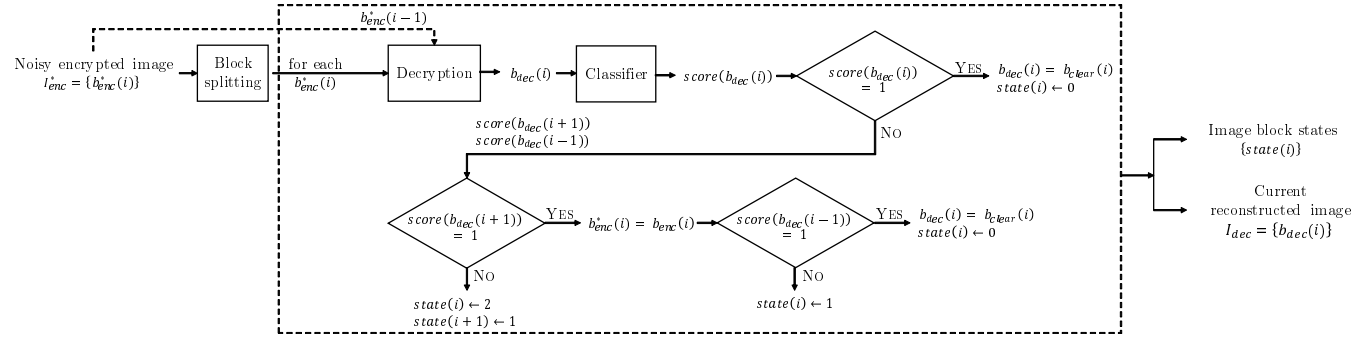
Note that a classifier is used to discriminate clear from encrypted pixel blocks in both steps. Indeed, clear and encrypted blocks do not have the same properties. In particular, clear blocks are more homogeneous than encrypted blocks or are based on a specific pattern. Based on these features, a classifier can be trained to provide a score to make this distinction. Two examples of classifiers are described and used in Section IV.

1) Initialization step: The first step of our algorithm is the initialization step, as illustrated in Fig. 3.a. During this step, the reconstructed image I_{dec} is initialized and a pixel block state $state(i)$ is associated to each block. Three different pixel block states are possible and described in Table II. They are used to learn the advancement of the correction during the whole pixel block analysis and correction process. If $state(i) = 0$, this means that the corresponding decrypted pixel block is considered as clear, *i.e.* correctly decrypted. In this case, the pixel block does not need to be corrected. If $state(i) = 1$ or $state(i) = 2$, this means that a decrypted pixel block is considered as probably incorrectly decrypted. In the case of $state(i) = 2$, we do not know if this is because of the noise spreading from the previous encrypted pixel block during decryption or due to noise corruption of the current encrypted pixel block itself. In the case of $state(i) = 1$, we consider that this is because of noise spreading from the previous encrypted pixel block during decryption. Indeed, one can deduce that the current encrypted pixel block is not noisy if the following block is recognised as $state(i) = 0$. In practice, note that, in a sequence of neighboring decrypted pixel blocks considered as probably incorrectly decrypted, the last decrypted pixel block is always recognised as $state(i) = 1$ and for the previous ones, $state(i) = 2$.

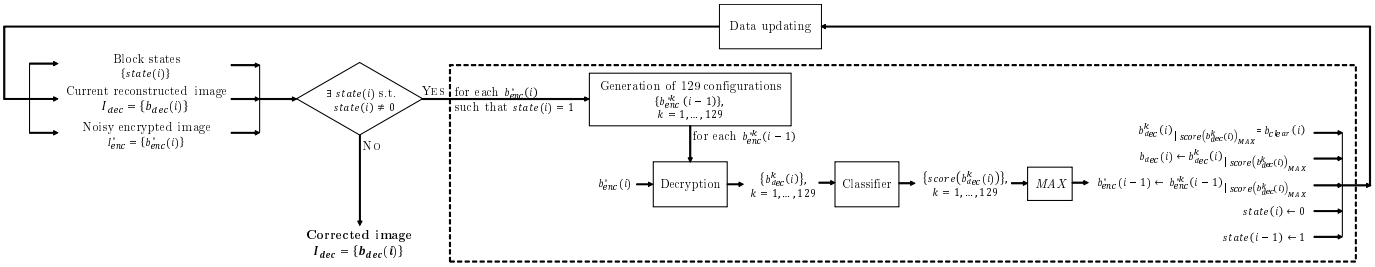
Value	Description	Correction
0	"pixel block considered as clear"	complete
1	"pixel block considered as probably incorrectly decrypted pixel block due to noise spreading from the previous pixel block during decryption"	in progress
2	"pixel block considered as probably incorrectly decrypted pixel block due to noise corruption during transmission/storage or noise spreading from the previous pixel block during decryption"	to correct later

TABLE II: Pixel block states $state(i)$ meaning.

Let us consider a noisy encrypted image I^*_{enc}. This image has been encrypted using the CFB-then-ECB mode-based AES algorithm and noised during its transmission across a network or storage onto a cloud platform. First, I^*_{enc} is split into blocks $b^*_{enc}(i)$, with $0 \leq i < \#blocks$, of 4×4 pixels. Each block $b^*_{enc}(i)$ is decrypted, using the previous neighboring block $b^*_{enc}(i-1)$ due to the use of the CFB-then-ECB encryption mode. Then, a decrypted version of each block $b_{dec}(i)$ is obtained. A classifier is then used to know if $b_{dec}(i)$ corresponds to an original pixel block in clear $b_{clear}(i)$ or if it seems to be an encrypted pixel block, which means that it is probably an incorrectly decrypted pixel block. The classification score $score(b_{dec}(i))$ is between 0 and 1, where 1 indicates that a pixel block is in clear. On the other hand, the closer the score is to 0, the more the pixel block is assimilated to an encrypted block. If $score(b_{dec}(i)) = 1$, then we are sure that $b_{dec}(i) = b_{clear}(i)$. This means that the two conditions of Eq. (9) are verified: both previous and current encrypted pixel blocks are not noisy. In this case, we have $state(i) = 0$. If $score(b_{dec}(i)) < 1$, then we have to observe the scores



(a) Initialization of the pixel block states and of the reconstructed image.



(b) Correction step and pixel block states updating.

Fig. 3: Overview of our proposed noisy image correction method based on two main steps.

associated to the two neighboring pixel blocks $b_{dec}(i-1)$ and $b_{dec}(i+1)$, as illustrated in Fig. 3.a. If $score(b_{dec}(i+1)) = 1$, this means that the current encrypted pixel block $b^*_{enc}(i)$ is not a noisy one. Then, we have $b^*_{enc}(i) = b_{enc}(i)$ and $state(i)$ is initialized to 1. In fact, in this case, $b_{dec}(i+1)$ is correctly decrypted, which could not have been the case if $b^*_{enc}(i)$ is noisy due to the noise spreading phenomenon using the CFB-then-ECB encryption mode. Conversely, if it is not the case, $b^*_{enc}(i)$ may have been corrupted itself by a noise and $state(i)$ is initialized to 2. If $score(b_{dec}(i+1)) = 1$ and $score(b_{dec}(i-1)) = 1$, this means that the current decrypted block is surrounded by two pixel blocks in clear. Therefore, this current pixel block is also necessary in clear and $state(i)$ is put equal to 0. Indeed, previous and following pixel blocks scores indicate that $b_{dec}(i)$ is not badly decrypted due to noise spreading from $b^*_{enc}(i-1)$ and $b^*_{enc}(i)$ have not been corrupted by noise.

