Protection of Color Images by Selective Encryption !!

W. Puech, A. Bors and J.M. Rodrigues

Abstract This chapter presents methods for the protection of privacy associated with specific regions from images or image sequences. In the proposed approaches, regions of interest (ROI) are detected during the JPEG compression of the images and encrypted. We address the problem of simultaneous partial encryption (PE), selective encryption (SE) and image compression. The PE is performed in the ROI selected from the image by specifying a color range. The SE is performed by using the Advanced Encryption Standard (AES) algorithm with the Cipher Feedback (CFB) mode on a subset of the Huffman coefficients corresponding to the AC frequencies chosen according to the level of required security. In this study we consider the encryption of images compressed by JPEG and of image sequences compressed by motion JPEG. Our approach is performed without affecting the compression rate and by keeping the JPEG bitstream compliance. In the proposed method, the SE is performed in the Huffman coding stage of the JPEG algorithm without affecting the size of the compressed image. The most significant characteristic of the proposed method is the utilization of a single procedure to simultaneously perform the compression and the partial encryption rather than using two separate procedures. Our approach reduces then the computational effort and the required computation time. We provide an experimental evaluation of the proposed method when applied on still images as well as on sequences of motion JPEG compressed images acquired with surveillance video-cameras.

W. Puech
Laboratory LIRMM, UMR CNRS 5506, University of Montpellier II, 161, rue Ada, 34392 MONTPELLIER Cedex 05, FRANCE, e-mail: william.puech@lirmm.fr

A. Bors
Dept. of Computer Science, University of York, YORK YO10 5DD, U.K., e-mail: adrian.bors@cs.york.ac.uk

J.M. Rodrigues
Dept. of Computer Science, Federal University of Ceara, BRAZIL, e-mail: marconi@ufc.fr

1 Introduction

Digital rights management (DRM) systems enforce the rights of the multimedia property owners while ensuring the efficient rightful usage of such property. For example the privacy protection in the context of video-camera surveillance is a required feature in many situations. The technical challenges posed by such systems are formidable and previous approaches have not entirely succeeded in tackling them [7].

Multimedia data requires either full encryption or selective encryption depending on the application requirements. For example military and law enforcement applications require full encryption. Nevertheless, there is a large spectrum of applications that demands security on a lower level, as for example that ensured by partial encryption (PE) or selective encryption (SE). Such approaches reduce the computational requirements in networks with diverse client device capabilities [2]. In several papers, the distinction between selective encryption (SE), partial encryption (PE) and soft encryption is not very clear. In this chapter, the goal of PE of an image is to encrypt only regions of interest (ROI) which are defined within specific areas of the image. The goal of SE is to encrypt a well defined range of parameters or coefficients, as for example would be the higher spectrum of frequencies. PE and SE can be used to process and transmit images acquired by a surveillance video-camera. Indeed, in order to visualize these images in real time, they must be quickly transmitted and the full encryption is not really necessary. The security level of PE or SE is always lower when compared with the full encryption. On the other hand PE or SE decreases the data size to be encrypted and consequently requires lower computational time which is an important asset in wireless and portable multimedia systems. In this case we have a trade-off between the amount of encrypted data and the necessary computational resources.

JPEG is a commonly used image compression algorithm which is largely employed in image processing for security communication and in industrial applications [13]. JPEG image compression standard is employed in a large category of systems such as: digital cameras, portable telephones, scanners and various other portable devices. This study shows that PE or SE can be embedded in a standard coding algorithm such as JPEG, JPEG 2000, MJPEG or MPEG, while maintaining the bitstream compliance. In fact, using a standard decoder it should be possible to visualize the PE data in low resolution. On the other hand, with a specific decoding algorithm and a secret key it should be possible to correctly decrypt the PE data and get the high resolution whenever is desired.

In this chapter we present new approaches of PE and SE for JPEG compressed image sequences by using variable length coding (VLC). The proposed method is an improvement of previous methods presented in [14, 15]. We propose to encrypt selected bits in the Huffman coding stage of the JPEG. By using a skin detection procedure, we choose image blocks that definitely contain the faces of people. In our approach we use the Advanced Encryption Standard (AES) [3] in the Cipher Feedback (CFB) mode which is a stream cipher algorithm. This method is then applied to protect the privacy of people passing in front of a surveillance video-camera. Only the authorized persons, possessing the decrypting code are able to see the full video sequences. In Section 2 we provide a short description of JPEG and AES algorithms as well as an overview of previous research results in the area of image encryption. The proposed method is described in Section 3. Section 4 provides a set of experimental results, while Section 5 draws the conclusion of this study.

2 Description of the JPEG compressed images encryption system

Confidentiality is very important for low powered systems such as for example wireless devices. Always, when considering image processing applications on such devices we should use minimal resources. However, the classical ciphers are usually too slow to be used for image and video processing in commercial low powered systems. The selective encryption (SE) can fulfill the application requirements without the overhead of the full encryption. In the case of SE, only the minimum necessary data are ciphered. However, the security of SE is always lower when compared to that of the full encryption. The only reason to accept this drawback is the substantial computational reduction. We review the basic steps of the JPEG algorithm in Section 2.1, the AES algorithm in Section 2.2, while in Section 2.3 we present an overview of the previous work.

2.1 The JPEG algorithm

The standard JPEG algorithm decomposes initially the image in blocks of 8×8 pixels. These pixel blocks are transformed from the spatial to the frequency domain by the Discrete Cosine Transform (DCT). The DC coefficient corresponds to zero frequency and depends on the average component in each 8×8 pixel block, while the AC coefficients correspond to the frequency information. Then, each DCT coefficient is divided by its corresponding parameter from a quantization table, corresponding to the chosen quality factor and rounded afterwards to the nearest integer. The quantized DCT coefficients are mapped according to a predefined zigzag order into an array according to their increasing spatial frequency. Then, this sequence of quantized coefficients is used in the entropy encoding (Huffman coding) stage. The processing stages of the JPEG algorithm are shown in Fig. 1.

In the Huffman coding block, the quantized coefficients are coded by pairs $\{H,A\}$ where H is the head and A the amplitude. The head H contains the control information provided by the Huffman tables. The amplitude A is a signed-integer representing the amplitude of the nonzero AC, or in the case of DC is the difference between the DC coefficients of two neighboring blocks. Because the DC coefficients are highly predictable, they are treated separately in the Huffman coding. For the AC coefficients, H is composed of a pair $\{R,S\}$, where R is the runlength and

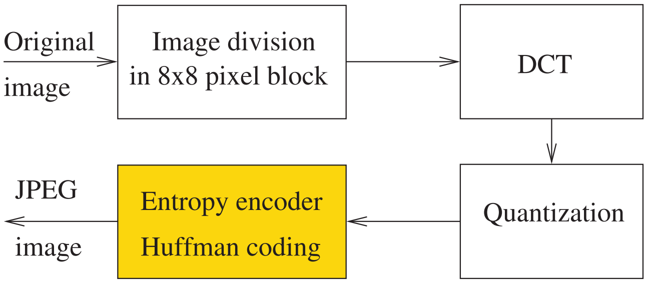


Fig. 1 JPEG algorithm.

S is the size of H, while for the DC coefficients, H is made up only by size S. The SE approach proposed in this chapter is essentially based on encrypting only certain AC coefficients.

For the AC coding, JPEG uses a method based on combining run-length and amplitude information. The runlength R is the number of consecutive zero-valued AC coefficients which precede a nonzero-value from the zigzag sequence. The size S is the amount of necessary bits to represent A. Two extra codes that correspond to $\{R,S\} = \{0,0\}$ and $\{R,S\} = \{15,0\}$ are used to mark the end of block (EOB) and a zero run length (ZRL), respectively. The EOB is transmitted after the last nonzero coefficient in a quantized block. The ZRL symbol is transmitted whenever R is greater than 15 and represents a run of 16 zeros. One of the objectives of our method is to encrypt the image while preserving the JPEG bitstream compliance in order to provide a constant bit rate.

2.2 *The AES encryption algorithm*

The Advanced Encryption Standard (AES) algorithm consists of a set of processing steps repeated for a number of iterations called rounds [3]. The number of rounds depends on the size of the key and the size of the data block. The number of rounds is 9 for example, if both the block and the key are 128 bits long. Given a sequence $\{X_1,X_2,...,X_n\}$ of bit plaintext blocks, each X_i is encrypted with the same secret key k producing the ciphertext blocks $\{Y_1,Y_2,...,Y_n\}$. To encipher a data block X_i in AES you first perform an AddRoundKey step by XORing a subkey with the block. The incoming data and the key are added together in the first AddRoundKey step. Afterwards, it follows the round operation. Each regular round operation involves four steps which are SubBytes, ShiftRows, MixColumns and AddRoundKey. Before producing the final ciphered data Y_i, the AES performs an extra final routine that is composed of (SubBytes, ShiftRows and AddRoundKey) steps.

