A Reversible Data Hiding Method for Encrypted Images

W. Puech, M. Chaumont and O. Strauss

LIRMM Laboratory, UMR CNRS 5506, University of Montpellier II
161, rue Ada, 34392 MONTPELLIER CEDEX 05, FRANCE

ABSTRACT

Since several years, the protection of multimedia data is becoming very important. The protection of this multimedia data can be done with encryption or data hiding algorithms. To decrease the transmission time, the data compression is necessary. Since few years, a new problem is trying to combine in a single step, compression, encryption and data hiding. So far, few solutions have been proposed to combine image encryption and compression for example. Nowadays, a new challenge consists to embed data in encrypted images. Since the entropy of encrypted image is maximal, the embedding step, considered like noise, is not possible by using standard data hiding algorithms. A new idea is to apply reversible data hiding algorithms on encrypted images by wishing to remove the embedded data before the image decryption. Recent reversible data hiding methods have been proposed with high capacity, but these methods are not applicable on encrypted images. In this paper we propose an analysis of the local standard deviation of the marked encrypted images in order to remove the embedded data during the decryption step. We have applied our method on various images, and we show and analyze the obtained results.

1. INTRODUCTION

The amount of digital images has increased rapidly on the Internet. Image security becomes increasingly important for many applications, e.g., confidential transmission, video surveillance, military and medical applications. For example, the necessity of fast and secure diagnosis is vital in the medical world.[1,2] Nowadays, the transmission of images is a daily routine and it is necessary to find an efficient way to transmit them over networks. To decrease the transmission time, the data compression is necessary. The protection of this multimedia data can be done with encryption or data hiding algorithms. Since few years, a problem is to try to combine compression, encryption and data hiding in a single step. For example, some solutions was proposed in[3] to combine image encryption and compression. Two main groups of technologies have been developed for this purpose. The first one is based on content protection through encryption. There are several methods to encrypt binary images or gray level images.[3–6] In this group, proper decryption of data requires a key. The second group bases the protection on digital watermarking or data hiding, aimed at secretly embedding a message into the data.[7,8] These two technologies can be used complementary[9,10] and mutually commutative.[11] Sinha and Singh proposed a technique to encrypt an image for secure image transmission.[12] In their approach the digital signature of the original image is added to the encoded version of the original image. The encoding of the image is done using an appropriate error control code. At the receiver end, after the decryption of the image, the digital signature can be used to verify the authenticity of the image. Encryption and watermarking algorithms rely on the Kerckhoffs principle[13]: all the details of the algorithm are known, and only the key to encrypt and decrypt the data should be secret.

Nowadays, a new challenge consists to embed data in encrypted images. Previous work proposed to embed data in an encrypted image by using an irreversible approach of data hiding.[14] The challenge was to find an encryption method robust to noise. Since the entropy of encrypted image is maximal, the embedding step, considered like noise, is not possible by using standard data hiding algorithms. A new idea is to apply reversible data hiding algorithms on encrypted images by wishing to remove the embedded data before the image decryption. Recent reversible data hiding methods have been proposed with high capacity,[15,16] but these methods are not applicable on encrypted images. In this paper we propose an analysis of the local standard deviation of the marked encrypted images in order to remove the embedded data during the decryption step.

william.puech@lirmm.fr, marc.chaumont@lirmm.fr, olivier.strauss@lirmm.fr

The rest of the paper is organized as follows. Section 2 presents the principle of image encryption by using AES algorithm and details the proposed reversible data hiding method for encrypted images. In Section 3, we show and analyze results of the proposed method applied to real images. Conclusion are finally drawn in Section 4.

2. PROPOSED METHOD

2.1. Image encryption

The use of computer networks for data transmissions has created the need of security. Many robust message encryption techniques have been developed to supply this demand. The encryption process can be symmetric, asymmetric or hybrid[17] and can be applied to blocks or streams.[18–20] Several asymmetric algorithms use long keys to ensure the confidentiality because a part of the key is known. These algorithms are not appropriate enough to be applied to images because they require a high computational complexity. In the case of block encryption methods applied to images, one can encounter three inconveniences. The first one is when we have homogeneous zones (regions with the same color), all blocks in these zones are encrypted in the same manner. The second problem is that block encryption methods are not robust to noise. Indeed, because of the large size of the blocks (which is at least of 128 bits) the encryption algorithms per block, symmetric or asymmetric, cannot be robust to noise. The third problem is data integrity. The combination of encryption and data-hiding can solve these types of problems.