At the end of the initialization step, all pixel block states ($state(i)$) are initialized and a first reconstructed image I_{dec} is obtained with all the pixel blocks which have been correctly decrypted $(I_{dec} = \{b_{dec}(i),\ 0 \leq i < \#blocks \mid state(i) = 0\})$. At the end of this step, the pixel blocks whose $state(i) = 0$ correspond to the original pixel blocks in clear, but all the other ones need to be fully analyzed and corrected.

2) Correction step: After the initialization step, a part of the decrypted pixel blocks have been identified as probable incorrectly decrypted and therefore, they need to be corrected. The correction step is completed by performing several rounds on the noisy encrypted image, until all the pixel blocks have not the $state(i) = 0$. The pixel block states and the current reconstructed image are also updated during the whole process. For each round, we focus on the pixel blocks $b^*_{enc}(i)$ with $state(i) = 1$. Indeed, as mentioned before, these pixel blocks refer to those that are not noisy, but whose decrypted versions $b_{dec}(i)$ are probably incorrectly decrypted due to the noise spreading phenomenon from $b^*_{enc}(i-1)$ using the CFB-then-ECB encryption mode. For each of these blocks, we propose to investigate the 129 possible configurations of the previous encrypted block before noise corruption. As there are not information about the location of the altered bit, any bit of the pixel block can be erroneous. Then, during the correction of a noisy encrypted image, we assume that one bit at most has been flipped into each pixel block.

As illustrated in Fig. 3.b, we then generate the 129 possible configurations $\{b^{*k}_{enc}(i-1),\ 1 \leq k \leq 129\}$ associated to the previous encrypted pixel block $b^*_{enc}(i-1)$. Indeed, there are the original configuration, plus the $8 \times (4 \times 4) = 128$ other possibilities, obtained by flipping one bit after another. The current encrypted pixel block $b^*_{enc}(i)$ is then decrypted using each of the 129 configurations. Therefore, we obtain 129 possible decrypted versions $\{b^k_{dec}(i),\ 1 \leq k \leq 129\}$ associated to $b^*_{enc}(i)$. All the decrypted pixel blocks $b^k_{dec}(i)$ are taken as inputs from the classifier. Associated scores $\{score(b^k_{dec}(i)),\ 1 \leq k \leq 129\}$ are computed. In most cases, 128 scores are low and only one is equal (or very close) to 1. The maximum score $score(b^k_{dec}(i))_{MAX}$ actually indicates that the associated $b^k_{dec}(i)$ corresponds to the original clear pixel block $b_{clear}(i)$. In the reconstructed image, $b_{dec}(i)$ is thus updated in consequence and the current pixel block state $state(i)$ is put equal to 0. Moreover, $b^*_{enc}(i-1)$ is also updated (by $b^{*k}_{enc}(i-1)$ such that $score(b^k_{dec}(i))_{MAX}$) and its state $state(i-1)$ is put equal to 1 (because $state(i) = 0$).

We perform this analysis and correction for each pixel block of the noisy encrypted image such that $state(i) = 1$. When

all the pixel blocks have been processed, if at least one state remains different to 0, then a next round is carried out on the whole image. In fact, this indicates that some blocks still need to be corrected. When all pixel blocks have been corrected, the reconstructed image I_{dec} corresponds to the expected image in clear.

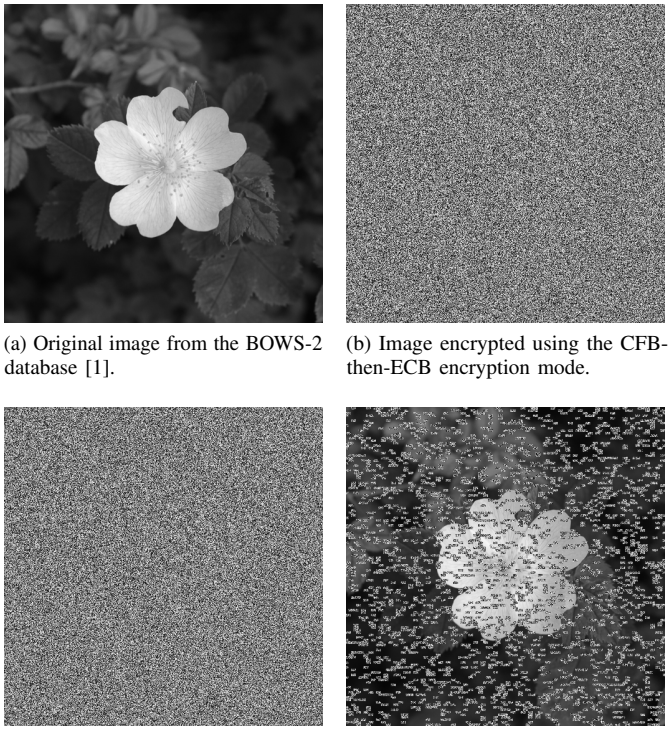
IV. Experimental results

In this section, we present the results we obtained by applying our proposed method to correct noisy encrypted images based on a CFB-then-ECB mode-based image encryption. In Section IV-A, we illustrate the proposed CFB-then-ECB encryption mode and compare this new encryption mode to more standard encryption modes like ECB and CFB. In Section IV-B, we describe two different classifiers which can be used in our proposed method in order to discriminate clear pixel blocks from encrypted pixel blocks. Section IV-C presents a full example of the proposed method of noisy encrypted image correction using these two specific classifiers. In Section IV-D, we present the achieved results on 100 images from the BOWS-2 database [1]. Finally, in Section IV-E, we compare our proposed encryption method with more standard image encryption approaches to correct noisy encrypted images, and discuss its efficiency.