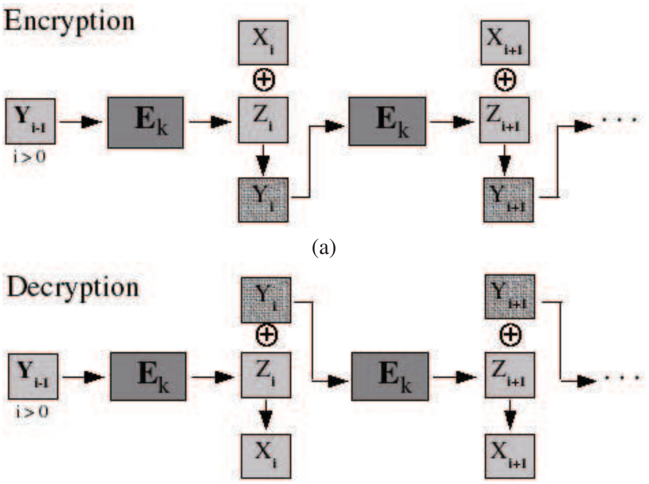
The AES algorithm can support several cipher modes: ECB (Electronic Code Book), CBC (Cipher Block Chaining), OFB (Output Feedback), CFB (Cipher Feedback) and CTR (Counter)[19]. The ECB mode is actually the basic AES algorithm. With the ECB mode, each plaintext block X_i is encrypted with the same secret key k producing the ciphertext block Y_i:

$$Y_i = E_k(X_i). \tag{1}$$

The CBC mode adds a feedback mechanism to a block cipher. Each ciphertext block Y_i is XORed with the incoming plaintext block X_{i+1} before being encrypted with the key k. An initialization vector (IV) is used for the first iteration. In fact, all modes (except the ECB mode) require the use of an IV. In CFB mode, Y_0 is substituted by the IV as shown in Fig. 2. The keystream element Z_i is then generated and the ciphertext block Y_i is produced as:

$$\begin{cases} Z_i = E_k(Y_{i-1}), \text{ for } i \geqslant 1 \\ Y_i = X_i \oplus Z_i \end{cases}, \tag{2}$$

where $\oplus$ is the XOR operator.



(a)

(b)

Fig. 2 CFB stream cipher: a) Encryption, b) Decryption.

In the OFB mode, Z_0 is substituted by the IV and the input data is encrypted by XORing it with the output Z_i. The CTR mode has very similar characteristics to OFB, but in addition it allows pseudo-random access for decryption. It generates the next keystream block by encrypting successive values of a counter.

Although AES is a block cipher, in the OFB, CFB and CTR modes it operates as a stream cipher. These modes do not require any special measures to handle messages whose lengths are not multiples of the block size since they all work by XORing the plaintext with the output of the block cipher. Each mode has its advantages and disadvantages. For example in ECB and OFB modes, any modification in the plaintext block X_i causes the corresponding ciphered block Y_i to be altered, but other ciphered blocks are not affected. On the other hand, if a plaintext block X_i is changed in CBC and CFB modes, then Y_i and all subsequent ciphered blocks will be affected. These properties mean that CBC and CFB modes are useful for the purpose of authentication while ECB and OFB modes treat separately each block. Therefore, we can notice that OFB does not spread noise, while the CFB does exactly that.

2.3 Previous work

Selective encryption (SE) is a technique aiming to save computational time or to enable new system functionalities by only encrypting a portion of a compressed bitstream while still achieving adequate security [9]. SE as well as partial encryption (PE) are applied only on certain parts of the bit stream. In the decoding stage both the encrypted and the non-encrypted information should be appropriately identified and displayed [2, 11, 15]. The protection of the privacy in the context of video-camera surveillance is a requirement in many systems. The technical challenges posed by such systems are high and previous approaches have not entirely succeeded in tackling them [7]. In the context of DRM systems, our study addresses the simultaneous SE, PE and compression for image sequences.

In [20] was proposed a technique called zigzag permutation applicable to DCT-based videos and images. On one hand this method provides a certain level of confidentiality, while on the other hand it increases the overall bit rate. Combining PE and image/video compression using the set partitioning in hierarchical trees was used in [2]. Nevertheless, this approach requires a significant computational complexity. A method that does not require significant processing time and which operates directly on the bit planes of the image was proposed in [10]. SE of video while seeking the compliance with the MPEG-4 video compression standard was studied in [21]. An approach that turns entropy coders into encryption ciphers using statistical models was proposed in [22]. In [4] it was suggested a technique that encrypts a selected number of AC coefficients. The DC coefficients are not ciphered since they carry important visual information and they are highly predictable. In spite of the constancy in the bit rate while preserving the bitstream compliance, this method produces codes which are not scalable. Moreover, the compression and the encryption process are separated and consequently the computational complexity is increased. Fisch *et al.* [5] proposed a method whereby the data are organized in a scalable bitstream form. These bitstreams are constructed with the DC and some AC coefficients of each block which are then arranged in layers according to their visual importance. The PE process is applied over these layers. Some encryption methods have been applied in the DCT coefficient representations of image sequences [2, 21, 26].

The Advanced Encryption Standard (AES) [3] was applied on the Haar discrete wavelet transform compressed images in [12]. The encryption of color images in the wavelet transform has been addressed in [11]. In this approach the encryption takes place on the resulting wavelet code bits. In [14] SE was performed on color JPEG images by selectively encrypting only the luminance component Y. The encryption of JPEG 2000 codestreams has been reported in [6, 8]. PE using a mapping function has been performed in [8]. It should be noticed that wavelet based compression employed by JPEG 2000 image coding algorithm increases the computational demands and is not used by portable devices.

The robustness of partially encrypted images to attacks which exploit the information from non-encrypted bits together with the availability of side information was studied in [16]. The protection rights of individuals and the privacy of certain moving objects in the context of security surveillance systems using viewer generated masking and the AES encryption standard has been addressed in [23]. In the following we describe our proposed approach to apply simultaneously PE, SE and JPEG compression in images.

3 The proposed partial and selective encryption method

The SE is done during the JPEG compression of the color image. Our approach consists of three steps: JPEG compression, partial encryption due to the ROI detection and selective encryption during the Huffman coding stage of JPEG. In Section 3.1 we present an overview of the proposed method. The color range based ROI detection used for PE is described in Section 3.2 and the SE is presented in Section 3.3. In Section 3.4 we explain the decryption of the protected image.

3.1 Overview of the method

In the case of image sequences each frame is treated individually. For each color frame, we apply the color transformation used by the JPEG algorithm, converting from the RGB to the YC_rC_b color space. The two chrominance components (C_r and C_b) are afterwards subsampled. The DCT and the quantization steps of the JPEG algorithm are applied on the three components Y, C_r and C_b. PE and SE are applied only on particular blocks corresponding to the Y component, during the Huffman coding stage because the luminance carries the most significant information [14]. In order to detect the particular blocks that we have to encrypt, we use the quantized DC coefficients of the two chrominance components (C_r and C_b). These quantized DC coefficients are not encrypted and could be used during the decryption stage. Using the quantized DC coefficients we detect the ROI as it will be described in Section 3.2. As part of the SE process, after the ROI detection, selected AC coefficients corresponding to a chosen block are encrypted in the Y component during the Huffman coding stage of JPEG. The detected blocks are selectively encrypted by using the AES algorithm with the CFB mode as it will be described in Section 3.3.

Afterwards, we partially encrypt in the area defined as ROI and we combine with SE by encrypting only the AC coefficients corresponding to the chosen higher range of frequencies. The overview of the method is presented in the scheme from Fig. 3.

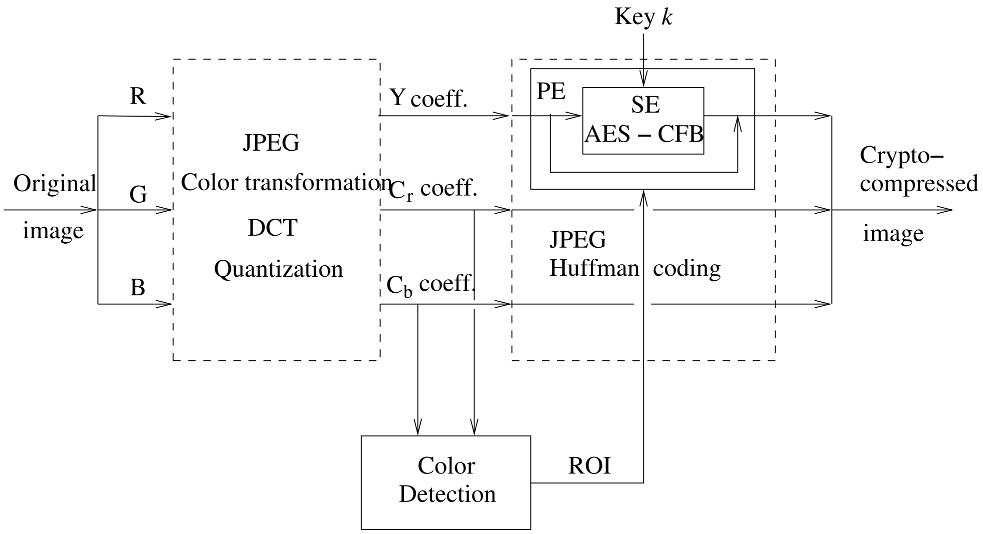


Fig. 3 Global overview of the proposed method.

