A Secret JPEG Image Sharing Method Over GF(2 M) Galois Fields

Pauline Puteaux, Félix Yriarte, William Puech

A Secret JPEG Image Sharing Method Over $GF(2^M)$ Galois Fields

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

***Abstract*— With the rise of exchanges over the cloud and social networks, JPEG images have taken an important place in world image transmission and storage. In order to avoid security breaches and combat threats on the Internet, numerous JPEG image security methods have been proposed in both the academic and industrial communities. Encryption methods have been specifically designed to make JPEG images visually secure using so called crypto-compression methods. The drawback of crypto-compression is that it depends on only one secret key. Indeed, if this key is lost, so is the totality of the content of the secret original image. In this paper, we propose a secret JPEG image sharing approach. Shamir's secret sharing scheme over Galois fields is used during the JPEG Huffman coding step, ensuring the visual security of the secret image in the compressed domain, while solving the issue of secret key loss. In addition, we also describe an eco-friendly scenario, dealing with a public shared JPEG image. In this scenario, we can obtain compressed shares because there is no need to duplicate the redundant information. According to our obtained results, our approach is fully format compliant and size preserving when compared to a standard JPEG compression of a secret original image, while ensuring the visual security of the content of the secret original image.**

***Index Terms*—Multimedia security, privacy protection, secret image sharing, JPEG compression.**

I. Introduction

In recent years, with the constant evolution of the Internet, digital images have taken a major part in data transmission. This multimedia data must be compressed and protected against illegal access and fraudulent usage. Social networks are particularly targeted, with more than two billion active users worldwide. Images passing through these networks are usually stored in the JPEG format and are personal.

JPEG (Joint Photographic Experts Group) is the most used image compression method, as it is fast and efficient [1], [2]. Seeing as it reliably offers high compression rates while keeping low distortion, the vast majority of web browsers, operating systems and smartphones, use the JPEG standard. For these reasons, being able to protect the visual security of images while complying with the compressed JPEG format has aroused the scientific community's interest. In particular, methods of crypto-compression have been developed to perform encryption in such a way that the encrypted data can always be represented in a standard format (format compliance property), while ensuring visual confidentiality [3]–[5].

P. Puteaux is with the Centre de Recherche en Informatique, Signal et Automatique de Lille (CRIStAL), Centre National de la Recherche Scientifique, Univ. Lille, Centrale Lille, Lille, France (e-mail: pauline.puteaux@cnrs.fr). F. Yriarte and W. Puech are with the Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM), Centre National de la Recherche Scientifique, Univ. Montpellier, Montpellier, France (e-mail: felix.yriarte.etu@univ-lille.fr; william.puech@lirmm.fr).

In cryptography, the secret key is an essential aspect of a protection scheme [6]. However, if the secret key is stolen or lost, the original data cannot be recovered. Moreover, privacy protection for everyone when sharing data involving multiple users is a major issue.

For example, Alice takes an image with her friends and publishes it on her personal page. Everyone in her social network then has access to this image following the privacy settings she has defined herself. However, her friends, who are depicted on the image are not part of the publication procedure. Indeed, they are not consulted and have not given their consent before the image publication.

In order to deal with these issues, Shamir [7] and Blakley [8] independently proposed two secret sharing schemes in 1979. In a (k, n) secret sharing scheme, n participants share a common secret. Each of them receives a personal shared value (called *share*). If k over n participants gather their shares, the trust level within the group is achieved and the secret is reconstructed. In Shamir's secret sharing scheme, a random secret k-order polynomial with coefficients in $\mathbb{Z}/p\mathbb{Z}$ (with p prime) is generated. The zero-order coefficient of this polynomial is then substituted with the secret value which is considered as an element in $\mathbb{Z}/p\mathbb{Z}$. A different evaluation of this polynomial is then given to each participant, according to an associated ID value. As at least k participants get together, a Lagrange polynomial interpolation can be used to recover the secret polynomial, and thus the secret. In 2002, Thien and Lin proposed a secret sharing method applied to images [9]. They adapted Shamir's method by applying a secret sharing scheme to every pixel's value to share a secret image. Since the maximum gray-level value of a pixel is 255, they considered polynomials over a $\mathbb{Z}/251\mathbb{Z}$ finite field, since p – the size of a $\mathbb{Z}/p\mathbb{Z}$ finite field – has to be prime. When using $\mathbb{Z}/251\mathbb{Z}$ field, gray-level values ranging from 251 to 255 can not directly be processed by such a secret sharing scheme. They can either be reduced to 250, which leads to information loss or not be processed, which decreases the proportion of pixels secured by the secret sharing scheme. Contrary to $\mathbb{Z}/p\mathbb{Z}$ finite fields, Galois fields can be generated of size p^M, with p prime and M an integer. Since 2 is prime, any binary word of size M can be processed in a secret sharing scheme over a Galois field $GF(2^M)$.

In this paper, we propose a joint JPEG compression and secret image sharing scheme over Galois fields, which is fully format compliant and preserves compression rate. During the

JPEG coding step, binary words $\mathbf{A}_i$ (of which the size is noted $|\mathbf{A}_i|$, where $|.|$ is the size operator) are formed by concatenating the non-zero quantized AC coefficients separately for each block, and fed to a Shamir secret sharing scheme over a Galois field of size $2^{|\mathbf{A}_i|}$. We then substitute the shared values with the associated bits of the secret frequency blocks amplitude values, to obtain shared blocks. Then, we also consider an eco-friendly scenario in which a public shared JPEG image is created, containing the redundant data, along with the reduced shares. This scenario largely decreases the storage size of the shares, while maintaining the security standard that would be otherwise obtained.

The rest of this paper is organized as follows. In Section II, we detail current state-of-the-art methods, related to secret image sharing and JPEG compression. Section III describes our proposed method of secret JPEG image sharing. Section IV presents experimental results and finally, this paper is concluded in Section V.

II. Previous work

In this Section, we present methods developed to ensure multimedia visual security. In Section II-A, we first detail current state-of-the-art secret sharing for multimedia security methods, specifically suitable for images or 3D objects. In Section II-B, we describe some approaches dealing with JPEG image security, in particular significant crypto-compression methods.

A. Secret sharing for multimedia security

There are two categories of methods for sharing a visual secret, namely visual cryptography proposed by Naor and Shamir in 1994 [10] and secret image sharing proposed for the first time by Thien and Lin in 2002 [9].

Visual cryptography consists in securely sharing visual information (text or image) between several people. In the approach [10] developed for binary images, two images called shares are generated after the sharing process. The black pixels of the secret image are protected and the white pixels are made random. The secret is then reconstructed by performing an exclusive-or between the two shares. Other methods have been proposed to share not only binary, but also gray level images [11]–[13]. They can also be extended to color images.

Secret image sharing is inspired by the methods developed independently in 1979 by Blakley [8] and Shamir [7]. Generally based on the concept of polynomial interpolation, it allows the sharing of an image between n users in a secure way. Each user receives a personal share in the form of an image. This share is unique and visually appears to have been generated randomly. The secret original content can only be reconstructed after gathering at least k of these shares with $k \leq n$. With $k-1$ shares, it is indeed impossible to obtain any information. The parameter k can be higher or lower depending on the level of trust within the sharing group. Note that the reconstructed secret image I' is very similar to the secret original image I and can even be strictly identical to I with so-called perfect methods. The secret image sharing process is illustrated in Fig. 1.

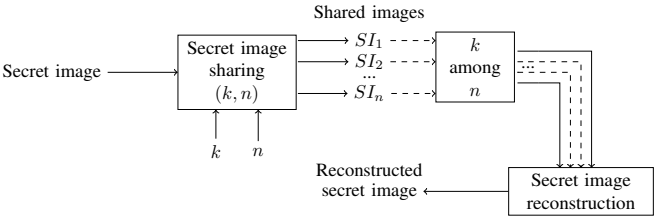


Fig. 1: Overview of a (k, n) image secret sharing scheme.

Thien and Lin proposed to use Shamir's method to share secret image pixel values [9]. Groups of k pixels are formed to substitute the coefficients of a polynomial over a finite field defined by a prime number. In order to approximate the maximal gray level pixel value of 255, the authors chose to use the prime number 251. Since the pixel values between 251 and 255 can not be covered, pixel values are truncated: the secret image sharing scheme is then lossy. A lossless version can be obtained by storing the overflow pixel values as additional information. Nevertheless, in this case, the size of the shared image is increased when compared to the secret original image. In order to deal with this problem, Yang *et al.* suggested using the Galois field $GF(2^8)$ in the calculation of polynomials [14]. In this way, their scheme is lossless and avoids size expansion. Beugnon *et al.* proposed a hierarchical secret image sharing scheme for social networks in order to answer the problem of multi-party privacy protection conflicts [15]. Based on the disjunctive multi-level approach of Belenkiy [16] applied to images and the use of the Galois field $GF(2^8)$, this solution ensures user privacy by securing their faces. Recently, Qin *et al.* described two reversible data hiding schemes in encrypted image using Shamir's secret sharing scheme over Galois fields $GF(p)$ and $GF(2^8)$ [17].