A. The proposed CFB-then-ECB mode-based image encryption method

In Fig. 4, we provide an illustration of the proposed CFB-then-ECB mode-based image encryption method. Using an original image from the BOWS-2 database [1] (512×512 pixels encoded with 256 grey-levels) illustrated in Fig. 4.a, we apply an AES encryption in CFB-then-ECB mode with blocks of 4×4 pixels to obtain the encrypted image illustrated in Fig. 4.b. Note that no visual information about the original image content remains (PSNR of $7.22\ dB$). Fig. 4.c presents a noisy encrypted image of Fig. 4.b achieved using noise with a BER $= 2.6 \times 10^{-3}$ which means that on average, one bit every three blocks is randomly flipped. Note that this BER value is relatively high in comparison with real-life values (displayed in Table I). PSNR between the original image and the noisy encrypted image remains low ($7.21\ dB$) and PSNR of $33.20\ dB$ between the encrypted image and the noisy encrypted image indicates that noise power is high. Fig. 4.d illustrates that a direct decryption without any analysis is not possible (PSNR of $14.97\ dB$), even if the secret key used during encryption is known. This is due to the large number of noisy encrypted blocks which are then incorrectly decrypted. Moreover, without analysis, it is not possible to localize incorrectly decrypted blocks and to discriminate them from correctly decrypted blocks from the original image.

In Fig. 5, we compare the directly decrypted images obtained from encrypted images using standard encryption modes such as ECB and CFB and the proposed CFB-then-ECB encryption mode. Using the ECB encryption mode, pixel blocks are encrypted independently from each other. As a result, after direct decryption, if a pixel block is badly reconstructed, there is no impact on its neighboring pixel blocks (Fig. 5.a). Using the CFB encryption mode, pixel blocks are encrypted by performing an XOR operation with an encrypted version of the previous encrypted block. Consequently, in Fig. 5.b, we can see that if an encrypted pixel block is noised, there are two different consequences in the directly decrypted image. First, a noisy bit in the current pixel block is badly decrypted. Moreover, due to the noise spreading phenomenon, all bits of the following pixel block are incorrectly decrypted. Note that for this encryption mode, it is not possible to exploit the fact that there are decryption errors in two neighboring blocks in order to correct them. Indeed, as only one bit is incorrectly reconstructed in the decrypted version of the noisy encrypted block, it is very difficult to identify it, especially when this is a least significant bit. Fig. 5.c illustrates the directly decrypted image obtained from the noisy encrypted image using the CFB-then-ECB mode-based image encryption method. With this new encryption method, if an encrypted pixel block is noised, then both the current and the following pixel blocks are badly decrypted. Therefore, during the correction phase, this noise spreading phenomenon can be exploited. Indeed, two badly decrypted neighboring blocks are easier to identify than an isolated one.



(a) Original image from the BOWS-2 database [1].

(b) Image encrypted using the CFB-then-ECB encryption mode.

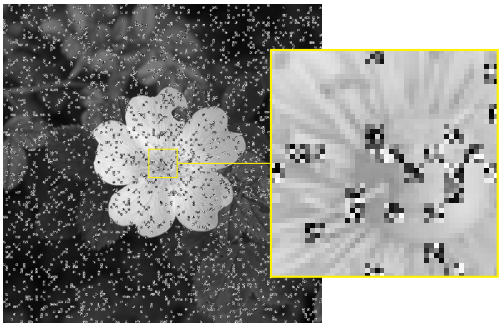
(c) Noisy encrypted image.

(d) Directly decrypted image.

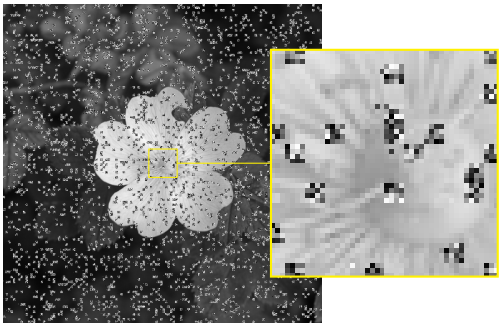
Fig. 4: The problem of noisy encrypted image decryption without correction.

B. The two classifiers used

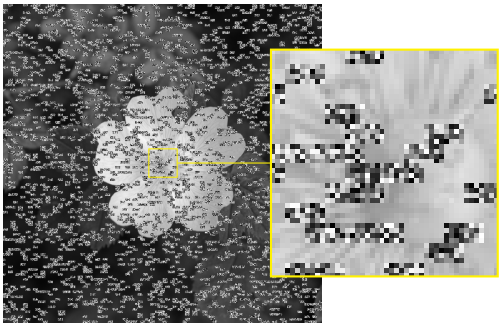
The problem of discriminating a clear pixel block from an encrypted one is very difficult, in particular when the pixel block size is very small. In this section, we describe two classifiers which can be integrated into our proposed noisy



(a) ECB encryption mode.



(b) CFB encryption mode.



(c) CFB-then-ECB encryption mode.

Fig. 5: Directly decrypted images obtained from encrypted images using different encryption modes.

encrypted image correction method to infer if a pixel block of 4×4 pixels is in clear or encrypted.

1) Local entropy-based classifier: Puteaux and Puech have shown in [23] that Shannon entropy can be efficient to differentiate a clear pixel block from an encrypted one. Indeed, in the encrypted domain, the pixel distribution tends to be uniform. Therefore, the entropy value is close to maximal and then, larger than in the clear domain.

Let b_i be a block of k pixels encoded on l grey-levels. Local entropy (*i.e.* inside the pixel block) is bounded by the logarithm of the minimal value between its size k and the number of grey-levels l:

$$H_{(k,l)}(b_i) \leq \log_2(\min(k,l)) \ bpp. \tag{10}$$

Therefore, for pixel blocks of 4×4 pixels encoded on 256 grey-levels, the maximal value is reached when each pixel value is different. In this case, the pixel sample is sparse, because all of the grey-level values cannot be present in b_i. For this reason, the entropy measurement may be erroneous and a pixel block in clear may be considered as encrypted. A solution consists of quantizing the number of grey-levels for the entropy measurement in order to find the best trade-off between k and l [23]. The authors also point out that, for pixel blocks of 4×4 pixels, best classification results are achieved with 8 grey-levels and by performing the entropy measurement, not on the image itself, but on its distance map generated by calculating differences between pixels.

The proposed local entropy-based classifier then uses optimal parameters. The classification score associated to a pixel block b_i is therefore defined as:

$$score(b_i) = 1 - \frac{H_{(16,8)}(distance\_map(b_i))}{\log_2(\min(16,8))}. \tag{11}$$

2) CipherNet classifier: In order to discriminate clear pixel blocks from encrypted ones, we propose a new Specialized Light Convolutional Network called CipherNet with the architecture presented in Fig. 6. For our application case, we consider a block b_i of 4×4 pixels as an input of our CNN. In the first layer, several high-pass filters from the Spatial Rich Model [8] are applied. These high-pass filters, illustrated in Fig.7, are used to extract the high frequencies. After that, three layers of convolution and two layers of pooling allow us to get a 1024-D feature vector. Finally, a prediction $pred(b_i)$ between 0 and 1 is obtained, where 0 is for the class "Clear pixel block" and 1 is for the class "Encrypted pixel block". The classification score associated to a pixel block b_i is therefore defined as:

$$score(b_i) = 1 - pred(b_i). \tag{12}$$

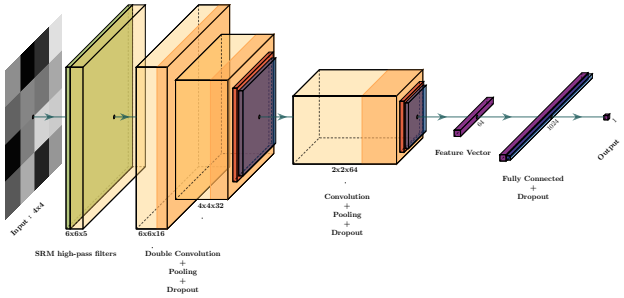


Fig. 6: The proposed CipherNet architecture.