3.2 *Detection of the ROI using the chrominance components*

The ROI's, representing skin information in our application, are selected using the average color in a 8×8 pixel block, as indicated by the zero frequency (DC coefficients) DCT coefficients. We use the DC coefficients of the C_r and C_b components, denoted as DC_{C_r} and DC_{C_b}, respectively, to detect the human skin according to:

$$\sqrt{\left(\frac{DC_{C_r}}{8} - C_{r_s}\right)^2 + \left(\frac{DC_{C_b}}{8} - C_{b_s}\right)^2} < T \, , \qquad (3)$$

where C_{b_s} and C_{r_s} are the reference skin color in YC_rC_b space and T is a threshold [1, 25]. These parameters are chosen such that the entire range of human skin is detected.

The DC coefficients that fulfill the condition (3) are marked indicating the ROI. Nevertheless, we segment areas that are not always contiguous due to the noise and the uncertainity when choosing a value for the threshold T. Consequently, we have to smooth the chosen image areas in order to ensure contiguity. For enforcing smoothness and contiguity of the ROI we apply morphological opening (erosion followed by dilatation) [18] onto the mapping formed by the marked and non-marked DC coefficients. Smoothed regions of marked DC coefficients indicate the image areas that must be encrypted from the original image. Each marked DC coefficient corresponds to a block of 8×8 pixels. In the following we describe the SE method which is applied to the Huffman vector corresponding to the Y component.

3.3 *Selective encryption of quantified blocks during the Huffman coding stage of JPEG*

Let us consider $Y_i = X_i \oplus E_k(Y_{i-1})$ as the notation for the encryption of a n bit block X_i, using the secret key k with the AES cipher in CFB mode as given by equation (2), and performed as described in the scheme from Fig. 2. We have chosen to use this mode in order to keep the original compression rate. Indeed, with the CFB mode for each block, the size of the encrypted data Y_i can be exactly the same one as the size of the plaintext X_i. Let $D_k(Y_i)$ be the decryption of a ciphered text Y_i using the secret key k. In the CFB mode, the code from the previously encrypted block is used to encrypt the current one as shown in Fig. 2.

The proposed SE is applied in the entropy encoding stage during the creation of the Huffman vector. The three stages of the proposed algorithm are: the construction of the plaintext X_i, described in Section 3.3.1, the encryption of X_i to create Y_i which is provided in Section 3.3.2 and the substitution of the original Huffman vector with the encrypted information, which is explained in Section 3.3.3. These operations are performed separately in each selected quantified DCT block. Consequently, the blocks that contain many details and texture will be strongly encrypted. On the other hand, the homogeneous blocks, *i.e.* blocks that contain series of identical pixels, are less ciphered because they contain a lot of null coefficients which are represented by special codes in the Huffman coding stage. The overview of the proposed SE method is provided in Fig. 4.

3.3.1 The construction of plaintext

For constructing the plaintext X_i, we take the non-zero AC coefficients of the current block i by accessing the Huffman vector in reverse order of its bits in order to create $\{\text{H, A}\}$ pairs. The reason for ordering the Huffman code bits from those corresponding to the highest to those of the lowest frequencies (the reverse order of the zigzag DCT coefficient conversion from matrix to array as used in JPEG) is because the most important visual characteristics of the image are placed in the lower frequencies, while the details are located in the higher frequencies. The human visual system is more sensitive to the lower frequencies when compared to the higher range of frequencies. Therefore, by using the Huffman bits corresponding to the decreasing frequency ordering we can calibrate the visual appearance of the resulting image. This means that we can achieve a progressive or scalable encryp-

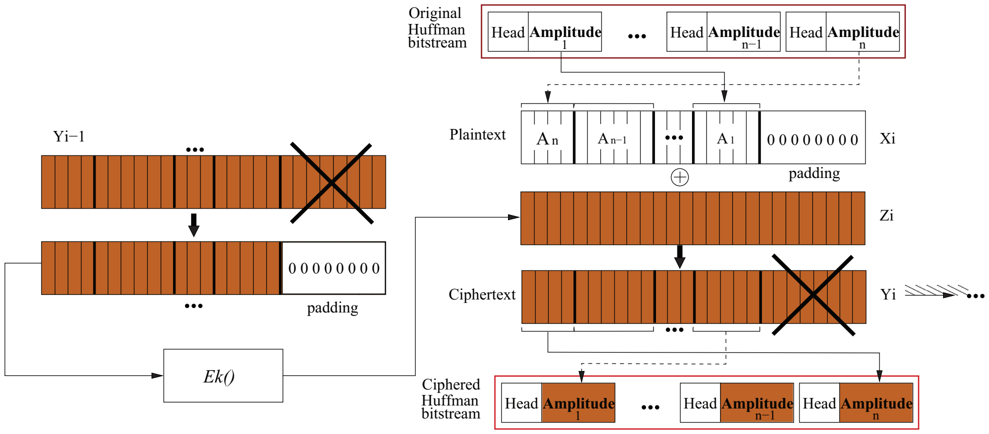


Fig. 4 Global overview of the proposed SE method.

tion with respect to the visual effect. The resulting image will have a higher level of encryption as we increasingly use the lower range of frequencies.

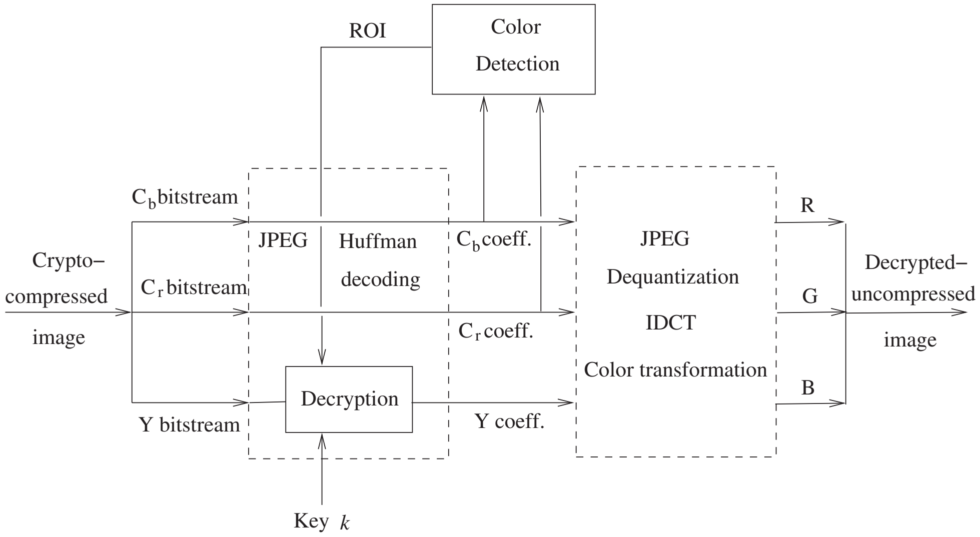


Fig. 5 Global overview of the decryption.

A constraint C is used in order to select the quantity of bits to encrypt from the plaintext X_i. The constraint C graduates the level of ciphering and the visual quality of the resulting image. For each block, the plaintext length $L(X_i)$ to be encrypted depends on both the homogeneity of the block and the given constraint C:

$$0 \leq L(X_i) \leq C, \tag{4}$$

where $C \in \{4, 8, 16, 32, 64, 128\}$ bits. When $C = 128$, AES will fully use the available block of Huffman bits while for the other values several blocks are grouped in order to sum up to 128 bits which is the standard size of AES as explained in Section 2.2. The constraint C specifies the maximum quantity of bits that must be considered for encryption in each block as in VLC. On the other hand, the homogeneity depends on the content of the image and limits the maximum quantity of bits that can be used for encryption from each Huffman block. This means that a block with great homogeneity will produce a small $L(X_i)$. The Huffman vector is encrypted as long as $L(X_i) \leq C$ and the sequence of selected bits does not include those corresponding to the DC coefficient. Then, we apply a padding function $p(j) = 0$, where $j \in \{L(X_i) + 1, \ldots, C\}$, to fill in the vector X_i with zeros up to C bits. In cryptography, padding is the practice of adding values of varying length to the plaintext. This operation is done because the cipher works with units of fixed size, but messages to be encrypted can vary in length. Several padding schemes exist, but we will use the simplest one, which consists of appending null bits to the plaintext in order to bring its length up to the block size. Historically, padding was used to increase the security of the encryption, but in here it is used for rather technical reasons with block ciphers, cryptographic hashing and public key cryptography [17].