The Advanced Encryption Standard (AES) algorithm consists of a set of processing steps repeated for a number of iterations called rounds.[21] 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 $\{Y_1, Y_2, ..., Y_n\}$, as described in the scheme from Fig. 1.

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. In the SubBytes step, each byte of the block is replaced by its substitute in a substitution box (S-Box). In cryptography, an S-box is a basic component of symmetric key algorithms used to obscure the relationship between the plaintext and the ciphertext. The next one is the ShiftRows step where the rows are cyclically shifted over different offsets. The next step is the MixColumns, where each column is multiplied with a matrix over the Gallois Field, denoted as GF(2^8). The last step of the round operation is another AddRoundKey. It is a simple XOR with the actual data and the subkey for the current round. Before producing the final ciphered data Y_i, the AES performs an extra final routine that is composed of (SubBytes, ShiftRows and AddRoundKey) steps, as shown in Fig. 1.

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).[22] 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. The keystream element Z_i is then generated and the ciphertext block Y_i is produced 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

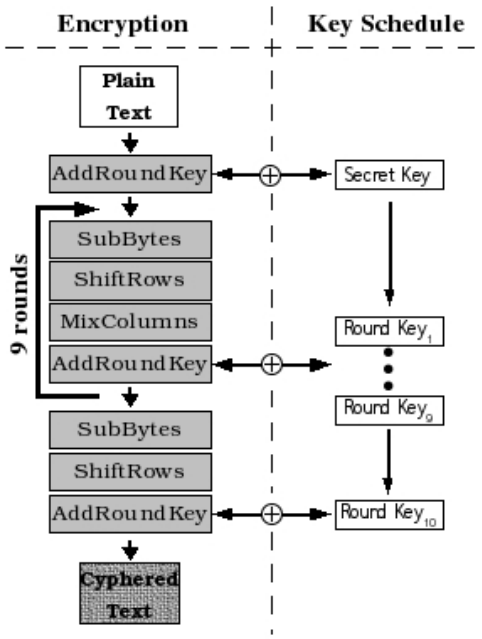


Figure 1. The scheme of the AES algorithm containing 9 rounds of processing steps.

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.

In this paper, for the proposed method, the ECB mode of AES algorithm has been chosen to encrypt the images. The images are thus encrypted by blocks of 128 bits which correspond to 16 gray level pixels. We can first measure the image information content with the entropy $H(X)$. If an image X has M gray levels $\alpha_\jmath$, with $0 \leq \jmath < M$, and the probability of gray level $\alpha_\jmath$ is $P(\alpha_\jmath)$, the entropy $H(X)$, without considering the correlation of gray levels, is defined as:

$$H(X) = -\sum_{\jmath=0}^{M-1} P(\alpha_\jmath) log_2(P(\alpha_\jmath)). \tag{2}$$

If the encryption algorithm is efficient, the entropy $H(Y)$ of an encrypted image Y must be maximal and then greater than the entropy $H(X)$ of the original image X:

$$H(Y) \geq H(X). \tag{3}$$

2.2. Encoding algorithm

The coding algorithm is composed of two steps which are the encryption and the data hiding step. The overview of the encoding method is shown in Fig. 2.

For each block X_i composed of n pixels p_j of an image of N pixels, we apply the AES encryption algorithm by block:

$$Y_i = E_k(X_i), \tag{4}$$

where $E_k()$ is the encryption function with the secret key k and Y_i is the corresponding cipher-text to X_i. One can note that the sizes of X_i and Y_i are identical.