In order to train our classifier and ensure its efficiency, we have considered a database of 32 million pixel blocks, 16 million are from clear images and 16 million from CFB-then-ECB encrypted images. Then, this database has been split into 3 balanced sub-datasets: 17 million pixel blocks for the training phase, 4 million for the validation phase and 11 million for the testing phase. For the training phase, we have used batches of 32 pixel blocks and only one epoch has been needed to reach the convergence of our model.

$$\frac{1}{3}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -3 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(a) HSB.

$$\frac{1}{3}\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -3 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(b) VBH.

$$\frac{1}{4}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 2 & -4 & 2 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(c) RF0.

$$\frac{1}{12}\begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{bmatrix}$$

(d) RF1.

$$\frac{1}{2}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(e) RF2.

Fig. 7: The five SRM kernels used for high-pass filtering of the first layer of the proposed CipherNet.

3) Performance comparisons: Whatever the classifier, the classification score $score(b_i)$ associated to a pixel block b_i is between 0 and 1. If $score(b_i) = 1$, we assume that we are sure that b_i is in clear. Conversely, if $score(b_i) \neq 1$, this means that b_i is (probably) an encrypted pixel block.

(a) Local entropy-based classifier.

(b) CipherNet classifier.

Fig. 8: Prediction maps obtained with the two used classifiers: pixel blocks predicted as encrypted are in red, $score(b_i) \neq 1$.

In Fig. 8, we displayed the prediction maps obtained with the local entropy-based classifier (Fig. 8.a) and the CipherNet classifier (Fig. 8.b) on a clear image from the BOWS-2 database [1]. Pixel blocks which are correctly predicted ($score(b_i) = 1$) are shown in clear. However, we can see that several textured pixel blocks (in red) are incorrectly predicted as being encrypted. These pixel blocks represent respectively 10.36% and 1.56% of the total number of pixel blocks with the local entropy-based classifier and the CipherNet classifier. Based on these initial results, this suggests that the CipherNet classifier seems to be more efficient at inferring if a block is in clear or encrypted.

Table III shows the two classifiers performances by providing confusion matrices, accuracy and F1-score. These results are obtained using 1,638,400 pixel blocks, where 819,200 are in clear and 819,200 are encrypted. In Table III.a and Table III.b, we have categorized the pixel blocks b_i as follows:

- TP (True Positive): b_i is in clear and predicted as being in clear ($score(b_i) = 1$),
- FP (False Positive): b_i is encrypted and predicted as being in clear ($score(b_i) = 1$),

Local entropy-based	
TP = 80%	FP = 0%
FN = 20%	TN = 100%

(a)

CipherNet	
TP = 91%	FP = 0%
FN = 9%	TN = 100%

(b)

	Classifier			
	Local entropy-based		CipherNet	
	Accuracy	F1-score	Accuracy	F1-score
Min.	0.2235	0.3654	0.3020	0.4639
Max.	0.9996	0.9998	1	1
Average	0.5784	0.7092	0.8054	0.8798
Q1	0.3996	0.5710	0.6823	0.8112
Median	0.5419	0.7029	0.8488	0.9182
Q3	0.7466	0.8549	0.9673	0.9834

(c)

TABLE III: The two classifiers performance measurements: a) and b) Confusion matrices, c) Accuracy and F1-score. Results obtained using 1,638,400 pixel blocks, 819,200 are in clear and 819,200 are encrypted.

- TN (True Negative): b_i is encrypted and predicted as being encrypted ($score(b_i) \neq 1$),
- FN (False Negative): b_i is in clear and predicted as being encrypted ($score(b_i) \neq 1$).

For both classifiers, we can see that there are no false positives, which means that if a pixel block is encrypted, it is never predicted as being in clear. This is particularly important for the aim to correct noisy encrypted pixel blocks. Indeed, we have to be able to identify all the incorrectly decrypted pixel blocks in the directly decrypted image. We can also observe that most of the pixel blocks in clear are correctly predicted whatever the used classifier, and especially using the CipherNet classifier (91%). Table III.c also demonstrates the efficiency of the two classifiers and the superiority of CipherNet compared to the local entropy-based classifier (accuracy of 0.8054 *vs* 0.5784 on average). Moreover, it is important to notice that none of the two used classifiers are able to perfectly predict all the pixel blocks. This highlights the fact that they cannot be used alone to perform the correction of noisy encrypted images. Indeed, they must be integrated into our new proposed algorithm.

C. Full example of the proposed method

In Fig. 9, we illustrate the whole process of our proposed method on the *Lena* image with a size of 512×512 pixels encoded with 256 grey-levels (Fig. 9.a). First, we provide the obtained prediction maps using the two classifiers presented in Section IV-B. Fig 9.b and Fig. 9.c are respectively the prediction maps obtained with the local entropy-based classifier and with the CipherNet classifier. In both figures, we represent in red the pixel blocks predicted as encrypted, indicating that they are incorrectly predicted ($score(b_i) \neq 1$). As already observed in Section IV-B, these pixel blocks are located in textured areas and on contours. Moreover, we can see that more pixel blocks are incorrectly predicted using the local entropy-based classifier than using the CipherNet classifier ($2,595$ pixel blocks (16%) *vs* 539 pixel blocks (3%)). Fig. 9.d