The length of amplitude A in bits is extracted using H. These values are computed and tested according to equation (4). In the proposed method, only the values of the amplitudes $(A_n, A_{n-1} ... A_1)$ are considered to build the vector X_i. The Huffman vector is composed of a set of pairs {H, A} and of marker codes such as ZRL and EOB. If the smallest AC coefficients are zero, the Huffman bitstream for this block must contain the mark EOB. In turn, the ZRL control mark is found every time that sixteen successive AC coefficients which are zero are followed by at least one non-zero AC coefficient. In our method, we do not make any change in the head H or in the mentioned control marks. To guarantee the compatibility with any JPEG decoder, the bitstream should only be altered at places where it does not compromise the compliance with the original format.

The homogeneity in the image leads to series of DCT coefficients of value almost zero in the higher range of frequencies. The DCT coefficients can be used to separate the image into spectral sub-bands. After quantization these coefficients become exactly zero [24]. The plaintext construction is illustrated in Fig. 4.

3.3.2 Encryption of the plaintext with AES in the CFB mode

According to equation (2), in the encryption step with AES in the CFB mode, the previous encrypted block Y_{i-1} is used as the input of the AES algorithm in order to create Z_i. Then, the current plaintext X_i is XORed with Z_i in order to generate the encrypted text Y_i.

For the initialization, the IV is created from the secret key k according to the following strategy. The secret key k is used as the seed of the pseudo-random number generator (PRNG). Firstly, the secret key k is divided into 8 bits (byte) sequences. The PRNG produces a random number for each byte component of the key, that defines the order of IV formation. Then, we substitute Y_0 with the IV, and Y_0 is used in AES to produce Z_1.

As illustrated in Fig. 4, with the CFB mode of the AES algorithm, the generation of the keystream Z_i depends of the previous encrypted block Y_{i-1}. Consequently, if two plaintexts are identical $X_i = X_j$ in the CFB mode, then always the two corresponding encrypted blocks are different, $Y_i \neq Y_j$.

3.3.3 Substitution of the original Huffman bitstream

The third step is the substitution of the original information in the Huffman vector by the encrypted text Y_i. As in the first step (construction of the plaintext X_i), the Huffman vector is accessed in the sequential order, while the encrypted vector Y_i is accessed in the reversed order. Given the length in bits of each amplitude $(A_n, A_{n-1}, \ldots, A_1)$, we start substituting the original amplitude in the Huffman vector by the corresponding parts of Y_i as shown in Fig. 4. The total quantity of replaced bits is $L(X_i)$ and consequently we do not necessarily use all the bits of Y_i.

3.4 *Image decryption*

In this section we describe the decryption of the protected image. During the first step we apply the Huffman decoding on the C_r and C_b components. After the Huffman decoding of the two chrominance components we apply the color detection in order to retrieve an identical ROI with the one that had been encrypted. By knowing the ROI it is possible to know which blocks of the Y component should be decrypted during the Huffman decoding stage and which blocks should be only decoded.

The decryption process in the CFB mode works as follows. The previous block Y_{i-1} is used as the input to the AES algorithm in order to generate Z_i. By knowing the secret key k, we apply the same function $E_k(\cdot)$ as that used in the encryption stage. The difference is that the input of the encrypting process is now the ciphered Huffman vector. This ciphered vector is accessed in the reverse order of its bits in order to construct the plaintext Y_{i-1}. Then, it will be used in the AES to generate the keystream Z_i. The keystream Z_i is then XORed with the current block Y_i to generate X_i, as shown in Fig. 2.b. The resulting plaintext vector is split into segments in order to substitute the amplitudes $(A_n, A_{n-1} ... A_1)$ in the ciphered Huffman code and to generate the original Huffman vector. Afterwards, we apply the Huffman decoding and retrieve the quantized DCT coefficients. After the dequantization and the inverse DCT we transform the image from YC_rC_b color space to RGB color space. The overview of the decryption is shown in Fig. 5.

In order to decrypt the image the user needs the secret key. Nevertheless, without the secret key it is still possible to decompress and visualize the image in low resolution format because our approach fulfills the JPEG bitstream compliance and the Huffman bits corresponding to the DC coefficients of the DCT are not encrypted.

4 Experimental results

In this section we analyze the results when applying SE onto the Huffman coding of the high frequency DCT coefficients as well as PE in the region of interest (ROI) from JPEG compressed images and motion JPEG compressed frames from image sequences.

4.1 Analysis of joint selective encryption and JPEG compression

We have applied simultaneously our selective encryption and JPEG compression as described in Section 3, on several images. In this section, we show the results of SE when applied in the entire JPEG compressed image. The original Lena image 512×512 pixels is shown in Fig. 6. The compressed JPEG Lena image with a quality factor (QF) of 100 % is shown in Fig. 7.a and the compressed JPEG with a QF of 10 % is shown in Fig. 7.d.

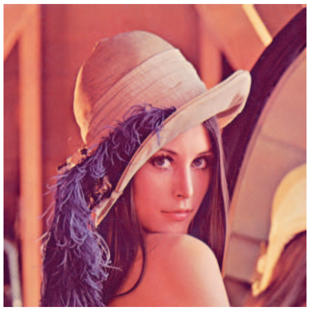
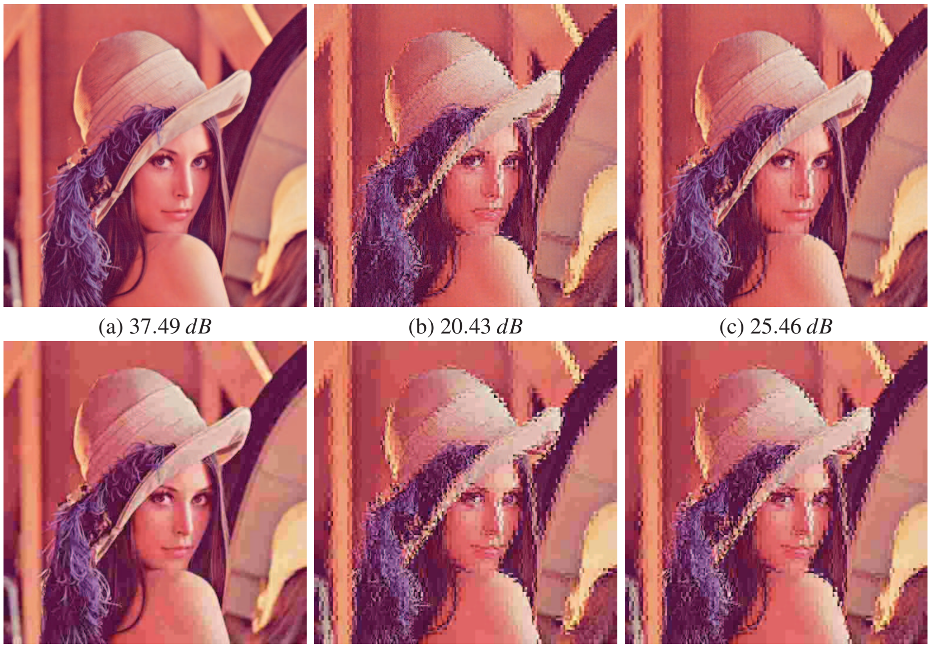


Fig. 6 Original Lena image.

In a first set of experiments, we have analyzed the available space for encryption in JPEG compressed images. In Table 1 we provide the number of bits available for selective encryption for Lena of 512×512 pixels corresponding to various JPEG quality factors. In the same table, for each QF we provide the distortion calculated as the PSNR (Peak to Signal Noise Ratio) as well as the average number of available bits for SE per block of quantized DCT coefficients. We can observe that when


(a) 37.49 *dB* (b) 20.43 *dB* (c) 25.46 *dB*

(d) 27.53 *dB* (e) 20.31 *dB* (f) 20.60 *dB*

Fig. 7 a) JPEG compressed image with QF=100%, b) Image (a) with $C = 128$ bits/block, c) Image (a) with $C = 8$ bits/block, d) JPEG compressed image with QF=10%, e) Image (d) with $C = 128$ bits/block, f) Image (d) with $S = 8$ bits/block.

QF is lower and implicitly the image compression is higher, we are able to embed fewer bits in the compressed image. This is due to the fact that JPEG compression creates flat regions in the image blocks resulting in the increase of the number of AC coefficients equal to zero. Consequently, the Huffman coding creates special blocks for such regions which our method does not encrypt. Not all the available bits provided in the third column of Table 1 are actually used for SE because of the limit imposed by the constraint C. For optimizing the time complexity, C should be smaller than the ratio between the average number of bits and the block size.