More recently, other methods have been designed for secret 3D object sharing [18]–[21]. Elsheh and Hamza proposed to directly apply the secret sharing schemes of Blakley [8] and Thien and Lin [9] to 3D objects in order to ensure their visual confidentiality [18]. Anbarasi and Mala described an approach to share several 3D objects in parallel with a verification system to detect forged shares during their reconstruction [19]. Lee *et al.* presented a method to obtain a group of n 3D objects when at least k of them have been downloaded in high quality in a streaming context [20]. Beugnon *et al.* designed a format-compliant selective secret 3D object sharing method [21]. Relevant bits within vertices coordinates are selected to be secured and the shared information substitutes original bits in the shared 3D object.

B. JPEG image security

For 30 years, JPEG [1] (Joint Photographic Experts Group) compression has been the most widely used compression format for storing and exchanging digital images since its last standard ISO/IEC 10918-1 ITU-T Recommendation T.81. was published in 1993. Currently, its high usage is mainly due to historical

reasons and is still preferred in many applications. Indeed, the vast majority of displays, including mobile devices, are compatible with the many versions and extensions of JPEG (like JPEG XR or JPEG XT, for example). In addition, the Independent JPEG Group (IJG) writes and distributes a free library (libjpeg), which is powerful and regularly updated, and is widely used [2]. JPEG efficiency results from lossy compression and the use of minimal redundancy codes.

In recent years, interest in JPEG image security has grown. In particular, the goal of crypto-compression is to perform encryption in such a way that the encrypted data can always be represented in a meaningful format (format compliance property), while ensuring visual confidentiality. In the JPEG format, not all binary data needs to be encrypted. When decompressing without the encryption key, the image in the spatial domain is then selectively encrypted. For these kinds of methods, JPEG compression and encryption are performed together. We can list four categories of methods: sign bit encryption, substitution encryption, permutation encryption and hybrid encryption.

Shi and Bhargava designed one of the first crypto-compression approaches to directly encrypt the binary JPEG stream [22]. The authors propose to encrypt the sign bit of AC and DC coefficients (the sign of the difference for DC coefficients). A binary pseudo-random sequence is generated according to a secret key and the encryption is performed by an exclusive or-operation of this sequence with the sequence of all sign bits. This method preserves the JPEG structure of the image, which can be viewed with standard image editors. In addition, the compression ratio is not (or only slightly) changed since the number of bits is unchanged. However, this method has been shown to be insecure by Said [23].

Substitution-based crypto-compression is a class of methods that substitutes clear values with their encrypted versions. In general, the values are encrypted using the exclusive-or operator between the clear values and a pseudo-random sequence generated with a secret key. This operation allows it to remain compatible with the format and to limit the expansion of the file size. Van Droogenbroeck and Benedett proposed to encrypt the AC coefficients after the DCT transformation, but not the DC coefficients, because they contain important visible information and are predictable [3]. Following this idea, Puech and Rodrigues proposed a selective encryption method for JPEG images that encrypts both DC and AC coefficients [4]. In this method, all the codes for the (optional) DC coefficients and the non-zero AC coefficients are concatenated to form a 128 bit stream. This bitstream is then encrypted using the AES algorithm [24] in OFB mode. Finally, the JPEG image is constructed with the encrypted coefficient values. Rodrigues *et al.* extend this concept by proposing an encryption which is performed on automatically detected regions of interest, such as human skin [25]. These regions are detected using the plaintext DC coefficients of the chrominance components. The selective encryption is applied to the blocks of the luminance component, during the entropy coding phase.

Permutation-based crypto-compression consists in scrambling the elements of an image using a pseudo-random number generator. Kurihara *et al.* designed an encryption scheme followed by compression in which the blocks are permuted in the spatial domain [26]. This scheme can be attacked using a puzzle solver, as demonstrated by [27]. Li and Yuan proposed the Full Inter-Block Shuffle (FIBS) method, which consists of scrambling coefficients of the same frequency between blocks [5]. He *et al.* presented a crypto-compression method with improved format compatibility and file size preservation [28]. This scheme is based on permutations of DC coefficient codes, AC coefficient codes and, finally, full MCUs.

In order to introduce the confusion and diffusion properties jointly, some authors have turned to hybrid approaches. Minemura *et al.* swapped JPEG blocks [29]. Within each block, the non-null AC coefficients are then permuted. Then, the sign bits are encrypted performing an exclusive-or operation with a pseudo-random sequence to increase the perturbations. Finally, the DC coefficients are grouped by close values using frequency domain edge detection for processing. Unterweger and Uhl describe a crypto-compression method based on three steps [30]. The first step is to swap the code order of the coefficients using the AES algorithm [24] in OFB mode. The bits of the coefficient values are then inverted according to the pseudo-random sequence used in the previous step. Finally, the order of the blocks is pseudo-randomly changed.

In addition, methods for image analysis and processing in the crypto-compressed domain have been described. In this context, neither the content of the plaintext data nor the encryption key used is known. Methods of data hiding in JPEG crypto-compressed images have been designed [31]–[36]. In these approaches, additional data is embedded directly into the crypto-compressed domain, thus preserving the JPEG structure. For example, Qian *et al.* used Huffman code mapping [34] and He *et al.* exploited invariant run-lengths [35]. Moreover, an effective way to recompress JPEG crypto-compressed images has been described [37]. To perform a recompression, each quantized encrypted frequency coefficient is divided by two at each recompression step. In this way, the least significant bit of these coefficients are removed and some coefficients become equal to zero.

Crypto-compression methods, like all encryption methods, have certain shortcomings, like depending on a key and a single container in which the secret information is embedded. This container can be lost, destroyed or altered during an attack, making it impossible to recover the original content in clear. Secret sharing solves these problems. For this reason, in the rest of this paper, we develop the first method (to our knowledge) of secret image sharing along with JPEG compression.

III. Proposed method of joint secret image sharing and JPEG compression

In this section, we develop our proposed method of joint secret image sharing and JPEG compression. The main objective of this work is maximising the ratio between image distortion (to ensure visual security) over the compression

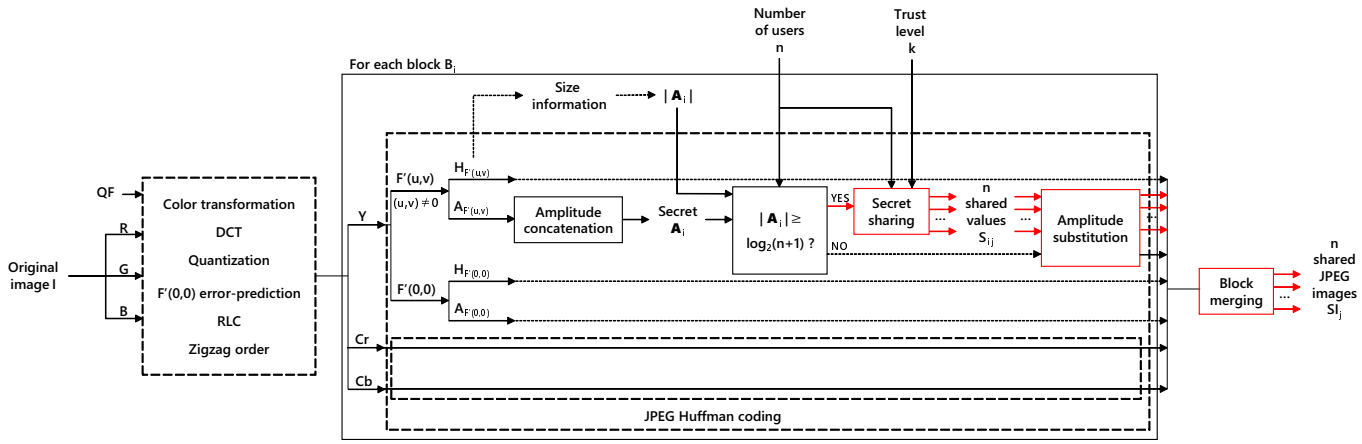


Fig. 2: Overview of the proposed method of joint secret image sharing and JPEG compression.

rate. The proposed method is format compliant, preserves the compression rate of JPEG and allows us to reconstruct a secret JPEG image, which is exactly the same as the one obtained by applying a standard JPEG compression. As presented in Fig. 2, the secret sharing of the non-zero quantized AC coefficients is jointly performed during the Huffman coding stage of the JPEG compression. For each block Minimum Code Unit (MCU), a binary word is obtained by concatenating the amplitude parts. Each binary word is shared using the secret sharing scheme of Shamir [7] over the appropriate Galois field, according to its size. Then, the bits of the shared value substitute those of the secret block amplitude values. Due to the adaptation to the frequency block sizes, there is no size expansion and the JPEG structure is preserved: the proposed method is fully format compliant.