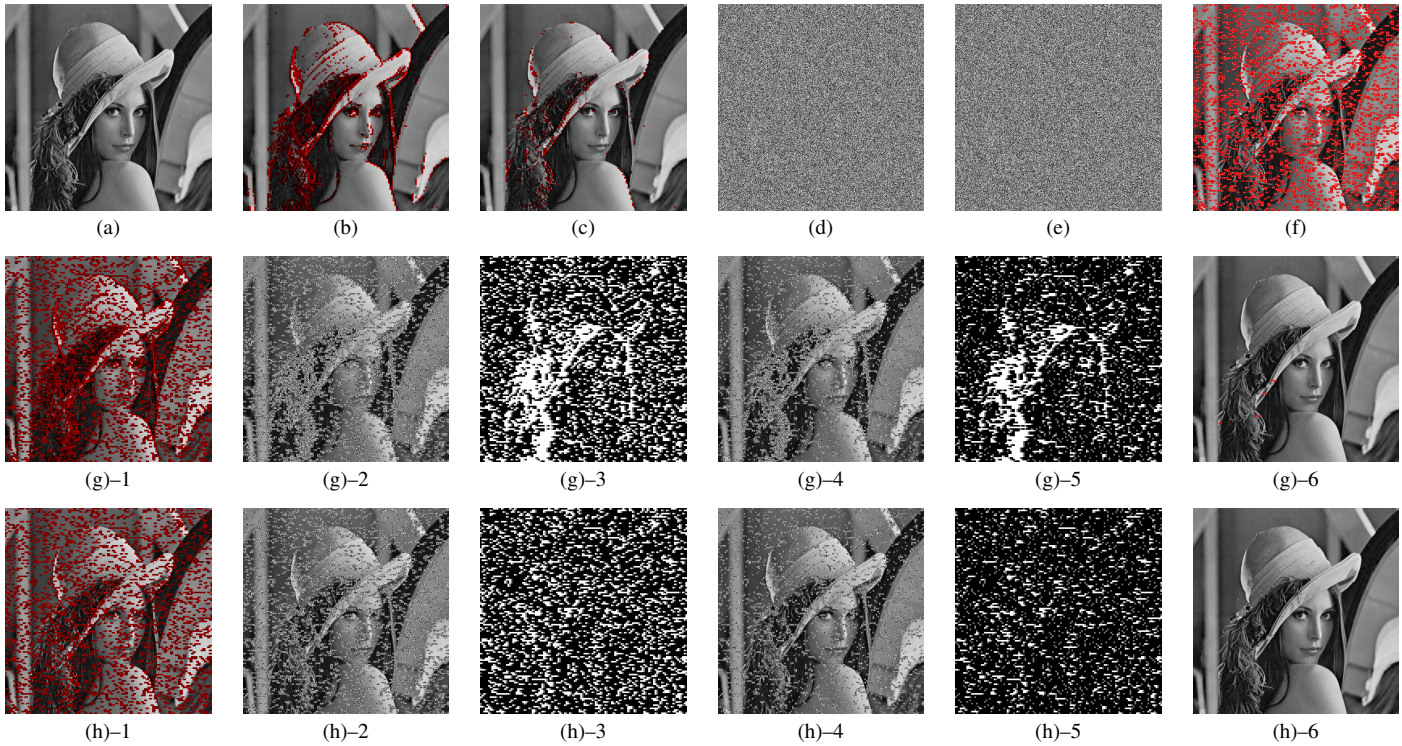


Fig. 9: The proposed noisy encrypted image correction method: a) Original *Lena* image. Prediction maps obtained from (a): b) with the local entropy-based classifier (in red: pixel blocks predicted as encrypted, $score(b_i) \neq 1$), c) with the CipherNet classifier. d) Encrypted image associated to (a) using the CFB-then-ECB encryption mode, e) Noisy encrypted image obtained from (d), f) Directly decrypted image without correction obtained from (e) (framed in red: incorrectly decrypted pixel blocks). Correction using: g) the local entropy-based classifier and h) the CipherNet classifier: 1) Prediction map obtained from (f) (in red: pixel blocks predicted as encrypted, $score(b_i) \neq 1$), 2) Reconstructed image from (e) after the initialization step, 3) Pixel block state map associated to (g)-2 (in black: $state(i) = 0$, in grey: $state(i) = 1$, in white: $state(i) = 2$), 4) Reconstructed image from (e) after the initialization step and one round of the correction step, 5) Pixel block state map associated to (g)-4 (in black: $state(i) = 0$, in grey: $state(i) = 1$, in white: $state(i) = 2$), 6) Final reconstructed image from (e) after the initialization step and the whole correction step (27 rounds for the local entropy based-classifier and 12 rounds for the CipherNet classifier) (framed in red: pixel blocks predicted as encrypted).

is the encrypted image associated with Fig. 9.a using the CFB-then-ECB encryption mode. Note that the content of the original image is not visible anymore, as indicated by a very low PSNR value ($8.55\ dB$). During the transmission, this encrypted image has been randomly noised with a BER of 2.6×10^{-3}, which randomly corrupts on average one bit every three pixel blocks (Fig. 9.e). This noise introduction has no impact on the confidentiality of the original image content (PSNR $= 8.54\ dB$). In addition, the encrypted image in Fig. 9.d and its noisy version in Fig. 9.e are quite different, as indicated by a PSNR of $33.54\ dB$. If the noisy encrypted image in Fig. 9.e is directly decrypted, as presented in Fig. 9.f, many pixel blocks are incorrectly decrypted ($5,028$ pixel blocks (31%), framed in red). Therefore, even knowing the secret key used during the encryption, the original image content cannot be recovered due to the noise. A PSNR of $13.73\ dB$ between Fig. 9.a and Fig. 9.f also highlight the necessity to apply our method of noisy encrypted image correction during the decoding phase. Note that incorrectly decrypted pixel blocks are always by pairs due to the CFB-then-ECB encryption mode. Indeed, when a pixel block is incorrectly decrypted due to noise corruption, its neighbor is also incorrectly decrypted due to noise spreading.

Fig. 9.g and Fig. 9.h illustrate the obtained results using our correction algorithm with the local entropy-based classifier and the CipherNet classifier respectively. Fig. 9.g-1 and Fig. 9.h-1 show the prediction maps obtained using the two classifiers on the directly decrypted image Fig. 9.f. We can see that all the incorrectly decrypted pixel blocks are correctly identified as still being encrypted. But, one can note that some pixel blocks in clear are predicted as encrypted, especially when the local entropy-based classifier is used. Indeed, these pixel blocks are part of those identified in Fig. 9.b and Fig. 9.c. Fig. 9.g-2 and Fig. 9.h-2 show the reconstructed images obtained from the noisy encrypted image (Fig. 9.e) after the initialization step, and Fig. 9.g-3 and Fig. 9.h-3 are their associated pixel block states maps. In Fig. 9.g-2 and in Fig. 9.h-2, all the pixel blocks identified as clear are displayed in clear and the probable incorrectly decrypted ones remain in their encrypted version. Note that some of the incorrectly predicted pixel blocks (see Fig. 9.g-1 and Fig. 9.h-1) are identified as being in clear using the information that both the previous and the following pixel