In Fig. 8 we provide the graphical representation of the last column from Table 1, displaying the variance of the ratio between the number of available bits for SE and the total number of block bits. We can observe that this variance decreases together with the QF as the number of flat regions in the compressed image increases. For improving the time requirements of the proposed encryption method a smaller constraint C should be used.

In Fig. 9 we show the evaluation of the PSNR between the encrypto-compressed Lena image and the original, for several QF and for various constraints C. In the same figure, for comparison purposes we provide the PSNR between the compressed image with different QF and the original image. From this figure we can

		Bits available for SE		
Quality Factor	PSNR (dB)	Total in the Y component	Percentage from Y component	Average Bits/block
100	37.49	537936	25.65	131
90	34.77	153806	7.33	38
80	33.61	90708	4.33	22
70	32.91	65916	3.14	16
60	32.41	50818	2.42	12
50	32.02	42521	2.03	10
40	31.54	34397	1.64	8
30	30.91	26570	1.27	6
20	29.83	17889	0.85	4
10	27.53	8459	0.40	2

Table 1 Results for various JPEG quality factors.

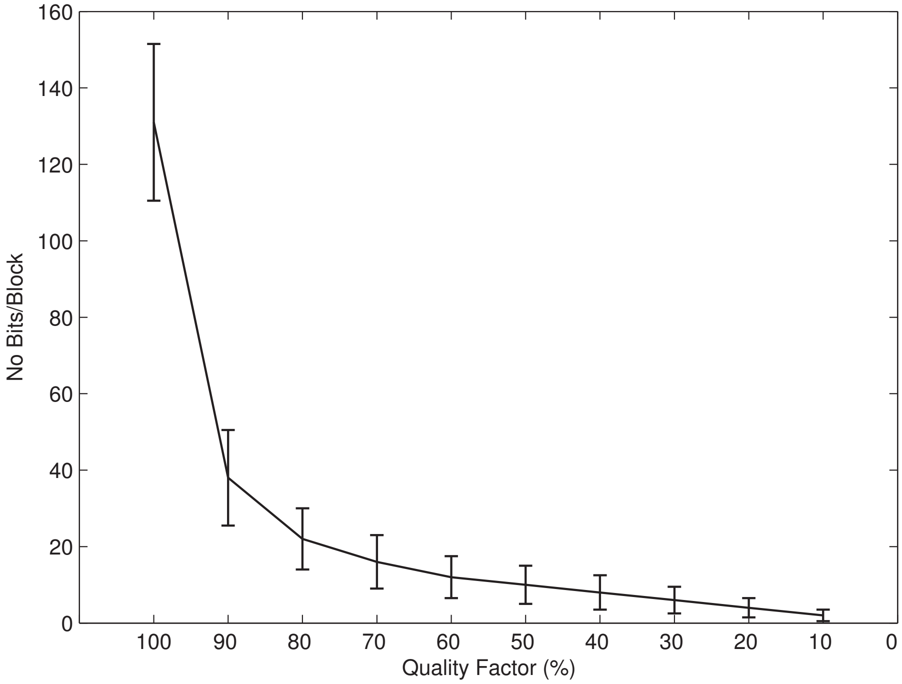


Fig. 8 The ratio between the average number of bits available for SE and the block size. The variance is indicated as a confidence interval.

observe that for a higher C we encrypt a larger number of bits and consequently the image is more distorted with respect to the original. It can be observed that when $C \in \{32, 64, 128\}$, the difference in the PSNR distortion is similar and varies slowly when decreasing the QF.

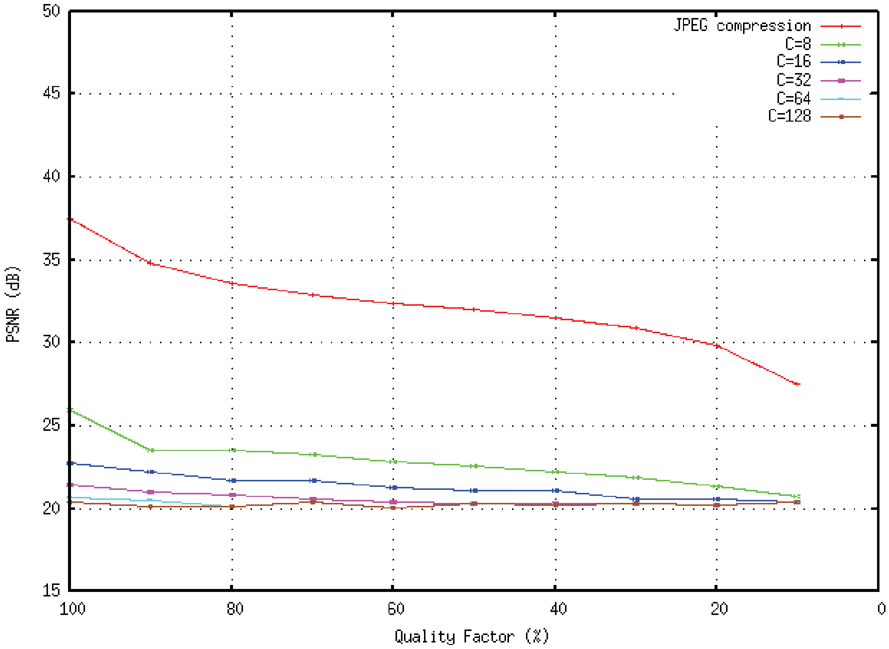


Fig. 9 PSNR of encrypto-compressed Lena image for various quality factors and constraints.

In Fig. 7.b we show the original Lena image encrypted using a constraint $C = 128$ bits per block of quantized DCT coefficients, while in Fig. 7.c the same image is encrypted using a constraint of $C = 8$ bits/block. In Fig. 7.e we show Lena image with QF of 10 %, encrypted using a constraint $C = 128$ bits/block, while in Fig. 7.f the same image is encrypted using a constraint $C = 8$ bits/block. We can see that the degradation introduced by the encryption in the image with QF=100 %, from Fig. 7.b, is higher than the degradation in the image from Fig. 7.c because in the latter we encrypt more bits per block. When combining a high JPEG compression level (QF=10 %) with selective encryption, as shown in the images from Figs. 7.e and 7.f, we can observe a high visual degradation with respect to the images from Figs. 7.b and 7.c, respectively. The higher distortion is caused by the increase in the number of block artifacts. The distortion is more evident when observing some image features as for example the eyes.

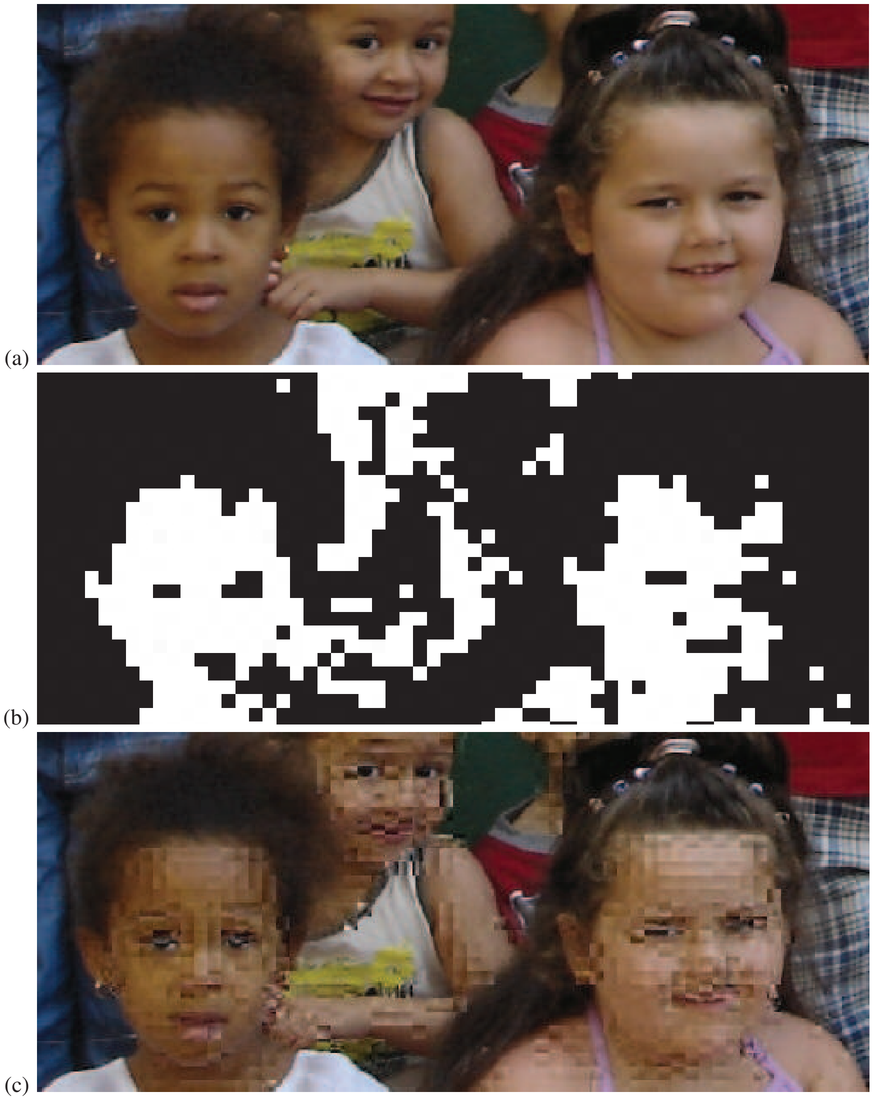


Fig. 10 Selective and partial encryption of skin: a) Original image 416 × 200 pixels, b) ROI detection, c) Protected image.