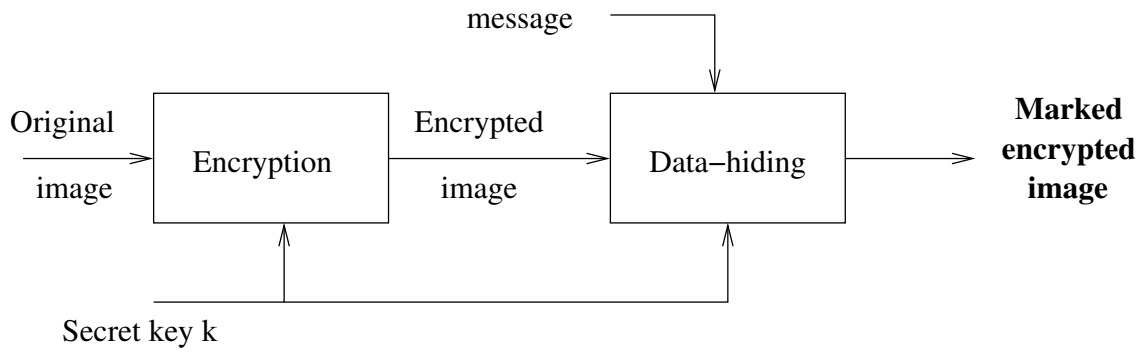


Figure 2. Overview of the encoding method.

During the data hiding step, in each cipher-text we modify only one bit of one encrypted pixel of Y_i:

$$Yw_i = DH_k(Y_i), \tag{5}$$

where $DH_k()$ is the data hiding function with the secret key k and Yw_i is the marked cipher-text. We used bit substitution-based data hiding method in order to embed the bits of the hidden message. For each block Y_i, the secret key k is used as the seed of the pseudo-random number generator (PRNG) to substitute the bit of a pixel with the bit to hidden. At the end of the coding process we get a marked encrypted image. Since we embed 1 bit in each block of n pixels, the embedding factor is equal to $1/n$ bit per pixel.

2.3. Decoding algorithm

The decoding algorithm is also composed of two steps which are the extraction of the message and the decryption-removing. The overview of the decoding method is presented in the scheme from Fig. 3. The extraction of the message is very simple: it is just enough to read the bits of the pixels we have marked by using the secret key k and the same PRNG. But after the extraction, each marked cipher-text is still marked. The problem is then to decrypt the marked encrypted image. The decryption removing is done by analyzing the local standard deviation during the decryption of the marked encrypted images.

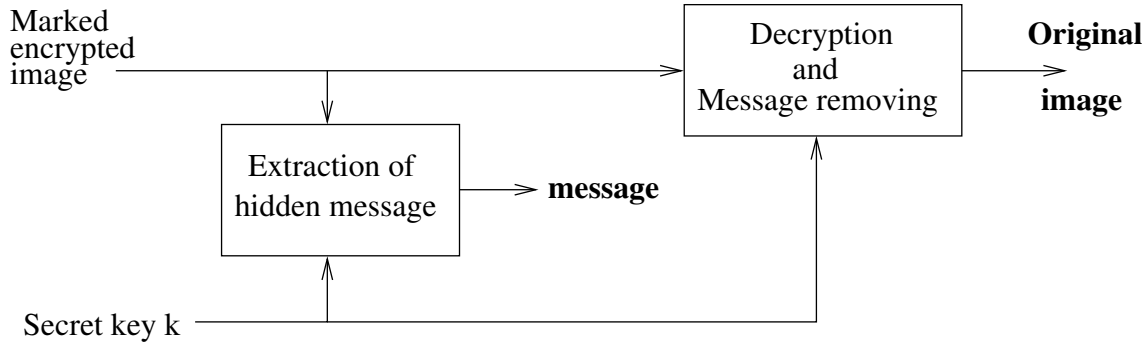


Figure 3. Overview of the decoding method.

To analyze the variation of the local standard deviation σ for each block X_i, taking account of its neighbors to calculate the local mean $\overline{X_i}$, we have:

$$\sigma(X_i) = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(p_j - \overline{X_i})^2}, \tag{6}$$