The encoding phase of our proposed approach is described in Section III-A, whereas the decoding phase is detailed in Section III-B. Finally, as an extension, a more efficient scenario in terms of global compression rate using a public shared JPEG image and reduced shares is presented in Section III-C.

A. Encoding phase

Let us consider a secret original (raw bitmap) RGB image. The secret sharing is completed jointly during the JPEG compression of the secret original image which is considered as a secret image to share. The first step of JPEG consists in applying a color transformation from RGB (Red, Green, Blue) to YCrCb space (luminance/chrominance space, Y for the luminance and Cr/Cb for the chrominance). The Discrete Cosine Transform (DCT) and quantization steps, considering a given quality factor QF, are then performed separately by blocks of 8×8 pixels on the three components Y, Cr and Cb. Each frequency block is then encoded using a MCU obtained by scanning the quantized DCT coefficients in a zigzag order, according to their increasing spatial frequency. A specific coding is then used to encode the quantized DC coefficient $F'(0,0)$. For the quantized AC coefficients $F'(u,v)$ (with $0 \leq u,v < 8$ and $(u,v) \neq (0,0)$), a run-length coding algorithm is applied to compress the consecutive coefficients equal to zero. Each non-zero quantized AC coefficient $F'(u,v)$ is encoded using a pair $(H_{F'(u,v)},\ A_{F'(u,v)})$. The header part $H_{F'(u,v)}$ contains the run-length $RL_{F'(u,v)}$ and the amplitude size parameter $size_{F'(u,v)}$. The amplitude part $A_{F'(u,v)}$ represents a code for the coefficient value. In this paper, we present the secret sharing method on the luminance component only, for a better understanding. However, it could be applied in the same way to the chrominance components.

The secret sharing scheme we apply is directly derived from the method proposed by Shamir in 1979 [7]. It is adapted to be suitable along with a JPEG compression, while allowing format compliance and size preservation. A secret image has to be shared between n participants. These participants define a trust level k within their group. They choose this threshold such that $1 < k \leq n$, which indicates the minimum number of users necessary to reconstruct the secret JPEG image. For each participant, a unique ID x_j, with $1 \leq j \leq n$ and such that $1 \leq x_j \leq n$, is associated and n shared JPEG images SI_j are then generated as illustrated in Fig. 2.

In order to share the content of a frequency block B_i, with $0 \leq i < N$, N the number of frequency blocks in the secret image, all the amplitude parts $A_{F'(u,v)}$ such that $(u,v) \neq (0,0)$ of the associated MCU are concatenated to form a binary word $\mathbf{A}_i$. One can note that the binary size of $\mathbf{A}_i$ (noted $|\mathbf{A}_i|$, where $|.|$ is the size operator) can be expressed using the size of each amplitude contained in the header part:

$$|\mathbf{A}_i| = \sum_{\forall (u,v)\neq(0,0) \ | \ F'(u,v)\neq(0,0)} size_{F'(u,v)}. \qquad (1)$$

In order to preserve the original size of the JPEG compressed image, each binary word $\mathbf{A}_i$ is then considered as a secret value over $GF(2^{|\mathbf{A}_i|})$ of the block B_i to share between the n participants.

If $|\mathbf{A}_i| \geq \log_2(n+1)$, a random sequence of values a_l, with $1 \leq l \leq k$ and such that $a_l \in GF(2^{|\mathbf{A}_i|})$, is generated. These values are then used to define a $(k-1)$-order polynomial with values in the Galois field $GF(2^{|\mathbf{A}_i|})$:

$$f_i(x) = \left(\sum_{l=1}^{k-1} a_l x^l \right) + \mathbf{A}_i. \qquad (2)$$

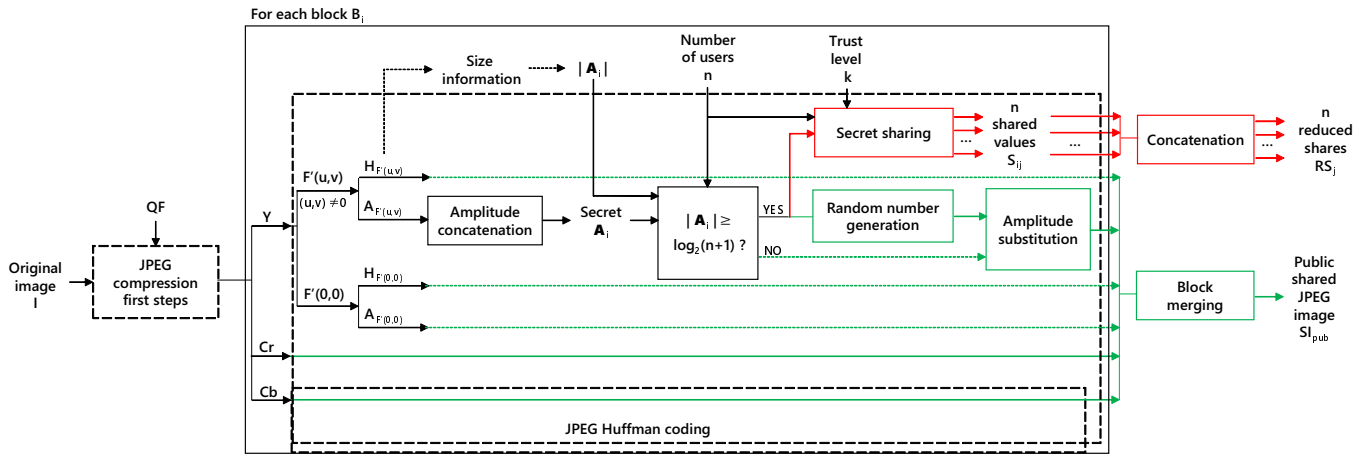


Fig. 3: Overview of the scenario using a public shared JPEG image.

Using this polynomial, for each user of ID x_j, each shared value S_{ij} associated to $\mathbf{A}_i$ is computed as $f_i(x_j)$, with $1 \leq j \leq n$. One can note that each S_{ij} is encoded using exactly the same number of bits as $\mathbf{A}_i$. Therefore, in order to obtain the shared version of the block B_i in a shared JPEG image SI_j, according to each amplitude size parameter $size_{F'(u,v)}$, amplitude values $A'_F(u,v)$ are substituted by associated bits in S_{ij} in order.

Otherwise, note that if $|\mathbf{A}_i| < \log_2(n+1)$, it is not possible to define a unique ID x_j per user because the number of distinct elements in $GF(2^{|\mathbf{A}_i|})$ is too small. Consequently, in this case, $\mathbf{A}_i$ can not be shared and so we propose that its value remains in clear. An overview of the full encoding scheme is presented in Fig. 2.

B. Decoding phase

The decoding phase consists of reconstructing the secret JPEG image from k different shared JPEG images SI_j. Therefore, it includes two main steps: the reconstruction of the secret JPEG image's amplitude values $A'_F(u,v)$ and the JPEG decompression.

During the reconstruction phase, two scenarios can be encountered. If k users over n are involved in the decoding process, Lagrange's interpolation is used to retrieve the secret values $\mathbf{A}_i$ associated to each frequency block B_i.

From each shared JPEG image $SI_{j'}$, with $1 \leq j' \leq k$, to reconstruct a binary word $\mathbf{A}_i$ such that $|\mathbf{A}_i| \geq \log_2(n+1)$, the associated shared value $S_{ij'} = f_i(x_{j'})$ is extracted by concatenating the amplitude values, in a similar way as conducted during the encoding phase. As $f_i(.)$ is a $(k-1)$-order polynomial function, one can note that its coefficients can be recovered as soon as k evaluations of this polynomial are known. In particular, the secret value $\mathbf{A}_i$ corresponds to the constant term of the polynomial $f_i(.)$ over $GF(2^{|\mathbf{A}_i|})$:

$$f_i(x) = \sum_{j'=0}^{k-1} f_i(x_{j'}) \times \prod_{\alpha=0,\ \alpha\neq j'}^{k-1} \frac{x - x_\alpha}{x_{j'} - x_\alpha}. \quad (3)$$

Note that, as the binary words $\mathbf{A}_i$ such that $|\mathbf{A}_i| < \log_2(n+1)$ remain in clear during the encoding phase, their associated bits are just copied without any modification during the reconstruction phase.