4.2 Selective encryption of the region of interest in color images

In this section we have applied our encryption method to the color image illustrated in Fig. 10.a[1] and on a color image sequence shown in Fig. 11.a. We use the DC

[1] In order to display the image artifacts produced by our encrypto-compression algorithm we have cropped a sub-image of 416 × 200 pixels

components of the chrominance in order to select the region of interest (ROI) which in this case corresponds to the skin. Based on several experimental tests, for the initial color image in the RGB space displayed in Fig. 10.a, we consider the following values in equation (3): $T = 15$, $C_{r_s} = 140$ and $C_{b_s} = 100$. The resulting ROI's are shown in Fig. 10.b. We can observe that all the skin regions, including the faces are correctly detected. Each selected DC coefficient corresponds to a pixel block marked with white in Fig. 10.b. Only these blocks are selectively encrypted. We can observe that a diversity of skin colors has been appropriately detected by our skin selection approach defined by equation (3). We have then PE and SE encrypted the original image from Fig. 10.a by using the proposed skin detection procedure. We encrypt 3597 blocks from a total of 11136 blocks in the full image of 1024 × 696 pixels, resulting in the encryption of only 7.32 % from the image. The resulting SE and PE image is shown in Fig. 10.c.

For our experiments on the color image sequence illustrated in Fig. 11.a, we have extracted four images (#083, #123, #135, #147) from a sequence of 186 images acquired with a surveillance video-camera. Each one of them is in JPEG format with a QF of 100%. For the encryption we have used the AES cipher in the CFB stream cipher mode with a key of 128 bits long.

Each RGB original image, 640 × 480 pixels, of the extracted sequence, shown in Fig. 11.a was converted to YC_bC_r. An example of the image components Y, C_b and C_r for the frame #83 is shown in Fig. 12. For the skin selection we have used the DC of chrominance components C_b and C_r. The binary images were filtered using a morphological opening operation (erosion followed by dilatation) [18] to obtain the neat binary images illustrated in Fig. 11.b. The detection of the human skin region, in this case mostly of human faces, is represented by the white pixels. We have mapped a white pixel in the binary image as corresponding to a block of 8 × 8 pixels from the original image. Finally, we have applied the method described in this study to generate the selectively encrypted images.

	Total ciphered			
Image	Quant. Blocks	Coeff.	Bits	Blocks %
083	79	2547	10112	1.65
123	113	3042	14464	2.35
135	159	4478	20352	3.31
147	196	5396	25088	4.08

Table 2 Results of PE and SE in a sequence of images acquired with a surveillance video-camera

Table 2 shows the cryptography characteristics for each image. For the frame #083 we have detected 79 blocks representing people faces. This means that 2547 AC coefficients are encrypted, corresponding to 10112 bits in the Huffman code. The number of encrypted blocks corresponds to 1.6 % of the total number of blocks from the original image. For the frame #123 we have 113 blocks. In this frame we have encrypted 3042 AC coefficients which represent 14464 bits corresponding

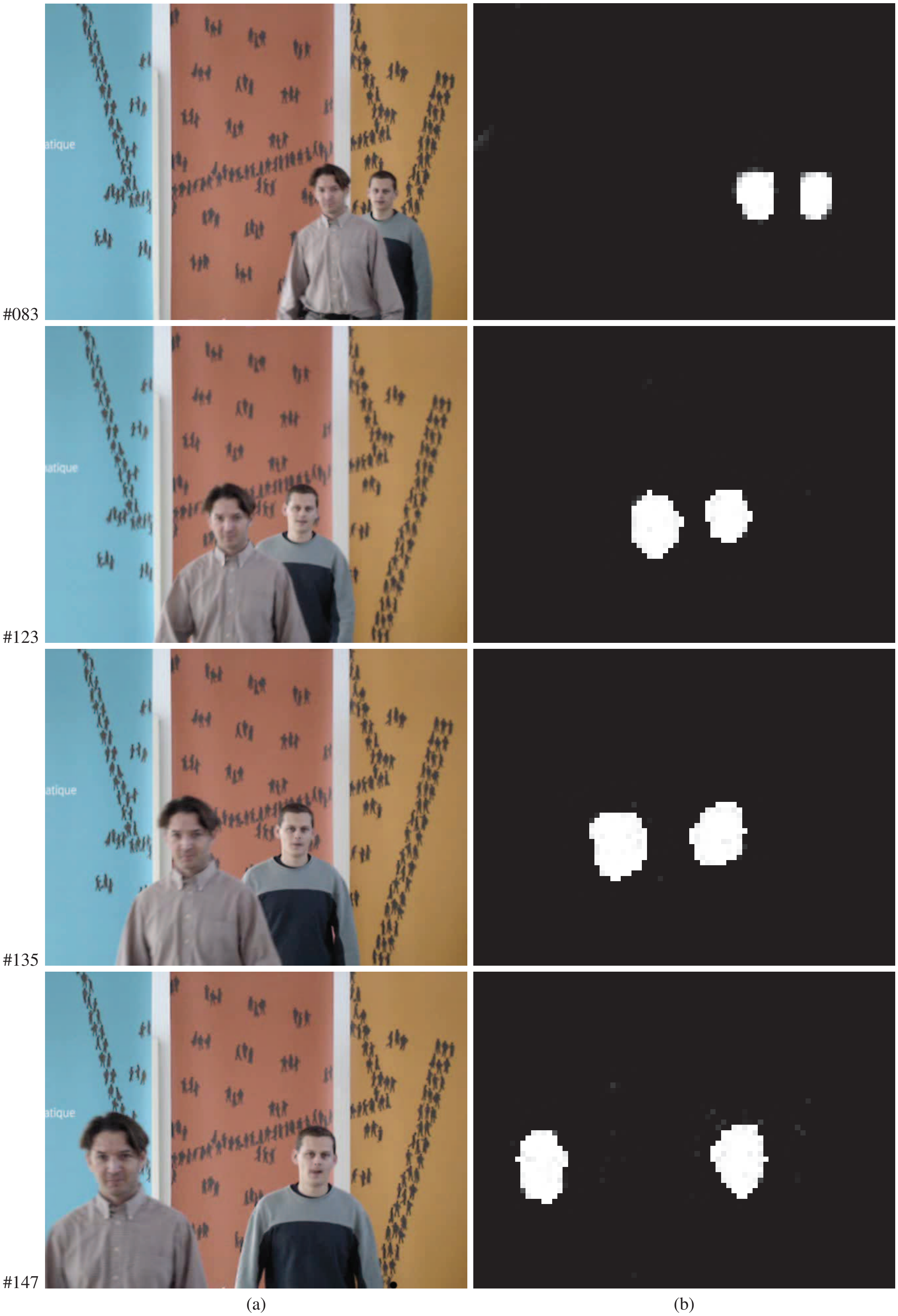


Fig. 11 a) Sequence of original images, b) Detection of the ROI representing the skin.