blocks are in clear. If we compare the sequence length of the probable incorrectly decrypted pixel blocks between Fig. 9.g-3 and Fig. 9.h-3, we can observe that the CipherNet classifier is more efficient than the local entropy-based classifier. Moreover, the maximal sequence length indicates the necessary number of rounds to correct the whole noisy encrypted image during the second step of our algorithm (correction step). Fig. 9.g-4 and Fig. 9.h-4 are the reconstructed images obtained from the noisy encrypted image (Fig. 9.e) after one round of the correction step, and Fig. 9.g-5 and Fig. 9.h-5 are their associated pixel block state maps. We can see that all the pixel blocks in Fig. 9.g-2 and Fig. 9.h-2 such that $state(i) = 1$ (*i.e.* represented in grey on the pixel block state maps) are then correctly decrypted and represented in clear in Fig. 9.g-4 and Fig. 9.h-4. Moreover, in Fig. 9.g-5 and Fig. 9.h-5, their state is put to zero ($state(i) = 0$) and the state of the previous block is updated to 1 ($state(i - 1) = 1$). Fig. 9.g-6 and Fig. 9.h-6 are the final reconstructed images after the whole correction. On the one hand, Fig. 9.g-6 is obtained using the local entropy-based classifier after 27 rounds of the correction step. We can see that most of the pixel blocks (99.93%) are correctly reconstructed, but 12 highly textured ones in clear (actually 6 pairs of pixel blocks) remain incorrectly decrypted. The PSNR between the reconstructed image and the original image in Fig. 9.a is equal to $40.43\ dB$. On the other hand, Fig. 9.h-6 is obtained using the CipherNet classifier after only 12 rounds in the correction step. In this case, all the pixel blocks are correctly recovered and the final reconstructed image is exactly the same as the original image in Fig. 9.a (PSNR $\rightarrow +\infty$). This highlights the fact that using the CipherNet classifier in our proposed algorithm is the best choice to correct noisy encrypted images.

D. Obtained results on a large image database

We have applied our proposed method of correction of noisy encrypted images on 100 grey-level images (512×512 pixels) randomly chosen in the BOWS-2 database [1]. These images have strong statistical variabilities in their content. Table IV shows the obtained results with the two different classifiers. Whatever the used classifier, we can see that most of the pixel blocks from noisy encrypted images are correctly reconstructed. Indeed, on average, 99.53% of the pixel blocks using the local entropy-based classifier and 99.85% of the pixel blocks using the CipherNet classifier are correctly reconstructed. Moreover, in the worst case scenario (very textured images), more than 92% of the pixel blocks are still correctly reconstructed (92.96% using the local entropy-based classifier and 95.80% using the CipherNet classifier). We can also remark that 26 images out of 100 using the local entropy-based classifier and 51 images out of 100 using the CipherNet classifier are perfectly reconstructed, which means that there are no errors at all. Finally, we can conclude that, as already shown previously, the CipherNet classifier is more efficient in our application case.

Fig. 10 illustrates examples of image areas which are difficult to reconstruct using both classifiers. Note that, in these areas, the pixel blocks of the original images do not obtain a classification score equal to 1. Therefore, the associated reconstructed pixel blocks are those whose score is maximal among all the possible configurations. As illustrated in the first and second rows, the CipherNet classifier is more efficient than the local entropy-based classifier to discriminate clear blocks with a pattern and encrypted blocks. However, in the third row, we can see that for text, the results obtained with the local entropy-based classifier are more encouraging than with the CipherNet classifier. In fact, the quantization step before the local entropy computation allows us to significantly improve the obtained results for these kind of patterns. Finally, the last row shows a very textured area (leaves of a tree) in the original image. We can observe that the two classifiers do not allow us to perfectly reconstruct all the pixel blocks. Moreover, we can also notice that the incorrectly predicted pixel blocks are not the same when using the two classifiers. As a conclusion, it could be interesting to combine them together to achieve better results.

	Classifier	
	Local entropy-based	CipherNet
Min.	92.96	95.80
Average	99.53	99.85
Q1	99.51	99.94
Median	99.91	100
Q3	100	100
# perfectly reconstructed images	26	51

TABLE IV: Percentage of correctly reconstructed pixel blocks per image using our method with two different classifiers (results obtained using 100 images (512×512 pixels, *i.e.* 16,384 blocks) randomly chosen in the BOWS-2 database [1]).

E. Performance comparisons with other image encryption methods

We have compared the efficiency of error correction for images encrypted using the proposed CFB-then-ECB mode-based image encryption method with those encrypted using standard encryption approaches: XOR encryption [20], [36], [37], pixel scrambling [9], [13], [35] and pixel block scrambling [4], [28], [29]. For this purpose, we carried out our experiments on 100 images (512×512 pixels) of the BOWS-2 database [1]. In Table V, we can note that there is no effective method, from our knowledge, for correcting encrypted images using these more standard encryption methods in the case of noise introduction. We then performed the correction in the following way:

- We consider all the pixels of the encrypted image as potentially noisy;
- It is assumed that at most one bit per pixel has been altered by the noise;
- Consequently, the correction is carried out by examining all the possible configurations for each pixel. Finally, the configuration which is the most correlated with the previously reconstructed neighboring pixels is considered as the original clear value.

In contrast, images encrypted using the proposed CFB-then-ECB mode-based image encryption method are corrected with

	PSNR (dB)			NPCR (%)			UACI (%)		
	Min.	Average	Max.	Min.	Average	Max.	Min.	Average	Max.
XOR encryption	21.48	30.29	35.86	11.61	41.87	75.09	0.29	1.41	4.52
Pixel scrambling	21.49	30.29	36.04	11.64	41.87	75.09	0.29	1.41	4.53
Pixel block scrambling	21.49	30.28	35.74	11.66	41.87	75.06	0.29	1.41	4.52
Ours (Local entropy-based)	20.50	50.22	$+\infty$	92.99	99.53	100.00	0.00	0.14	2.04
Ours (CipherNet)	23.39	70.69	$+\infty$	95.82	99.85	100.00	0.00	0.04	1.13

TABLE V: Performance comparisons between our proposed method and standard image encryption approaches.

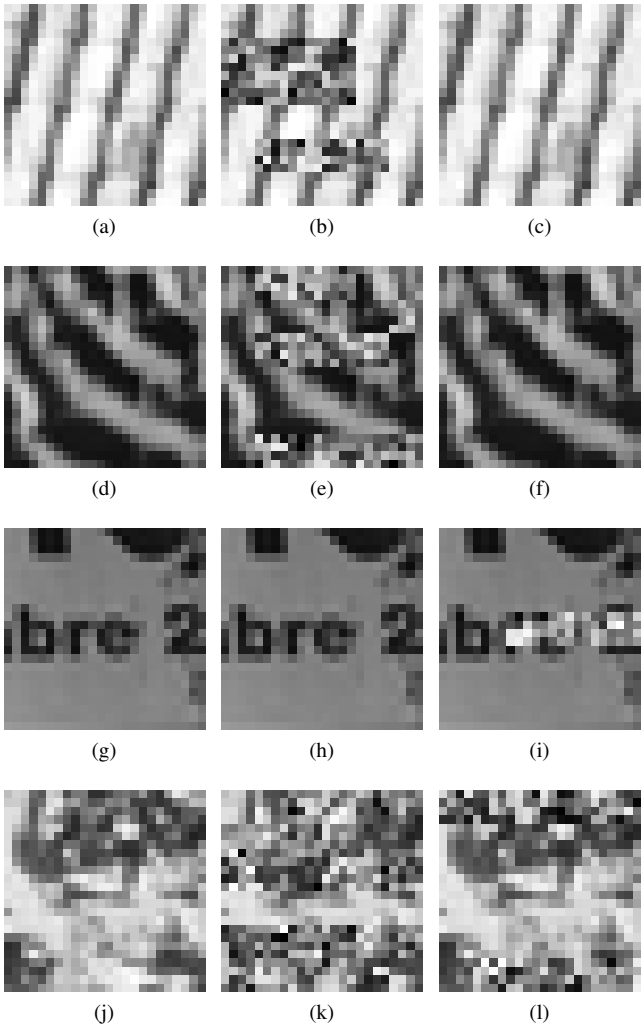


Fig. 10: Comparisons of the obtained results using our method with two different classifiers: rows) different cropped images of size 24×24 pixels from images of the BOWS-2 database [1]; crops from: first column) original images, second column) reconstructed images using the local entropy-based classifier, third column) reconstructed images using the CipherNet classifier.