Otherwise, note that if $k' < k$ users met each other, the threshold k is not reached: the trust level in the group is not sufficient. As a consequence, the secret JPEG image cannot be reconstructed.

Finally, the decompressed RGB image in clear is obtained by performing the inverse quantization operation, the I-DCT transformation, to go from the frequency domain to the spatial domain, and then, the inverse color transformation, to convert the YCrCb image to RGB space. Note that the reconstructed secret JPEG image is exactly the same as the one obtained by applying a standard JPEG compression, which is more or less similar to the secret original image, as a function of the QF value.

C. Reduced share: scenario using a public shared JPEG image

Regarding the proposed method described in Section III-A, one can note that the shared secret information is only about the amplitude values from a JPEG image. In other words, as this processing is selective and done to ensure the JPEG format compliance, the structure of the JPEG information remains in clear (as for example, DC coefficients and header parts). Depending on the number of users n, some blocks also remain in clear because the size of the sequence obtained by concatenating the amplitudes is smaller than $\log_2(n+1)$. Therefore, a scenario using a public shared JPEG image can be derived from the proposed method. The idea is to remove the data remaining in clear from each share – and thus redundant between shares – to reduce their binary size.

An overview of this scenario using a public shared JPEG image is provided in Fig. 3. First, a public shared JPEG image SI_{pub} is generated by substituting the amplitude parts of the non-zero quantized AC coefficients of all the blocks (which are used for the sharing process) with a binary capacity large enough to be secured by random numbers (with a specific binary size in order to achieve JPEG format compliance). Moreover,

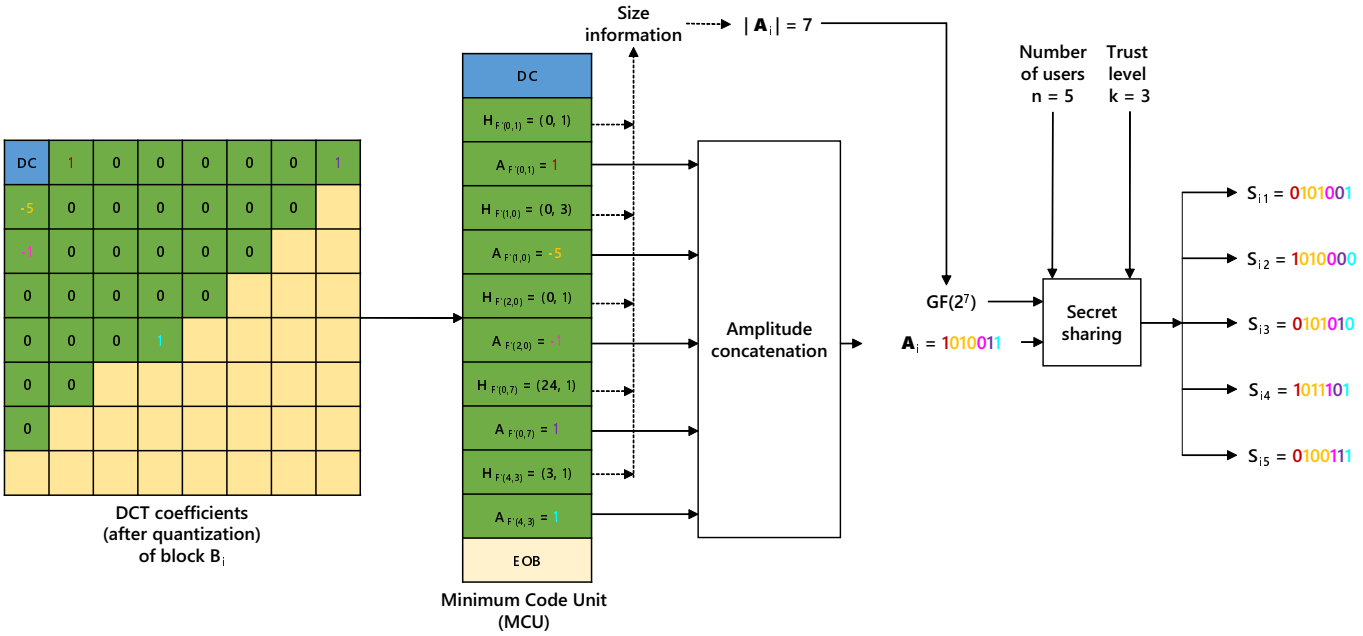


Fig. 4: Our proposed method of joint secret image sharing and JPEG compression applied to one block B_i.

in the secret sharing scheme, the to-be-shared information with our proposed method is much smaller than the size of the secret JPEG image. Then, for each user, a reduced share RS_j is obtained by concatenating the shared values associated to each frequency block of the secret JPEG image. Finally, the amplitude values of the public shared JPEG image SI_{pub} can be substituted (according to the size parameters) in order to visualize the content of RS_j.

D. Full example on a block

In Fig. 4, an example of an application of the proposed method of joint secret image sharing and JPEG compression is illustrated for one block B_i. After the quantization step, the quantized DCT coefficients are stored in zigzag order in the MCU. All the non-zero quantized AC coefficients are considered for secret sharing with parameters $(k, n) = (3, 5)$. In the example illustrated in Fig. 4, there are five non-zero quantized AC coefficients $F'(0,1)$, $F'(1,0)$, $F'(2,0)$, $F'(0,7)$ and $F'(4,3)$, with sizes equal to 1, 3, 1, 1 and 1 respectively. Bits of the codes of their amplitude parts are concatenated to form a binary word $\mathbf{A}_i = 1010011$ of size $|\mathbf{A}_i| = 7$. For example, the code of the $F'(1,0)$ coefficient's amplitude (in orange color and in second position) is 010, encoding the value -5 in the Huffman table. Note that, as $|\mathbf{A}_i| = 7$, the assumption $|\mathbf{A}_i| \geq \log_2(n+1)$ $(7 \geq \log_2(6))$ is ensured. Then, $\mathbf{A}_i$ is shared using the Shamir's approach over the Galois field $GF(2^{|\mathbf{A}_i|}) = GF(2^7)$. During the secret sharing process, 5 shared values S_{ij} associated to each user x_j, with $1 \leq j \leq n$, are computed. These 5 values are not the same as the value of the binary word in clear and are different from each other: they are similar to pseudo-randomly generated values. In order to obtain the 5 shared versions of the block B_i, amplitude values are substituted according to their size parameter. For example, in Fig. 5, one can see the shared version of B_i associated to the user of ID x_1 in its shared JPEG image SI_1. Moreover, note that if the eco-friendly scenario is used, the JPEG structure (in particular, headers of the 5 non-zero quantized AC coefficients) is stored in the public shared JPEG image SI_{pub}, while the shared value S_{ij} is written in the reduced share RS_j associated to each participant of ID x_j.

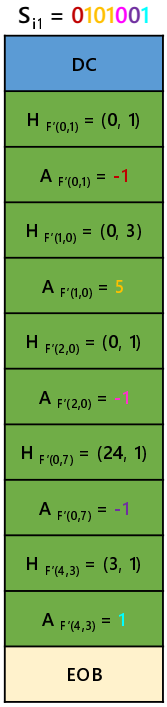


Fig. 5: Shared version of the block B_i associated to the user of ID x_1, according to the example illustrated in Fig. 4.

Fig. 6: Secret original gray level image from the UCID database [38] (512×384 pixels, 197 kB).

IV. Experimental results

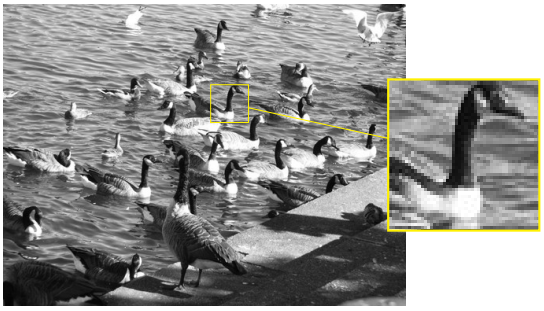
In this section, we present the obtained results by applying our proposed method of joint secret image sharing and JPEG compression. In Section IV-A, we discuss a detailed example for the proposed method applied to a single image. In Section IV-B, we present the results we obtained on the full UCID database of 1,338 images [38]. In Section IV-C, we perform some experiments using the public shared JPEG image scenario. Finally, in Section IV-D, we provide a discussion in which we analyze the security level of our proposed method.