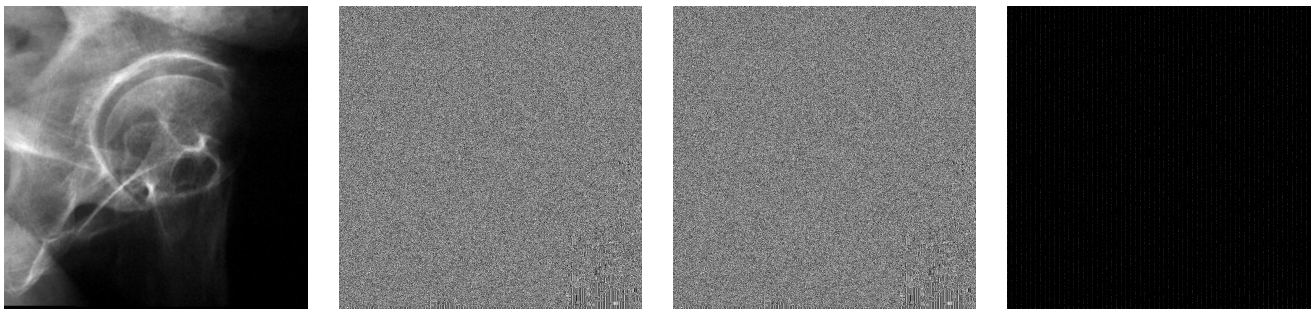
with n the size of the pixel block to calculate the local mean and standard deviation, and $0 \leq i < \frac{N}{n}$, if N is the image size.

For each marked cipher-text Yw_i we apply the decryption function $D_k()$ for the two possible values of the hidden bit (0 or 1) and we analyze the local standard deviation of the two decrypted blocks $X0_i$ and $X1_i$. In the encrypted image, the entropy must be maximal and greater than the original one as described in Equ. (3). Moreover, the local standard deviation of the encrypted image is higher than for an original image. From this assumption we decided to compare for each block the local standard deviation of $X0_i$ with $X1_i$ and we select the bit value where the local standard deviation is the smaller:

$$\begin{cases} X_i = D_k(Y0_i) \text{ if } \sigma(D_k(Y0_i)) < \sigma(D_k(Y1_i)) \\ \quad\; = D_k(Y1_i) \text{ else.} \end{cases} \tag{7}$$

3. RESULTS AND CONCLUSION

We have applied our method on various gray level images and we show the results of the proposed method applied on a medical image (1024×1024 pixels) illustrated in Fig. 4.a and the image of Baboon (512×512 pixels), Fig. 7.a. We have encrypted the original image Fig. 4.a by using the AES algorithm in ECB mode to get the encrypted image illustrated in Fig. 4.b. The size of the blocks is 16 pixels (128 bits). From this encrypted image we have then embedded 65536 bits to get the marked and encrypted image illustrated in Fig. 4.c. The image difference between the Fig. 4.b and c is illustrated in the Fig. 4.d. We can see the pixels where we have substituted one bit with the message. The PSNR of the marked and encrypted image illustrated in Fig. 4.c equals to 66.13 dB.



(a) (b) (c) (d)

Figure 4. a) Original medical image of 1024×1024 pixels, b) Encrypted image with AES in ECB mode, c) Encrypted and marked image with 65536 hidden bits, d) Difference between b) and c).

In Fig. 4.b and c, one can notice that the initial information is not visible anymore. By comparing the histogram of the initial image, Fig. 5.a, with that of the encrypted image, Fig. 5.b, we notice that the probabilities of appearance of every grey level are equitably distributed. The histogram of the encrypted image is flat, and from equation (2) we get very high entropy $H(Y)$ of 7.997 bits/pixel ($H(X) = 7.216$ bits/pixel for the original image). The information redundancy is very small and thus statistical attacks would be difficult.[22] From equation (6) we also analyzed the variation of the local standard deviation σ for each pixel while taking its neighbors into

account. The mean local standard deviation is equal to 68.278 gray levels for the marked encrypted image illustrated Fig. 4.c (the mean local standard deviation is equal to 1.349 gray levels for the original medical image Fig. 4.a.). Figs. 5.d and e illustrate the local standard deviation of the original medical image and of the marked encrypted image. These analyzes show that the encrypted images are protected against statistical attacks.