the correction method described in Section III-B, using the local entropy-based or the CipherNet classifier. Performance evaluation is completed using three statistical metrics widely used in the field of image encryption: PSNR (in dB), NPCR (in $\%$) and UACI (in $\%$) [33] between the original image and the reconstructed decrypted image after correction. The higher the PSNR value is, the better the quality of the reconstruction is. The NPCR rate corresponds to the amount of correctly recovered pixels in the reconstructed decrypted image after correction. The UACI rate is computed by observing the intensity differences between pixels from the original image and from the reconstructed decrypted image after correction. Thereby, a value close to 0% indicates a strong similarity between the pixels of the two images. As shown in Table V, we can see that the PSNR value is quite low for standard image encryption approaches. Indeed, an average value of approximately 30 dB indicates that the reconstructed image after correction is similar, but not as close to the original image. However, with our proposed CFB-then-ECB mode-based image encryption method, the correction is quite efficient whatever the used classifier, as indicated by a PSNR value of 50.22 dB using the local entropy-based classifier and of 70.69 dB using the CipherNet classifier. Concerning the NPCR value, we can see that it is very close to 100% with our proposed method, whereas it is less than 50% with the standard image encryption approaches. This ensures the superiority of our proposed method. Finally, the UACI rate is close to 0% for all the other compared approaches, even if, once again, our method achieves better results with an average value of 0.14% using the local entropy-based classifier and of 0.04% using the CipherNet classifier to perform the correction. In conclusion, our proposed method outperforms the more standard image encryption methods for noisy encrypted image correction.

Concerning the correction of noisy encrypted using the AES cryptosystem images, we have compared our proposed method with three other size preserving approaches: a standard error correcting approach based on Reed-Solomon (RS) codes [16] without size expansion, the method of Puteaux and Puech [23] and that of Islam *et al.* [12]. For a grey-level image with a size of 512×512 pixels encrypted using AES, in order to be able to correct 1 bit per block of 4×4 pixels, RS(255, 251) codes with symbols of 8 bits are used. In order to prevent a size expansion of 4 kB, some bits of the encrypted image are replaced by the computed error correcting codes. As a result, 98.47% of the pixel blocks are correctly reconstructed. However, with our new proposed method, as presented in Table IV, we achieve better results with more than 99.5% of correctly reconstructed pixel blocks with both classifiers. Regarding the approach of Puteaux and Puech [23], it should be noted that the CFB-then-ECB encryption mode is securer than the ECB encryption mode. This makes our new proposed method of noisy encrypted image correction more suitable for real life applications than [23]. According to Table IV, our proposed method achieves perfect recovery of 26 images using the local entropy-based classifier and 51 using the CipherNet classifier, while the Islam *et al.* approach [12] applied to grey-level images only succeeds to perfectly reconstruct 37 images in the best case scenario.

V. Conclusion

In this paper, we have described an efficient method of noisy encrypted image correction relying on a new CFB-then-ECB mode-based image encryption technique. The CFB-then-ECB encryption mode is based on a combination of CFB mode and ECB mode for AES encryption. Using this new encryption mode, we have shown that if one encrypted pixel block is noised, both the current and the following pixel blocks will be incorrectly reconstructed during the decryption phase. This noise spreading is then exploited to help the correction of noisy encrypted images. Indeed, two incorrectly decrypted neighboring pixel blocks are easier to identify than a single one. Our new proposed noisy encrypted image correction algorithm is based on two main steps, which are the initialization step and the pixel block analysis and correction step. A classifier to discriminate clear from encrypted pixel blocks is involved in both steps. With more than 99.5% of correctly reconstructed pixel blocks on average whatever the classifier used, our new proposed method is efficient to correct noisy encrypted images.

According to our experiments, it could be interesting to combine the results we obtained with the two classifiers in order to further improve the correction performances. Indeed, we do not achieve to perfectly recover the original image content after decryption every time, the results of incorrectly reconstructed pixel blocks are not the same using the two classifiers. In future work, we could also extend the proposed correction framework to noisy encrypted color images. In this case, the correlation between RGB components could also be exploited during the pixel block analysis and correction algorithm to achieve better results. In addition, in this current work, we consider that only one bit per encrypted pixel block is corrupted due to noise introduction. Furthermore, if the noise amount is more important or if it is not uniformly distributed, more than one bit per encrypted pixel block could be flipped. Therefore, we could also investigate this last point in the future.

Acknowledgment

We would like to thank the financial support of the ANR-16-DEFA-0001 OEIL (statistiques rObustEs pour l'apprentIssage Léger) research project of the French ANR/DGA challenge DEFALS (DEtection de FALSifications dans des images).

References

[1] P. Bas and T. Furon, "Image database of BOWS-2," http://bows2.ec-lille.fr/.

[2] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons & Fractals*, vol. 21, no. 3, pp. 749–761, 2004.

[3] G. Chen and T. Ueta, "Yet another chaotic attractor," *International Journal of Bifurcation and Chaos*, vol. 9, no. 07, pp. 1465–1466, 1999.

[4] T. Chuman, W. Sirichotedumrong, and H. Kiya, "Encryption-then-compression systems using grayscale-based image encryption for JPEG images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 6, pp. 1515–1525, 2018.

[5] M. Czapski and M. Nikodem, "Error detection and error correction procedures for the Advanced Encryption Standard," *Designs, Codes and Cryptography*, vol. 49, no. 1-3, pp. 217–232, 2008.

[6] J. Daemen and V. Rijmen, "AES proposal: Rijndael," *Original AES Submission to NIST*, 1999.

[7] Z. Erkin, A. Piva, S. Katzenbeisser, R. L. Lagendijk, J. Shokrollahi, G. Neven, and M. Barni, "Protection and retrieval of encrypted multimedia content: When cryptography meets signal processing," *EURASIP Journal on Information Security*, vol. 2007, p. 17, 2007.

[8] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[9] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *International Journal of Bifurcation and Chaos*, vol. 8, no. 06, pp. 1259–1284, 1998.

[10] Z.-H. Guan, F. Huang, and W. Guan, "Chaos-based image encryption algorithm," *Physics Letters A*, vol. 346, no. 1, pp. 153–157, 2005.