A. A detailed example for the proposed method

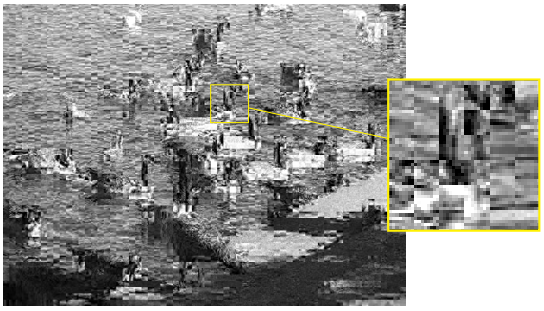
Let us apply our proposed method to the gray level image shown in Fig. 6 from the UCID database [38] (512×384 pixels, 197 kB). First, we apply the first steps of a JPEG compression, up to the Huffman coding stage to our secret original image (with QF $= 90$). For each resulting block, the AC coefficients are encoded using a Huffman code and then concatenated. These concatenations are called binary words $\mathbf{A}_i$, as presented in Section III-A. The obtained JPEG compressed image of size 79 kB is shown in Fig. 7a. The PSNR value between the secret original image and the secret JPEG image is of 38 dB.

We can then apply our proposed (k, n) secret sharing scheme to each binary word $\mathbf{A}_i$, after a Galois field of appropriate size has been generated, $GF(2^{|\mathbf{A}_i|})$. Note that, in a Galois field $GF(2^M)$, any binary word of size M can be generated. For the sake of this example, we use a $(3, 5)$ secret sharing scheme, *i.e.* a scheme with $n = 5$ users where $k = 3$ are needed to reconstruct the secret JPEG image. From each block, we get five evaluations, that is, each user gets an evaluation per block. By substituting the secret original binary words with these evaluations, five shared JPEG images are obtained.

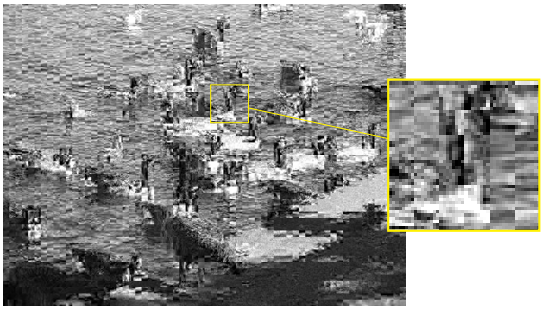
Two of the five different shared JPEG images are shown in Fig. 7b and Fig. 7c. Since the DC coefficients are left unchanged, a low resolution version of the secret original image can still be recognized when looking at an associated shared JPEG image. The PSNR value between a shared JPEG image and the secret original one (Fig. 6), or the compressed image (Fig. 7a), is of approximately 14 dB, which shows that the secret original image content is highly distorted. Consequently, the obtained visual security level is high.



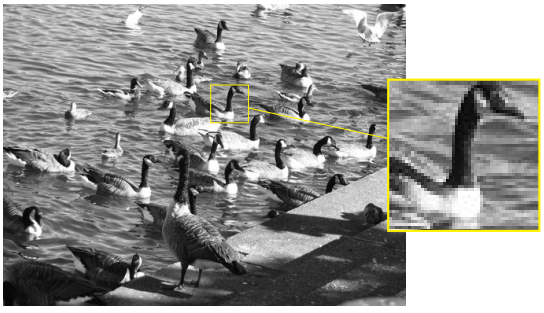
(a)



(b)



(c)



(d)

Fig. 7: Example of our proposed method: a) JPEG image obtained by compression using a quality factor QF = 90 (79 kB), b-c) Two shared JPEG images obtained with our proposed encoding method using parameters $k = 3$ and $n = 5$ (79 kB each), d) The reconstructed secret JPEG image, using 3 shared JPEG images as input of our decoding method (79 kB).

If the threshold of $k = 3$ is reached, the secret original image can be reconstructed. Each block of each shared JPEG image is

an evaluation of a secret polynomial. The zero-order coefficient of this secret polynomial is the clear JPEG compressed image block value. By applying the Lagrange interpolation method on these available evaluations, one can reconstruct the secret JPEG image (provided at least k users get together). Note that the reconstructed secret image (Fig. 7d) is exactly identical to the secret JPEG image illustrated in Fig. 7a, as represented by an infinite PSNR between the reconstructed and the compressed version of the secret original image. If less than $k = 3$ users get together, no information can be reconstructed, as the Lagrange interpolation method is *at most* able to generate a $k-2$ degree polynomial from $k-1$ evaluation points, whereas the secret polynomial encoding the blocks is of degree $k-1$.

B. Experiments on a full database

In this section, we analyse our results when processing the UCID image database ($1,338$ gray level images). Each image is compressed with JPEG at various quality factors (QF $\in \{25, 50, 75, 80, 90\}$).

Fig. 8 depicts the average PSNR at various QF. We note that the PSNR between a shared JPEG image and any of the secret JPEG image or the secret original image is almost constant although the QF varies. It can be deduced from our results that no matter the QF used for the compression, the shared JPEG images we generate give very little information about the content of the secret original image. Indeed, the PSNR between a shared JPEG image and the secret original image is lower than 20 dB.

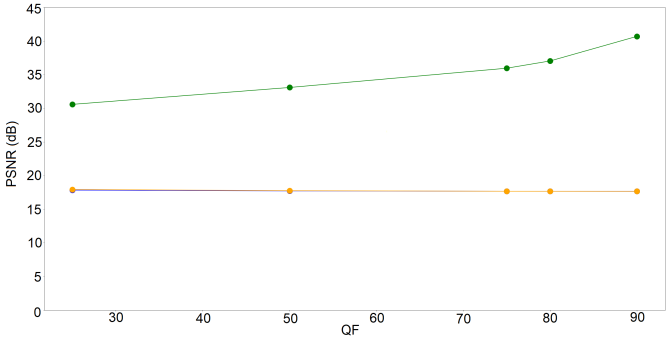


Fig. 8: Average PSNR for 1,338 images of the UCID database as a function of the QF: in green the PSNR between the secret original image and the secret JPEG image, in orange the PSNR between a shared JPEG image and the secret JPEG image, in blue (merged) the PSNR between a shared JPEG image and the secret original image.

In Fig. 9, one can see the percentage of *secured* bits over the share's final size in bits, which we call the *sharing space*, as a function of the QF. When processing an image with our method, we encode each frequency block using a Galois field of size $2^{|\mathbf{A}_i|}$. However, if the size of $\mathbf{A}_i$ is smaller than $\log_2(n+1)$, it is not possible to encode the block concatenated AC coefficients in n *different* binary words. These blocks are then left *in the clear domain* in every shared JPEG image. As presented in Fig. 9, changing the number of users has an effect on the sharing space. As n grows, the number of frequency blocks

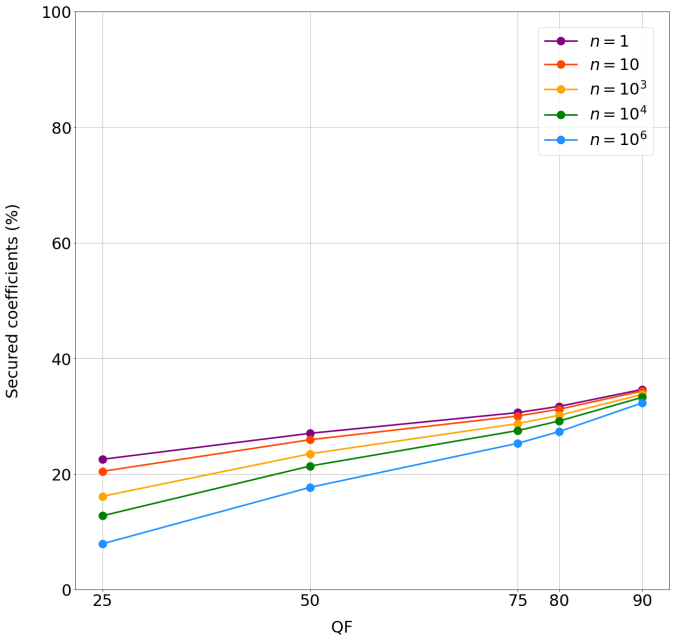


Fig. 9: Sharing space: percentage of *secured* bits over the image size (in bits), as a function of the QF. The different colored curves represent various values of the total number of users, n.

that are too small to be encoded grows as well, meaning that the rate of secured coefficients decreases. The purple curve, with $n = 1$, represents the theoretical maximal encoded blocks ratio for this database, that is, the theoretical value one would obtain if no blocks were left in clear. It can be pointed out that at QF = 90, no matter the number of users, all sharing space values are quite similar. This indicates that our scheme still encodes a *relatively* high percentage of blocks with about a million different users. Even in the case of a celebrity sharing an image on social media, the number of users would rarely go over a million. Although only 10% to 30% of the JPEG file data is secured in our scheme, note that each bit of relevant data is secured.