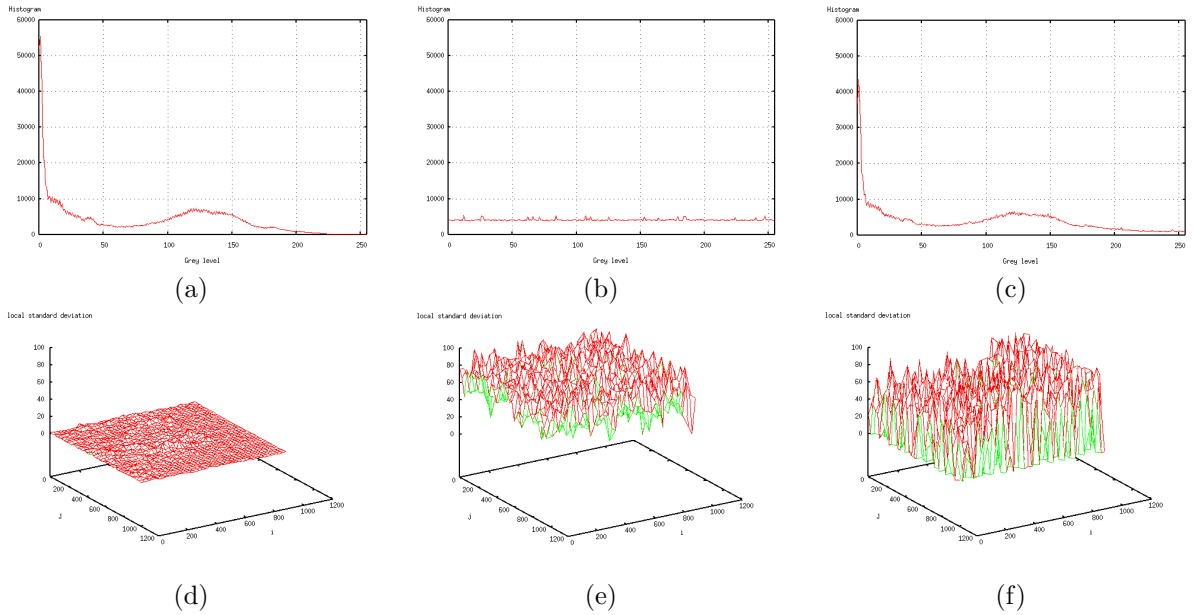


Figure 5. Histogram of: a) The original image, b) The marked encrypted image Fig. 4.c, c) The decrypted image Fig. 6.a., Local standard deviation of: d) The original image, e) The marked encrypted image Fig. 4.c, f) The decrypted image Fig. 6.a.

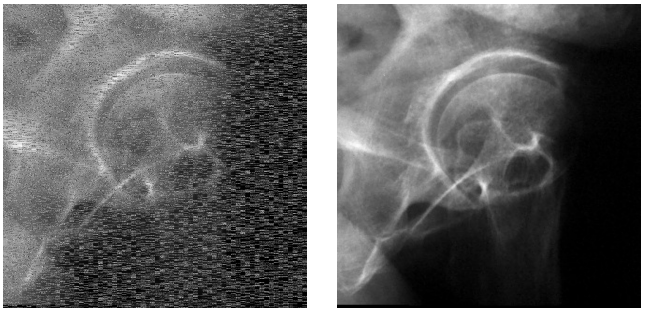
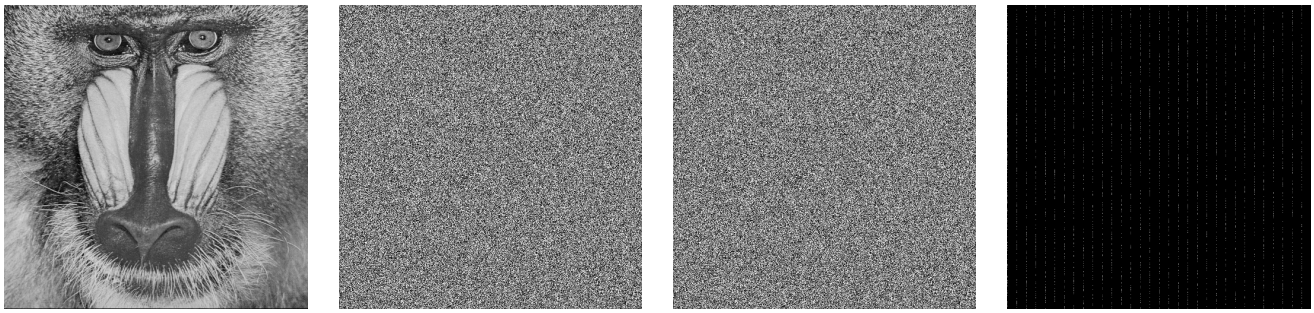


Figure 6. a) Extraction of the message and decryption of the marked image Fig 4.c, b) Decryption and deleting of the message by using the proposed method.