[11] X. Hu, W. Zhang, H. Hu, and N. Yu, "Non-local denoising in encrypted images," in *International Conference on Internet of Vehicles*. Springer, 2014, pp. 386–395.

[12] N. Islam, Z. Shahid, and W. Puech, "Denoising and error correction in noisy AES-encrypted images using statistical measures," *Signal Processing: Image Communication*, vol. 41, no. C, pp. 15–27, Feb. 2016.

[13] X. Kang and R. Tao, "Color image encryption using pixel scrambling operator and reality-preserving MPFRHT," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 7, pp. 1919–1932, 2018.

[14] P. Korshunov and T. Ebrahimi, "Scrambling-based tool for secure protection of JPEG images," in *IEEE International Conference on Image Processing*, 2014, pp. 3423–3425.

[15] Y. Mao, G. Chen, and S. Lian, "A novel fast image encryption scheme based on 3D chaotic baker maps," *International Journal of Bifurcation and Chaos*, vol. 14, no. 10, pp. 3613–3624, 2004.

[16] R. H. Morelos-Zaragoza, *The art of error correcting coding*. John Wiley & Sons, 2006.

[17] V. Ocheretnij, G. Kouznetsov, M. Gossel, and R. Karri, "On-line error detection and bist for the AES encryption algorithm with different S-box implementations," in *IEEE International On-Line Testing Symposium*, 2005, pp. 141–146.

[18] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 1999, pp. 223–238.

[19] A. Pedrouzo-Ulloa, J. R. Troncoso-Pastoriza, and F. Pérez-González, "Image denoising in the encrypted domain," in *IEEE International Workshop on Information Forensics and Security*, 2016, pp. 1–6.

[20] M. Preishuber, T. Hütter, S. Katzenbeisser, and A. Uhl, "Depreciating motivation and empirical security analysis of chaos-based image and video encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2137–2150, 2018.

[21] P. Premaratne and M. Premaratne, "Key-based scrambling for secure image communication," in *International Conference on Intelligent Computing*. Springer, 2012, pp. 259–263.

[22] W. Puech and J. M. Rodrigues, "Crypto-compression of medical images by selective encryption of DCT," in *European Signal Processing Conference*, 2005, pp. 1–4.

[23] P. Puteaux and W. Puech, "Noisy encrypted image correction based on Shannon entropy measurement in pixel blocks of very small size," in *European Signal Processing Conference*, 2018.

[24] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[25] S. M. SaghaianNejadEsfahani, Y. Luo, and S.-C. S. Cheung, "Privacy protected image denoising with secret shares," in *IEEE International Conference on Image Processing*, 2012, pp. 253–256.

[26] W. Trappe and L. C. Washington, *Introduction to cryptography with coding theory*. Pearson Education India, 2006.

[27] K. Usman, H. Juzoji, I. Nakajima, S. Soegidjoko, M. Ramdhani, T. Hori, and S. Igi, "Medical image encryption based on pixel arrangement and random permutation for transmission security," in *IEEE International Conference on e-Health Networking, Application and Services*, 2007, pp. 244–247.

[28] D. Van De Ville, W. Philips, R. Van de Walle, and I. Lemahieu, "Image scrambling without bandwidth expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 6, pp. 892–897, 2004.

[29] O. Watanabe, A. Uchida, T. Fukuhara, and H. Kiya, "An encryption-then-compression system for JPEG 2000 standard," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 1226–1230.

[30] S. B. Wicker, *Error control systems for digital communication and storage*. Prentice hall Englewood Cliffs, 1995, vol. 1.

[31] C. V. Wright, W.-C. Feng, and F. Liu, "Thumbnail-preserving encryption for JPEG," in *ACM Workshop on Information Hiding and Multimedia Security*, 2015, pp. 141–146.
[32] K. Wu, R. Karri, G. Kuznetsov, and M. Goessel, "Low cost concurrent error detection for the advanced encryption standard," in *IEEE International Test Conference*, 2004, pp. 1242–1248.
[33] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI randomness tests for image encryption," *Cyber journals: multidisciplinary journals in science and technology, Journal of Selected Areas in Telecommunications*, vol. 1, no. 2, pp. 31–38, 2011.
[34] T. Xiang, K.-W. Wong, and X. Liao, "Selective image encryption using a spatiotemporal chaotic system," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 17, no. 2, p. 023115, 2007.
[35] G. Ye, "Image scrambling encryption algorithm of pixel bit based on chaos map," *Pattern Recognition Letters*, vol. 31, no. 5, pp. 347–354, 2010.
[36] J.-C. Yen, H.-C. Chen, and S.-M. Wu, "Design and implementation of a new cryptographic system for multimedia transmission," in *IEEE International Symposium on Circuits and Systems*, 2005, pp. 6126–6129.
[37] J.-C. Yen and J.-I. Guo, "A new image encryption algorithm and its VLSI architecture," in *IEEE Workshop on Signal Processing Systems. Design and Implementation (Cat. No. 99TH8461)*, 1999, pp. 430–437.
[38] Z.-L. Zhu, W. Zhang, K.-W. Wong, and H. Yu, "A chaos-based symmetric image encryption scheme using a bit-level permutation," *Information Sciences*, vol. 181, no. 6, pp. 1171–1186, 2011.

Pauline Puteaux received her M.S. degree in Computer Science and Applied Mathematics, with specialization in Cybersecurity, from the University of Grenoble, France, in 2017. She is currently pursuing her Ph.D. degree with the Laboratory of Informatics, Robotics and Microelectronics of Montpellier, France. Her work has focused on multimedia security, and in particular, image analysis and processing in the encrypted domain. Since 2016, she has published 3 journal papers and 5 conference papers. She is a reviewer for Signal Processing (Elsevier), J. of Visual Communication and Image Representation (Elsevier), IEEE Trans. on Circuits & Systems for Video Technology and IEEE Trans. on Dependable and Secure Computing.

William Puech received his diploma of Electrical Engineering from the Univ. Montpellier, France (1991) and a Ph.D. Degree in Signal-Image-Speech from the Polytechnic National Institute of Grenoble, France (1997) with research activities in image processing and computer vision. He served as a Visiting Research Associate to the University of Thessaloniki, Greece. From 1997 to 2008, he has been an Associate Professor at the Univ. Montpellier, France. Since 2009, he is full Professor in image processing at the Univ. Montpellier, France. His current interests are in the areas of image forensics and security for safe transfer, storage and visualization by combining data hiding, compression and cryptography. He is the head of the ICAR team (Image & Interaction) in the LIRMM, has published more than 45 journal papers and 140 conference papers and is associate editor for 5 journals (JASP, SPIC, SP, JVCIR and IEEE TDSC) in the areas of image forensics and security. Since 2017, he is the general chair of the IEEE Signal Processing French Chapter and since 2018, he is a member of the IEEE Information Forensics and Security TC.