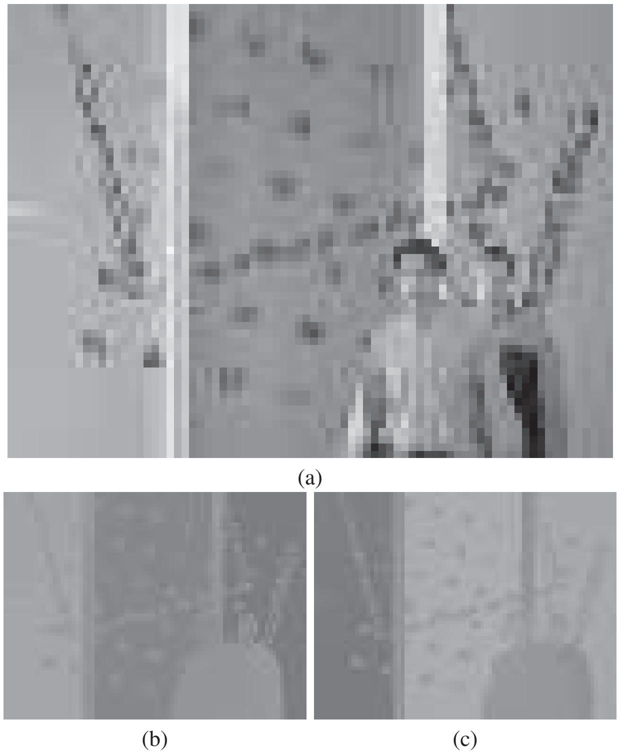
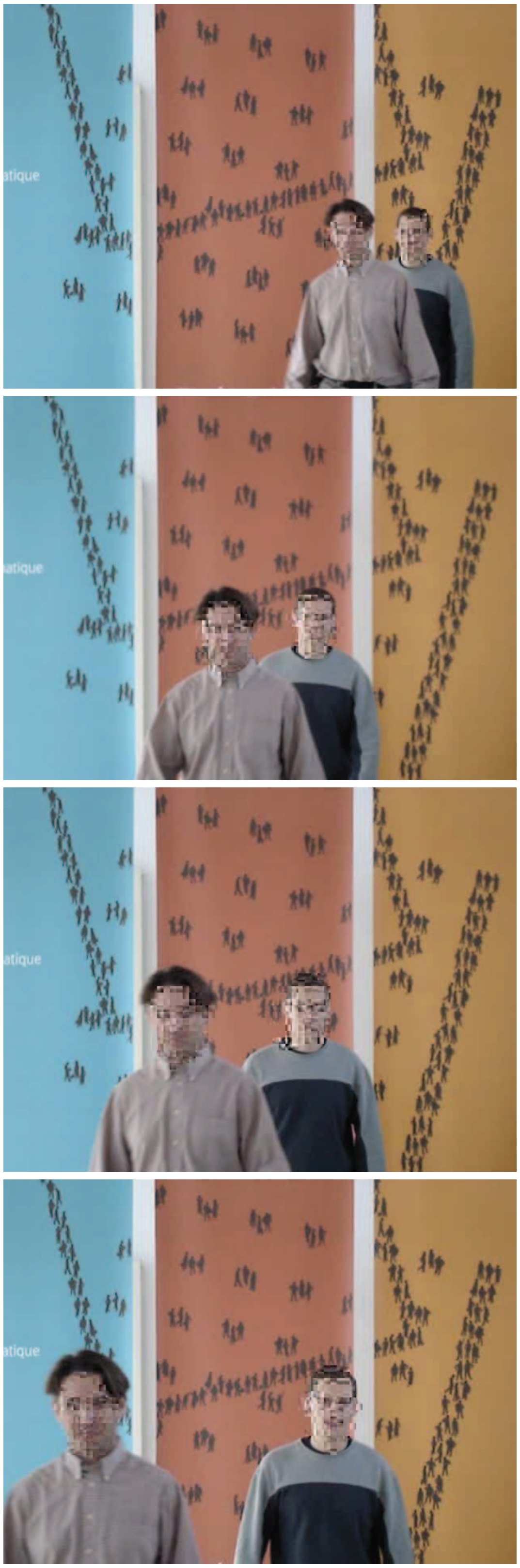


Fig. 12 DC coefficients of frame #083 for the three components YC_bC_r: a) Y component, b) C_r component, c) C_b component.

to 2.35 % from the total number of blocks in the image. The quantity of blocks for encryption increases because the two persons are getting closer to the video-camera. After analyzing Table 2, we can conclude that the amount of bits encrypted is very small relatively to the size of the whole image. This makes our method suitable for low powered systems such as surveillance video-camera. Fig. 13 shows the final results of face detection, tracking and the selective encryption of the chosen frames. In order to clearly show our results, we have cropped from frame #123 a detail of 216 × 152 pixels which is shown enlarged in Fig. 14.

4.3 Cryptanalysis and computation time of the SE method

It should be noted that security is linked to the ability to guess the values of the encrypted data. For example, from a security point of view, it is preferable to encrypt the bits that look the most random. However, in practice this trade-off is challeng-



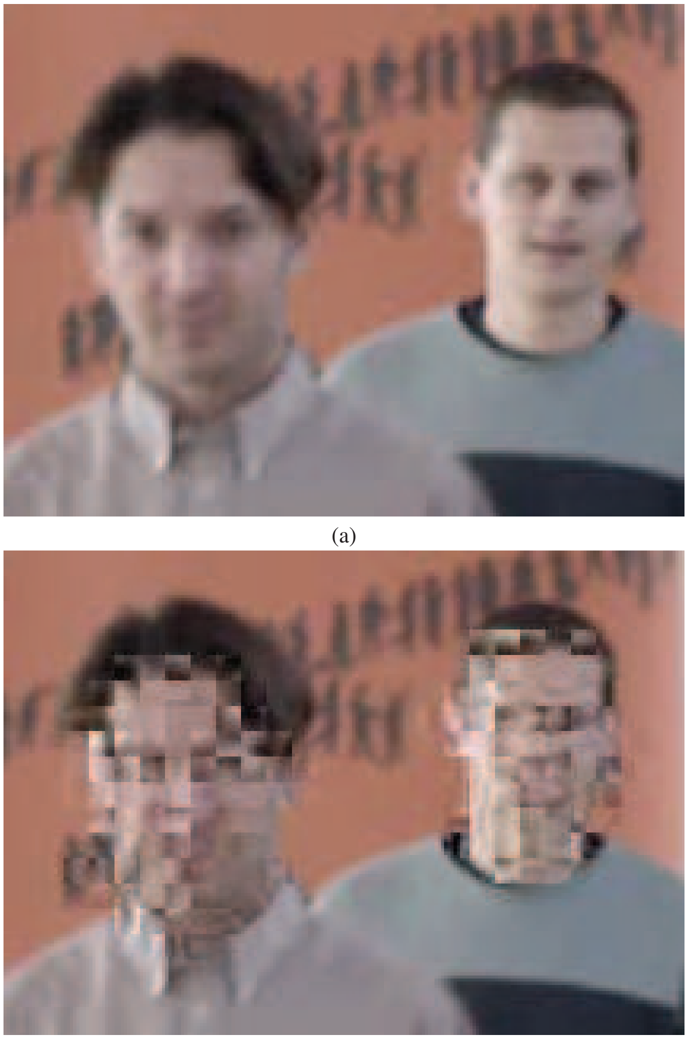
#083

#123

#135

#147

Fig. 13 Sequence of selectively encrypted images.



(a)

(b)

Fig. 14 Region of 216 × 152 pixels from frame #123: a) Original image, b) Protected image.

ing because the most relevant information, such as the DC coefficients in a JPEG encoded image are usually highly predictable [4].

In another experiment we have replaced the encrypted AC coefficients with constant values. For example, if we set the encrypted AC coefficients of all blocks from Fig. 7.b which shows Lena with QF = 100 %, $C = 128$ having $PSNR = 20.43\ dB$, to zero, we get the image illustrated in Fig. 15. Its PSNR with respect to the original image is 23.44 dB. We can observe that in SE, since we do not encode the Huffman

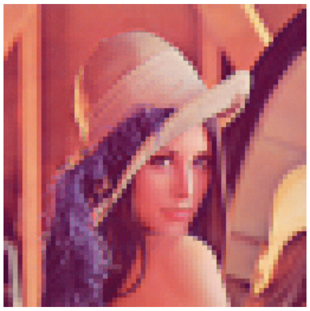


Fig. 15 Attack in the selectively encrypted image (Fig.7.b) by removing the encrypted data (23.44 *dB*).

coefficients corresponding to the DC component, the rough visual information can be simply recovered by replacing the ciphered AC coefficients with constant values. This action will result in an accurate but low resolution image.

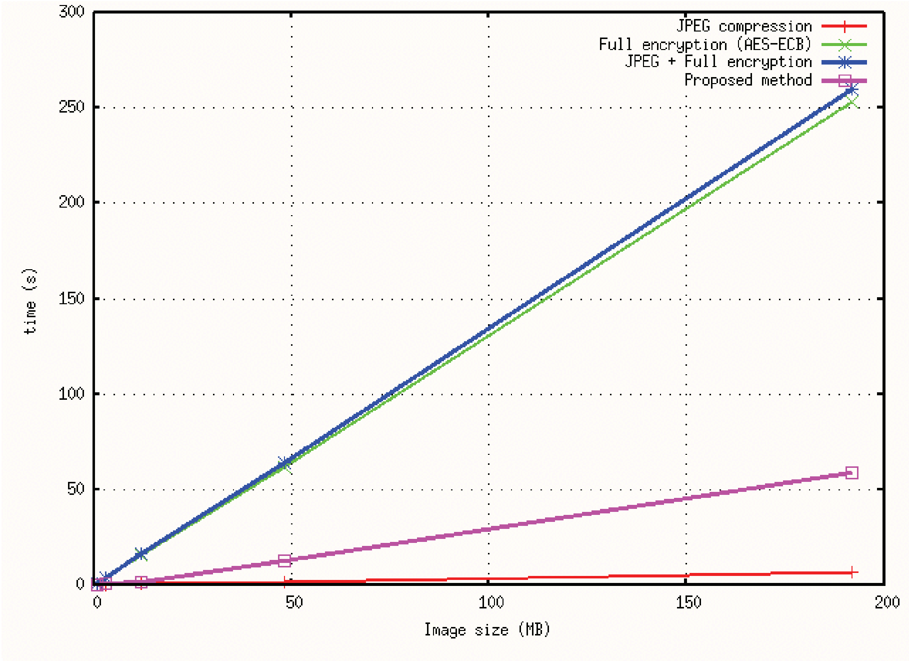


Fig. 16 Computation time (s) as a function of the image size (MB): blue curve for the use of separate JPEG compression and full encryption with AES and pink curve for our proposed method.