For the decoding process, after the message extraction, if we apply only the decryption on the image of Fig. 4.c, we get the image illustrated in Fig. 6.a. The histogram of this decrypted image is illustrated in Fig. 5.c. Even if this histogram looks similar to the original one, the quality of this decrypted image is very bad and its PSNR equals to 13.27 dB. The mean local standard deviation is equal to 34.010 gray levels for this image, and its local standard deviation is illustrated in Fig. 5.f. By analyzing the local standard deviation for each block during the decryption step we are able to find the original value of each bit and thus to remove the hidden data. The application of the equation (7) during the decryption step allows us to get the decrypted image illustrated

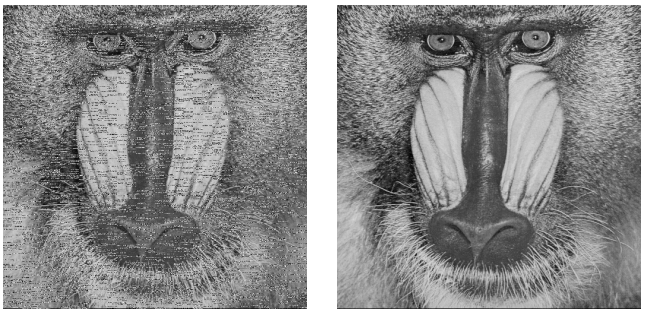
in Fig. 6.b. This decrypted image is exactly the original image with a $PSNR = \infty$.

We have applied the same process to the image of Baboon, Fig. 7.a, by using the AES algorithm in ECB mode to get the encrypted image illustrated in Fig. 7.b. The size of the blocks is also 16 pixels (128 bits). From this encrypted image we have then embedded 16384 bits to get the marked and encrypted image illustrated in Fig. 7.c. The image difference between the Fig. 7.b and c is illustrated in the Fig. 7.d. For the decoding process, after the extraction, if we apply only the decryption on the image of Fig. 7.b, we get the image illustrated in Fig. 8.a. By analyzing the local standard deviation for each block during the decryption step we are able to find the original value of each bit and thus to remove the hidden data and to get the decrypted image illustrated in Fig. 8.b.



(a) (b) (c) (d)

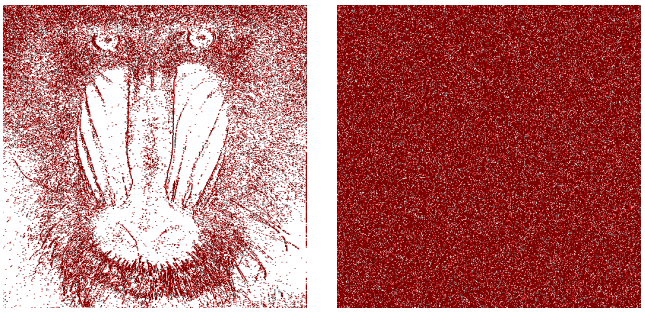
Figure 7. a) Original image of 512×512 pixels, b) Encrypted image with AES in ECB mode, c) Encrypted and marked image with 16384 hidden bits, d) Difference between b) and c).



(a) (b)

Figure 8. a) Extraction of the message and decryption of the marked image Fig 7.c, b) Decryption and deleting of the message by using the proposed method.

To compare our proposed method, we show in Fig. 9 the result of the application of a reversible data hiding method.[16] The Fig. 9.a illustrates the capacity of the method[16] applied to the image of Baboon. The 170914 white pixels correspond to the pixels where we can embed 2 or 3 bits per pixel. But correctives codes must be embedded in order to retrieve the original image. If there is not enough white pixels, then it is not possible to embed an useful message in the image. This is the case if we try to apply the method[16] on the encrypted image of the Fig. 7.b. Indeed in the Fig. 9.b, the number of white pixels is equal to 28 352, but 116 896 correctives codes are necessary. It is thus not possible to use high capacity reversible data hiding method[16] for encrypted images.



(a) (b)

Figure 9. Application of a reversible data hiding method[16]: a) On the original image of Baboon, b) On the encrypted image of the Fig. 7.b.

4. CONCLUSION

In conclusion, with our proposed reversible data hiding method for encrypted images we are able to embed data in encrypted images and then to decrypt the image and to rebuild the original image by removing the hidden data. In this paper, we detailed all the steps of the proposed method and we illustrated the method with schemes. We presented and analyzed various results by showing the plots of the local standard deviations.