Fig. 10 shows the differences in the AC coefficients' distribution of a JPEG compressed image when using different QF. It can be noted that as the QF decreases, the number of small size coefficients soars, while the bigger size coefficients become scarcer. At a lower QF, the number of coefficients equal to zero grows, which is why the peak at amplitude 0 goes higher as the QF decreases.

C. Experiments using the public shared JPEG image scenario

If we consider a scenario using a public shared JPEG image, the size of the shares is greatly reduced. According to Fig. 11, since only the secured AC coefficients are stored in reduced shares, this derived scenario allows for an additional compression rate ranging from 2.25 (with QF = 90 and $n = 10$) to 24.43 (with QF = 25 and $n = 10^6$) when compared to the initial scenario presented in this paper. With a reduced share (of which the size is greatly reduced) and the public shared

Coefficients size	1	2	3	4	5	6	7
Entropy value in the secret JPEG image (bpp)	0.9999	1.9250	2.9090	3.8708	4.8708	5.8012	6.5811
Entropy value in the shared JPEG image (bpp)	0.9999	1.9999	2.9994	3.9991	4.9928	5.9719	6.8472
Theoretical maximal entropy value (bpp)	1	2	3	4	5	6	7

TABLE I: Shannon entropy as a function of the coefficient size in the JPEG image.

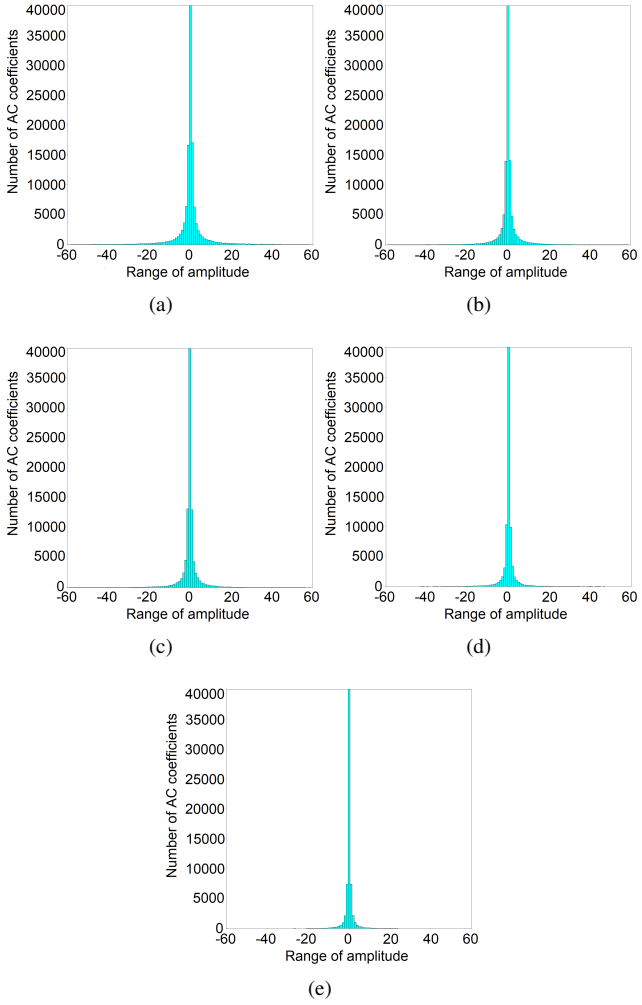


Fig. 10: Trend of the AC coefficients' distribution as the QF decreases, at QF a) 90, b) 80, c) 75, d) 50, and e) 25.

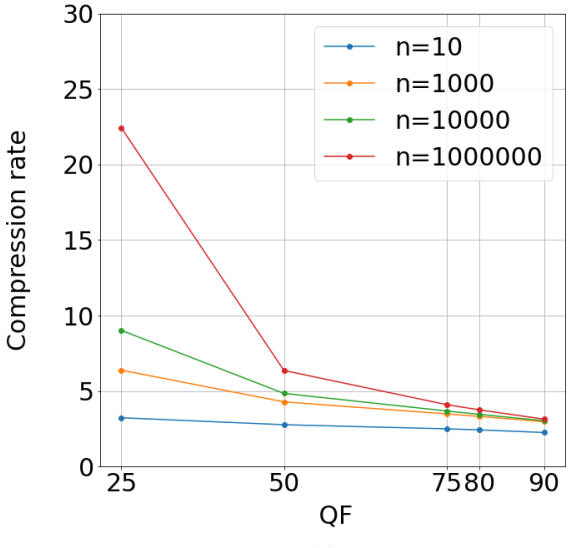


Fig. 11: Additional compression rate obtained with a scenario using a public shared JPEG image and reduced shares, as a function of the QF. The different colored curves represent various values of the total number of users, n.

Fig. 12: Example of a public shared JPEG image obtained from the original image in Fig. 6 by applying our proposed encoding method with parameters $k = 3$ and $n = 5$.

JPEG image, it is possible to substitute the random values to be able to see the visual content of the associated shared JPEG image. An example of a public shared JPEG image obtained from Fig. 6 by applying our proposed encoding method with parameters $k = 3$ and $n = 5$ is provided in Fig. 12. Note that the public shared JPEG image contains all the redundant information of the users' shared JPEG images (that is, DC coefficients, JPEG headers, and AC coefficients from the blocks with a binary capacity too small to be secured).

D. Discussion

In previous work, three visual security levels have been defined in the context of image selective encryption [39]:

1) Transparent level: high resolution is protected, but a degraded version of the image content can be visualized;
2) Sufficient level: the image content is made secure, but some shapes and edges are still viewable;
3) Confidential level: no visual information can be extracted.

Because our proposed method of joint secret sharing and image compression only processes the AC coefficients, one can still recognize some of the content of the secret original image in a shared JPEG image (the DC coefficients are indeed left unchanged). Thus, the visual security level is sufficient. In Fig. 13, we show an image obtained by representing only the DC coefficients values of the shared JPEG images in Fig. 7b, Fig. 7c, and Fig. 12. One can see that the image content remains

secure. Moreover, the obtained PSNR values we obtain when comparing a shared JPEG image and the JPEG compressed image (about 17 dB as a function of the QF which ranges from 25 to 90) are however similar to what one would obtain when comparing a compressed image and its crypto-compressed counterpart, such as the one obtained using the method of Puech *et al.* [40] for example. One can also note that the number of pixel changing rate (NPCR) [41] between the decompressed JPEG image (Fig. 7a) and the decompressed shared JPEG image (Fig. 7b) is about 98%, which indicates that most of the pixel values have been modified during the secret sharing process.

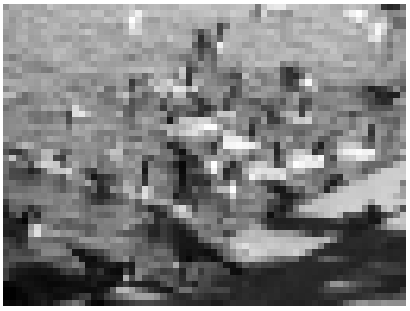
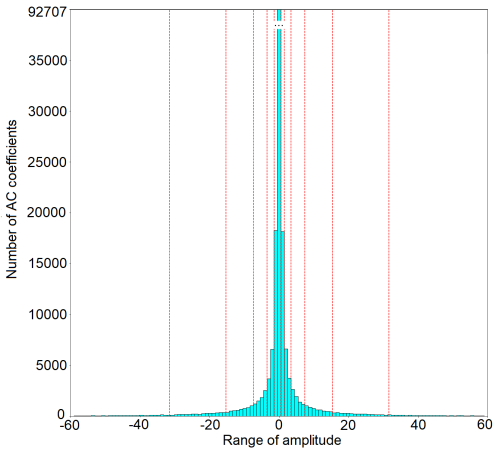


Fig. 13: Image obtained by representing only the DC coefficients values of the shared JPEG images in Fig. 7b and Fig. 7c (64×48 pixels).

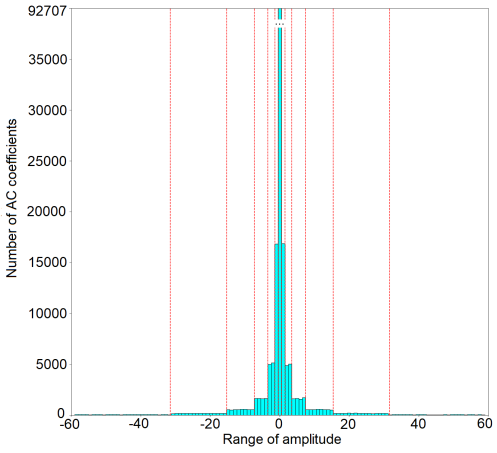
The distribution of the quantized AC coefficients of the JPEG compressed image in Fig. 7a is depicted in Fig. 14a, while the distribution of the AC coefficients in a shared JPEG image is depicted in Fig. 14b. We can observe that both distributions seem to follow a Laplacian distribution. We can also note that in Fig. 14b, all the coefficients with the same size follow a uniform distribution, as shown by the straight steps in the coefficients' distribution. For example, in the original domain, there are approximately 6,500 occurrences of coefficients with an absolute value of 2, and 3,500 occurrences of coefficients of which the absolute value is 3. When a shared JPEG image is created, same size coefficients follow a uniform distribution. Indeed, it can be seen that a plateau is formed for values -3, -2, 2, and 3, at 5,000 occurrences, indicating an equiprobability between same size coefficient values.