We present in Fig. 16 the comparison of the computation time as a function of the image size (MB) between the use of two separate algorithms (JPEG compression and full encryption with AES in ECB mode) and our proposed method[2]. Because of the SE, we concede that our method is slower as compared to a single standard processing such as JPEG (red curve). But it must be noted that to compress and partially encrypt the image, our method is faster as a whole than two standard methods separately applied. From Fig. 16 we can remark that computation time is divided by at least four if you compare with a standard approach based on two separate algorithms. For example for a sequence of 100 images of 768 kB each (512×512 pixels) we need only 19.0 s instead of 103.5 s (3.5 s for JPEG and 100.0s for the full encryption) to process the images. In order to transmit this image sequence, with our proposed method we can process in real time 5.26 images/s instead of 0.97 image/s. Thus, for this kind of applications and image sizes, the proposed method is very interesting because the time processing is five time smaller and can reach 5 image/s which is enough in a context of surveillance video-camera systems.

5 Conclusion

In this chapter, a new partial and selective encryption system has been proposed for JPEG compressed images. The encryption is performed in the Huffman coding stage of the JPEG algorithm using the AES encryption algorithm in the CFB mode. In this way the proposed encryption method does not affect the compression rate and the JPEG bitstream compliance. The selective encryption (SE) is performed only on the Huffman vector bits that correspond to the AC coefficients as provided by the DCT block of JPEG. The SE is progressively performed according to a constraint value onto the Huffman vector bits ordered in the reversing order of their corresponding frequencies. This procedure determines the desired level of selectivity for the encryption of the image content. The DC coefficient provided by the DCT is used as a marker for selecting the region of interest (ROI) for partial encryption. Due to the fact that the Huffman code corresponding to the DC component is not encrypted, a low resolution version of the image can be visualized without the knowledge of the secret key. In the decoding stage, we can use the DC coefficient value in order to identify the encrypted regions. The proposed methodology is applied for ensuring the personal privacy in the context of surveillance video-camera systems. The color range of skin is used to detect faces of people as ROI in video streams and afterwards to PE and SE them. Only authorized users that possess the key can decrypt the entire encrypted image sequence. The proposed method has the advantage of being suitable for mobile devices, which currently use the JPEG image compression algorithm, due to its lower computational requirements. The experiments have shown that we can achieve the desired level of encryption in selected areas of the image, while maintaining the full JPEG image compression compliance, under a minimal

[2] We have used a 2.0 GHz Intel Pentium PC.

set of computational requirements. Motion estimation and tracking can be used to increase the robustness and to speed up the detection of ROI. The proposed system can be extended to standard video coding systems such as those using MPEG.

References

1. D. Chai and K.N. Ngan. Face Segmentation Using Skin-Color Map in Videophone Applications. *IEEE Trans. on Circuits and Systems for Video Technology*, 9(4):551–564, Apr. 1999.
2. H. Cheng and X. Li. Partial Encryption of Compressed Images and Videos. *IEEE Trans. on Signal Processing*, 48(8):2439–2445, Aug. 2000.
3. J. Daemen and V. Rijmen. AES Proposal: The Rijndael Block Cipher. Technical report, Proton World Int.l, Katholieke Universiteit Leuven, ESAT-COSIC, Belgium, 2002.
4. M. Van Droogenbroeck and R. Benedett. Techniques for a Selective Encryption of Uncompressed and Compressed Images. In *Proc. of Advanced Concepts for Intelligent Vision Systems (ACIVS) 2002, Ghent, Belgium*, pages 90–97, Sept. 2002.
5. M. M. Fisch, H. Stgner, and A. Uhl. Layered Encryption Techniques for DCT-Coded Visual Data. In *Proc. European Signal Processing Conference (EUSIPCO) 2004, Vienna, Austria*, pages 821–824, Sep. 2004.
6. S. Imaizumi, O. Watanabe, M. Fujiyoshi, and H. Kiya. Generalized Hierarchical Encryption of JPEG2000 Codestreams for Access Control. In *Proc. IEEE Int. Conf. on Image Processing, Atlanta, USA*, pages 1094–1097, Oct. 2006.
7. E.T. Lin, A.M. Eskicioglu, R.L. Lagendijk, and E.J. Delp. Advances in Digital Video Content Protection. *Proc. of the IEEE*, 93(1):171–183, 2005.
8. J.L. Liu. Efficient Selective Encryption for JPEG 2000 Images Using Private Initial Table. *Pattern Recognition*, 39(8):1509–1517, Aug. 2006.
9. T. Lookabaugh and D.C. Sicker. Selective Encryption for Consumer Applications. *IEEE Communications Magazine*, 42(5):124–129, May 2004.
10. R. Lukac and K.N. Plataniotis. Bit-Level Based Secret Sharing for Image Encryption. *Pattern Recognition*, 38(5):767–772, May 2005.
11. K. Martin, R. Lukac, and K.N. Plataniotis. Efficient Encryption of Wavelet-Based Coded Color Images. *Pattern Recognition*, 38(7):1111–1115, Jul. 2005.
12. S.C. Ou, H.Y. Chung, and W.T. Sung. Improving the Compression and Encryption of Images Using FPGA-Based Cryptosystems. *Multimedia Tools and Applications*, 28(1):5–22, Jan 2006.
13. W.B. Pennebaker and J.L. Mitchell. *JPEG: Still Image Data Compression Standard*. Van Nostrand Reinhold, San Jose, USA, 1993.
14. J.-M. Rodrigues, W. Puech, and A.G. Bors. A Selective Encryption for Heterogenous Color JPEG Images Based on VLC and AES Stream Cipher. In *Proc. European Conference on Colour in Graphics, Imaging and Vision (CGIV'06), Leeds, UK*, pages 34–39, Jun. 2006.
15. J.-M. Rodrigues, W. Puech, and A.G. Bors. Selective Encryption of Human Skin in JPEG Images. In *Proc. IEEE Int. Conf. on Image Processing, Atlanta, USA*, pages 1981–1984, Oct. 2006.
16. A. Said. Measuring the Strength of Partial Encryption Scheme. In *Proc. IEEE Int. Conf. on Image Processing, Genova, Italy*, volume 2, pages 1126–1129, 2005.
17. B. Schneier. *Applied cryptography*. Wiley, New-York, USA, 1995.
18. J. Serra. *Image Analysis and Mathematical Morphology*. London: Academic Press, 1988.
19. D. R. Stinson. *Cryptography: Theory and Practice, (Discrete Mathematics and Its Applications)*. Chapman & Hall/CRC Press, New York, November 2005.
20. L. Tang. Methods for Encrypting and Decrypting MPEG Video Data Efficiently. In *Proc. ACM Multimedia*, volume 3, pages 219–229, 1996.

21. J.T. Wen, M. Severa, W.J. Zeng, M.H. Luttrell, and W.Y. Jin. A Format-Compliant Configurable Encryption Framework for Access Control of Video. *IEEE Trans. on Circuits and Systems for Video Technology*, 12(6):545–557, Jun. 2002.
22. C.P. Wu and C.C.J. Kuo. Design of Integrated Multimedia Compression and Encryption Systems. *IEEE Trans. on Multimedia*, 7(5):828–839, Oct. 2005.
23. K. Yabuta, H. Kitazawa, and T. Tanaka. A New Concept of Security Camera Monitoring with Privacy Protection by Masking Moving Objects. In *Proc. Advances in Multimedia Information Processing*, volume 1, pages 831–842, 2005.
24. J.H. Yang, H. Choi, and T. Kim. Noise Estimation for Blocking Artifacts Reduction in DCT Coded Images. *IEEE Trans. on Circuits and Systems for Video Technology*, 10(7):1116–1120, Oct. 2000.
25. M. Yeasin, E. Polat, and R. Sharma. A Multiobject Tracking Framework for Interactive Multimedia Applications. *IEEE Trans. on Multimedia*, 6(3):398–405, Jun. 2004.
26. W. Zeng and S. Lei. Efficient Frequency Domain Video Scrambling for Content Access Control. In *Proc. ACM Multimedia, Orlando, FL, USA*, pages 285–293, Nov. 1999.