In the proposed method, the embedding factor is 1 bit for 16 pixels. This small value of the embedding factor is only is to have to choose between two values for each block during the decryption. For the future, we are thinking to improve this method by increasing the payload but also the complexity.

REFERENCES

1. J. Bernarding, A. Thiel, and A. Grzesik, "A JAVA-based DICOM server with integration of clinical findings and DICOM-conform data encrytion," *International Journal of Medical Informatics* **64**, pp. 429–438, 2001.
2. R. Norcen, M. Podesser, A. Pommer, H. Schmidt, and A. Uhl, "Confidential Storage and Transmission of Medical Image Data," *Computers in Biology and Medicine* **33**, pp. 277–292, 2003.
3. A. Uhl and A. Pommer, *Image and Video Encryption: From Digital Rights Management to Secured Personal Communication*, Springer, 2005.
4. K. Chung and L. Chang, "Large encrypting binary images with higher security," *Pattern Recognition Letters* **19**, pp. 461–468, 1998.
5. C. Chang, M. Hwang, and T.-S. Chen, "A new encryption algorithm for image cryptosystems," *The Journal of Systems and Software* **58**, pp. 83–91, 2001.
6. A. Sinha and K. Singh, "A technique for image encryption using digital signature," *Optics Communications* **218**, pp. 229–234, 2003.
7. A. Eskicioglu and E. Delp, "An Overview of Multimedia Content Protection in Consumer Electronics Devices," *Signal Processing: Image Communication* **16**(7), pp. 681–699, 2001.
8. F. Y. Shih and S. Y. Wu, "Combinational image watermarking in the spatial and frequency domains," *Pattern Recognition* **36**, pp. 969–975, 2003.
9. X. Xu, S. Dexter, and A. Eskicioglu, "A Hybrid Scheme for Encryption and Watermarking," in *Proc. of SPIE, Electronic Imaging, Security and Watermarking of Multimedia Contents VI*, P. Wong and E. Delp, eds., **5306**, pp. 725–736, SPIE, IS&T, (San Jose, CA, USA), January 2004.
10. A. Lemma, S. Katzenbeisser, M. Celik, and M. van der Veen, "Secure Watermark Embedding through Partial Encryption," in *International Workshop on Digital Watermarking (IWDW 2006)*, **4283**, pp. 433–445, Springer Lecture Notes in Computer Science, 2006.

11. S. Lian, Z. Liu, R. Zhen, and H. Weng, "Commutative watermarking and encryption for media data," *Optical Engineering* **45**(8), pp. 080510–1–080510–3, 2006.
12. A. Sinha and K. Singh, "A Technique for Image Encryption Using Digital Signature," *Optics Communications* **218**, pp. 229–234, April 2003.
13. A. Kerckhoffs, "La cryptographie militaire," *Journal des sciences militaires* **9**, pp. 5–38, 1883.
14. W. Puech and J. Rodrigues, "A New Crypto-Watermarking Method for Medical Images Safe Transfer," in *EUSIPCO'04, Vienna, Austria*, pp. 1481–1484, 2004.
15. Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible Data Hiding," *IEEE Trans. on Circuits and Systems for Video Technology* **16**, pp. 354–362, Mar. 2006.
16. D. Coltuc and J.-M. Chassery, "High Capacity Reversible Watermarking," in *Proc. IEEE Int. Conf. on Image Processing, Atlanta, USA*, Oct. 2006.
17. P. Zimmermann, *PGP User's Guide*, MIT Press, Cambridge, 1994.
18. W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. on Information Theory* **26**(6), pp. 644–654, 1976.
19. D. Stinson, *Cryptography - Theory and Practice*, CRC Press, Boca Raton, Florida, USA, 1995.
20. B. Schneier, *Applied cryptography*, Wiley, New-York, USA, 1995.
21. J. Daemen and V. Rijmen, "AES Proposal: The Rijndael Block Cipher," tech. rep., Proton World Int.l, Katholieke Universiteit Leuven, ESAT-COSIC, Belgium, 2002.
22. D. R. Stinson, *Cryptography: Theory and Practice, (Discrete Mathematics and Its Applications)*, Chapman & Hall/CRC Press, New York, November 2005.