In order to ensure this result, the Shannon entropy calculation [42] is performed for each possible coefficient size in the JPEG image. The obtained results are presented in Table I. Regardless of the size, the value of the entropy is very high in the shared JPEG image (second row) and close to the maximal theoretical values (third row), which indicates that the same size quantized AC coefficient distributions tend to be uniform. In comparison, the value measured in the JPEG compressed image in the original domain is smaller (first row).

Moreover, according to Fig. 14, many coefficients are equal to zero after a JPEG compression. One should note that the shared coefficients have exactly the same size (in terms of bits) as the original ones. Therefore, our proposed method allows us to preserve the JPEG structure and the binary size of the standard JPEG compressed image.



(a)



(b)

Fig. 14: a) AC coefficients' distribution in the JPEG compressed image (Fig. 7a), at QF = 90, b) AC coefficients' distribution in a shared JPEG image (Fig. 7b), at QF = 90.

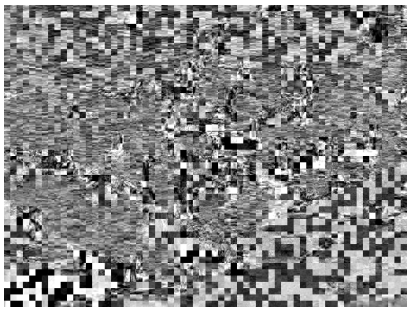


Fig. 15: Shared JPEG image obtained from the original image in Fig. 6 by applying our proposed encoding method on both AC and DC coefficients, with parameters $k = 3$ and $n = 5$.

In order to achieve a confidential level, our scheme can be extended to DC coefficients. To do so, the DC codes of all the MCUs are concatenated in order to obtain a binary sequence. Then, this sequence is split into words of 256 bits. These words are shared using the Shamir secret sharing method over $GF(2^{256})$. An example of a shared JPEG image obtained from the original image in Fig. 6 by applying our proposed

(a) (b) (c)

Fig. 16: Edge maps obtained using a Sobel filter on: a) The original image (Fig. 6), b) An associated shared JPEG image where only the AC coefficients are secured (Fig. 7c), c) An associated shared JPEG image where both the AC and DC coefficients are secured (Fig. 15).

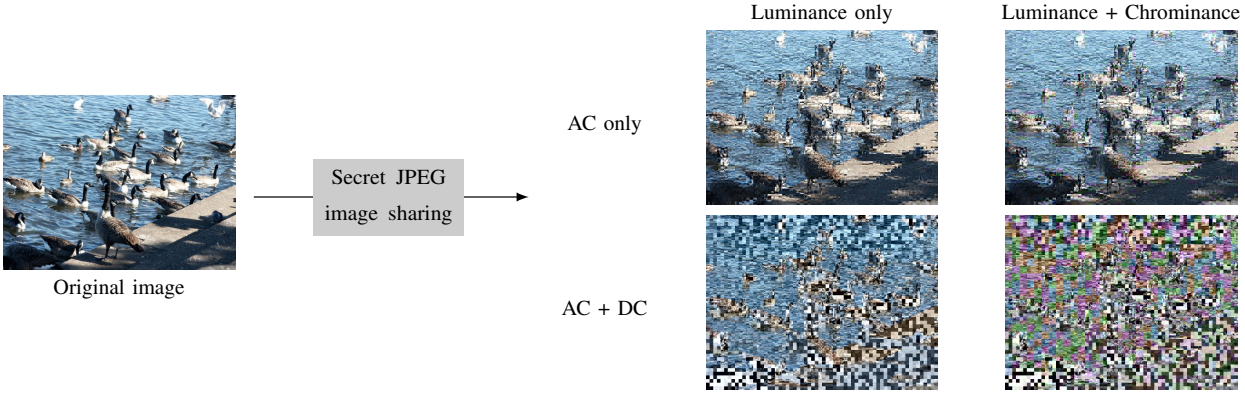


Fig. 17: Secret JPEG color image sharing with $\mathrm{QF} = 90$ and the four possible combinations of our method.

encoding method on both AC and DC coefficients is depicted on Fig. 15. One can see that little to no visual information can be extracted from this shared JPEG image. This is consistent with the computed PSNR value between this image and the original image, which is equal to less than 9 dB (*i.e.* five less decibels than what is obtained by securing the AC coefficients only).

In Fig. 16, we illustrate the edge maps obtained using a Sobel filter on the original image (Fig. 6), an associated shared JPEG image where only the AC coefficients are secured (Fig. 7c) and an associated shared JPEG image where both the AC and DC coefficients are secured (Fig. 15). In order to perform a deeper analysis on the visual security level, we propose to examine the intersection between the edges of the original image and those of an associated shared JPEG image, as introduced in [43]. To do so, we compute a F1-score by considering the following classes:

- True Positives: detected as edge pixels on both the original image and the shared JPEG image edge maps;
- False Positives: detected as edge pixels on the shared JPEG image edge map, but not on the original image edge map;
- False Negatives: detected as edge pixels on the original image edge map, but not on the shared JPEG image edge map.

The F1-score is equal to 0.41 when only the AC coefficients are secured (comparison between Fig. 16a and Fig. 16b) and to 0.31 when both the AC and DC coefficients are secured (comparison between Fig. 16a and Fig. 16c). Therefore, these two low values ensure the visual security of our proposed method, especially when the DC coefficients are also shared.

Note also that quantization tables are encrypted in several crypto-compression approaches. However, doing so prevents one from decoding and visualizing the secured visual content.

Finally, in Fig. 17, we provide an example of the application of our JPEG image secret sharing scheme on the colored version of the image in Fig. 6 using the four different possible combinations with our method: sharing of the AC coefficients only or of both the AC and DC coefficients and on the luminance component only or on both the luminance and chrominance components. One can see that, regarding the visual security level, the same observations as with gray level images can be made. Indeed, by only sharing AC coefficients, a sufficient visual security level is achieved, both with and without securing the chrominance components. Moreover, if the DC coefficients are also shared, a confidential visual security level is obtained. Note that, for particularly sensitive applications, it is preferable to secure the chrominance components in order to avoid visual information leakage.

V. Conclusion

In this paper, we proposed an efficient method of secret image sharing over Galois fields $GF(2^{|\mathbf{A}_i|})$ along with JPEG compression. According to a given quality factor QF, the first steps of JPEG compression are performed on a secret original image. During the JPEG Huffman coding step, for each block MCU, a binary word is obtained by concatenating bits of the non-zero quantized AC coefficients' amplitude parts. These binary words are then shared using the secret sharing scheme of Shamir with parameters $(k,\ n)$ over appropriate Galois fields $GF(2^{|\mathbf{A}_i|})$, according to their size. Furthermore, if the size of a binary word is smaller than $\log_2(n+1)$, the associated block MCU is left in the clear domain in every shared JPEG image. The reconstruction of the JPEG secret image is possible when at least k shared JPEG images among n are gathered. The proposed method is thus format compliant, preserves the compression rate of JPEG and allows us to reconstruct a secret JPEG image, which is exactly the same as the one obtained by applying a standard JPEG compression of the original secret image using QF. In addition, we also described an eco-friendly scenario using a public shared JPEG image and reduced shares. The idea is to remove the data remaining in clear from each share to reduce their binary size. According to our experimental results, the efficiency of our proposed method, whatever the used scenario, is demonstrated, in particular in terms of visual security of the secret JPEG image. Indeed, we achieve to obtain a sufficient visual security level, as with crypto-compression methods of the state-of-art.

In future work, we are interested in studying cheating scenarios in the context of secret JPEG image sharing. In cheating scenarios, dishonest participants (so called cheaters) share fake shared images in order to get an exclusive secret and honest participants to get a fake one. It should be pertinent to analyze the feasibility of this kind of attack using shared JPEG images.

References

[1] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. XVIII–XXXIV, 1992.

[2] Independent JPEG Group, 1991, https://www.ijg.org/.

[3] M. Van Droogenbroeck and R. Benedett, "Techniques for a selective encryption of uncompressed and compressed images," in *International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS)*, 2002, pp. 90–97.

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

[5] W. Li and Y. Yuan, "A leak and its remedy in JPEG image encryption," *International Journal of Computer Mathematics*, vol. 84, no. 9, pp. 1367–1378, 2007.

[6] A. Kerckhoffs, "La cryptographie militaire," *Journal des sciences militaires*, vol. IX, pp. 5–38, 161–191, 1883.

[7] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[8] G. R. Blakley, "Safeguarding cryptographic keys," in *Proceedings of the National Computer Conference*, vol. 48, 1979, pp. 313–317.

[9] C. Thien and J. Lin, "Secret image sharing," *Computers & Graphics*, vol. 26, no. 5, pp. 765–770, 2002.

[10] M. Naor and A. Shamir, "Visual cryptography," in *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer, 1994, pp. 1–12.

[11] X. Jia, D. Wang, D. Nie, and C. Zhang, "Collaborative visual cryptography schemes," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 5, pp. 1056–1070, 2016.

[12] C.-N. Yang, C.-C. Wu, and Y.-C. Lin, "k out of n region-based progressive visual cryptography," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 1, pp. 252–262, 2017.

[13] C.-N. Yang and Y.-Y. Yang, "On the analysis and design of visual cryptography with error correcting capability," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 6, pp. 2465–2479, 2020.

[14] C.-N. Yang, T.-S. Chen, K. H. Yu, and C.-C. Wang, "Improvements of image sharing with steganography and authentication," *Journal of Systems and Software*, vol. 80, no. 7, pp. 1070–1076, 2007.

[15] S. Beugnon, P. Puteaux, and W. Puech, "Privacy protection for social media based on a hierarchical secret image sharing scheme," in *IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 679–683.

[16] M. Belenkiy *et al.*, "Disjunctive multi-level secret sharing," *IACR Cryptology ePrint Arch.*, vol. 2008, p. 18, 2008.

[17] C. Qin, C. Jiang, Q. Mo, H. Yao, and C.-C. Chang, "Reversible data hiding in encrypted image via secret sharing based on $GF(p)$ and $GF(2^8)$," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[18] E. Elsheh and A. B. Hamza, "Secret sharing approaches for 3D object encryption," *Expert Systems with Applications*, vol. 38, no. 11, pp. 13 906–13 911, 2011.

[19] J. Anbarasi and A. Mala, "Verifiable multi secret sharing scheme for 3D models," *International Arab Journal of Information Technology (IAJIT)*, vol. 12, 2015.

[20] S.-S. Lee, Y.-J. Huang, and J.-C. Lin, "Protection of 3D models using cross recovery," *Multimedia Tools and Applications*, vol. 76, no. 1, pp. 243–264, 2017.

[21] S. Beugnon, W. Puech, and J.-P. Pedeboy, "Format-compliant selective secret 3-D object sharing scheme," *IEEE Transactions on Multimedia*, vol. 21, no. 9, pp. 2171–2183, 2019.

[22] C. Shi and B. Bhargava, "A fast MPEG video encryption algorithm," in *ACM International Conference on Multimedia (MM)*. ACM, 1998, pp. 81–88.

[23] A. Said, "Measuring the strength of partial encryption schemes," in *IEEE International Conference on Image Processing (ICIP)*, vol. 2. IEEE, 2005, pp. 1126–1129.

[24] J. Daemen and V. Rijmen, *AES proposal: Rijndael*, 1999.

[25] J. M. Rodrigues, W. Puech, and A. G. Bors, "Selective encryption of human skin in jpeg images," in *IEEE International Conference on Image Processing (ICIP)*. IEEE, 2006, pp. 1981–1984.

[26] K. Kurihara, S. Imaizumi, S. Shiota, and H. Kiya, "An encryption-then-compression system for lossless image compression standards," *IEICE Transactions on Information and Systems*, vol. 100, no. 1, pp. 52–56, 2017.

[27] T. Chuman, K. Kurihara, and H. Kiya, "On the security of block scrambling-based ETC systems against jigsaw puzzle solver attacks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 2157–2161.

[28] J. He, S. Huang, S. Tang, and J. Huang, "JPEG image encryption with improved format compatibility and file size preservation," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2645–2658, 2018.

[29] K. Minemura, Z. Moayed, K. Wong, X. Qi, and K. Tanaka, "JPEG image scrambling without expansion in bitstream size," in *IEEE International Conference on Image Processing (ICIP)*. IEEE, 2012, pp. 261–264.

[30] A. Unterweger and A. Uhl, "Length-preserving bit-stream-based JPEG encryption," in *ACM Workshop on Multimedia and Security (MMSec)*. ACM, 2012, pp. 85–90.

[31] Z. Qian, X. Zhang, and S. Wang, "Reversible data hiding in encrypted JPEG bitstream," *IEEE Transactions on Multimedia*, vol. 16, no. 5, pp. 1486–1491, 2014.

[32] Z. Qian, H. Zhou, X. Zhang, and W. Zhang, "Separable reversible data hiding in encrypted JPEG bitstreams," *IEEE Transactions on Dependable and Secure Computing*, 2016.

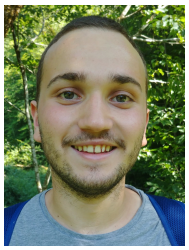
[33] J.-C. Chang, Y.-Z. Lu, and H.-L. Wu, "A separable reversible data hiding scheme for encrypted JPEG bitstreams," *Signal Processing*, vol. 133, no. C, pp. 135–143, 2017.

[34] Z. Qian, H. Xu, X. Luo, and X. Zhang, "New framework of reversible data hiding in encrypted JPEG bitstreams," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 2, pp. 351–362, 2019.

[35] J. He, J. Chen, W. Luo, S. Tang, and J. Huang, "A novel high-capacity reversible data hiding scheme for encrypted JPEG bitstreams," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 12, pp. 3501–3515, 2018.

[36] P. Puteaux, Z. Wang, X. Zhang, and W. Puech, "Hierarchical high capacity data hiding in JPEG crypto-compressed images," in *European Signal Processing Conference (EUSIPCO)*, 2021, pp. 725–729.
[37] V. Itier, P. Puteaux, and W. Puech, "Recompression of JPEG crypto-compressed images without a key," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 3, pp. 646–660, 2019.
[38] G. Schaefer and M. Stich, "UCID: An uncompressed color image database," in *Storage and Retrieval Methods and Applications for Multimedia*, vol. 5307, 2004, pp. 472–480.
[39] D. Engel, T. Stütz, and A. Uhl, "A survey on JPEG2000 encryption," *Multimedia systems*, vol. 15, no. 4, pp. 243–270, 2009.
[40] W. Puech, A. Bors, and J. Rodrigues, "Protection of colour images by selective encryption," in *Advanced Color Image Processing and Analysis*. Springer, 2013, pp. 397–421.
[41] 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 (JSAT)*, vol. 1, no. 2, pp. 31–38, 2011.
[42] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.
[43] Y. Mao and M. Wu, "Security evaluation for communication-friendly encryption of multimedia," in *IEEE International Conference on Image Processing (ICIP)*, vol. 1. IEEE, 2004, pp. 569–572.

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 and her PhD degree in computer science from the Université de Montpellier, France, in 2020. She is currently working as a researcher for the CNRS (French National Centre for Scientific Research) with the Centre de Recherche en Informatique, Signal et Automatique de Lille (CRIStAL), France. Her work has focused on multimedia security, and in particular, image analysis and processing in the encrypted domain. Since 2016, she has published eight journal articles and thirteen conference papers. She is a reviewer for Signal Processing (Elsevier), the Journal of Visual Communication and Image Representation (Elsevier), the IEEE Transactions on Circuits & Systems for Video Technology, and the IEEE Transactions on Dependable and Secure Computing. She has been a member of the IEEE Information Forensics and Security TC since 2023.
Photo credit: © Xavier PIERRE / CNRS



Félix Yriarte is a PhD student at CRIStAL, Lille, on AI and image processing for the detection of anomalies on industrial buildings using UAVs and multispectral imaging. Félix worked on multimedia security during an internship at the LIRMM in 2021, during which he dealt with JPEG security. During his second internship, which took place both at LIRIS, Lyon, and CRIStAL, Lille, France, he worked on document integrity verification.

William Puech received the 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 a 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, cryptography and machine learning. He is head of the ICAR team (Image and Interaction) in the LIRMM and has published more than 45 journal papers and 140 conference papers and is associate editor for 4 journals (SPIC, SP, JVCIR and IEEE TDSC) in the areas of image forensics and security. Since 2017 he has been the general chair of the IEEE Signal Processing French Chapter. He has been a member of the IEEE Information Forensics and Security TC between 2018 and 2020 and then again since 2022. Since 2021 he has also been member of the IEEE Image, Video and Multidimensional Signal Processing TC.