International Journal of Research in Engineering, Science and Management
Volume-2, Issue-1, January-2019
www.ijresm.com | ISSN (Online): 2581-5792

Visual Cryptography Data Hiding using LSB Algorithm

K. Seena[1], T. S. Sreejitha[2], Neenu Unnikrishnan[3], Aneega Jose[4], Rajasree Sethu Madhavan[5]

[1,2,3,4]*Assistant Professor, Department of Vocational Studies, St. Mary's College, Thrissur, India*

[5]*Managing Director, Fab Studioz IT Development & Research Division, Thrissur, India*

***Abstract*: It is a novel scheme for visual cryptography data encryption using LSB algorithm. In the initial phase, a content owner encrypts the original uncompressed image using an encryption key with a sharing process. Advance Encryption standard is used to encrypt the image. The data-hider may hide the data on least significant bits of the encrypted image using a data-hiding key to create a sparse space to accommodate some additional data. A sharing process of image is carried out for better security. The image recovery is possible only users when the user gets all shares of the image. With each share of an encrypted image containing additional data, if a receiver has the data-hiding key, he can extract the additional data though he does not know the image content. If the receiver has the encryption key and all shares, he can decrypt the received image similar to the original one, but cannot extract the additional data. If the receiver has both the data-hiding key and the encryption key with shares, he can extract the additional data and recover the original image without any error by exploiting the spatial correlation in natural image when the amount of additional data is not too large.**

***Keywords*: Image encryption, Image recovery, Reversible data hiding, Image sharing, Data Embedding, Watermarking**

1. Introduction

With the continued growth of multimedia applications, security is an important issue in communication and image storage, and encryption is the best way to ensure security. An encryption image technique attempts to convert the original image to another image, which is difficult to identify as the original image. The purpose is to keep the image confidential among users and embedding of data into an encrypted image, in other words, it is essential that not just anyone can determine the contents of the image without a decryption key. In addition, the algorithm can find application where special storage and transmission security and reliability of digital images necessary such as military communication and information technology industries, etc. In fact, the use of a communication network to exchange data presents certain risks, which requires the existence of appropriate security measures. For example, the transmitted images can be saved and copied during their transmission without loss of image quality. Image and data can be hacked in time during an exchange of digital information storage and this is of course illegal. It is therefore necessary to develop a tool for effective protection of transferred data against arbitrary interference.

Data encryption is very often the only effective way to meet these requirements. In this paper we are interested in the security of image and data. These features of the data make the algorithms developed in the literature unusable in their traditional form due to speed limitations and loss of information that can be caused by advanced encryption standard algorithm. It can be argued that there is no particular encryption algorithm, which satisfies the requirements of all image types. In order to decrease the high correlation among pixels and increase the entropy value of an image, we propose a process based on shifting the rows and columns of the image using the following technique. The shifting process will be used to divide the original image into a number of blocks that are then shifted through the rows and the columns within the image based on a shifted table that is generated by another algorithm before the encryption process starts. The image is then fed into the AES encryption algorithm. Fig. 1 gives the sketch. A content owner encrypts the original image using an encryption key, and data sender can embed data in an encrypted image using a data-hiding key though he does not know the original content. With an encrypted image, containing embedded data a receiver may decrypt it according to the encryption key with all shares. In the scheme, the data can be extracted from the encrypted image using data hiding key.

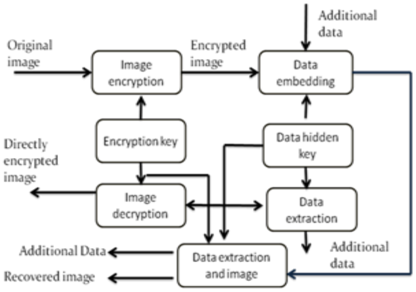

Fig. 1. General block diagram

2. Method

The proposed scheme is made up of image encryption, image sharing, data embedding and data-extraction/image-recovery phases. The content owner encrypts the original uncompressed image using an encryption key to produce an encrypted image. The data-hider hides the secret data on the least significant bits (LSB) of the encrypted image using a data-hiding key to create a sparse space to accommodate the additional data. A sharing process of image is then carried out. At the receiver side, the data embedded in the created space can be easily retrieved from the encrypted image containing additional data according to the data-hiding key. Since the data embedding only affects the LSB, a decryption with the encryption key with all shares can result in an image similar to the original version. When using both of the encryption and data-hiding keys, the embedded additional data can be successfully extracted and the original image can be perfectly recovered by exploiting the spatial correlation in natural image. Fig. 2 shows the three cases at the receiver side.

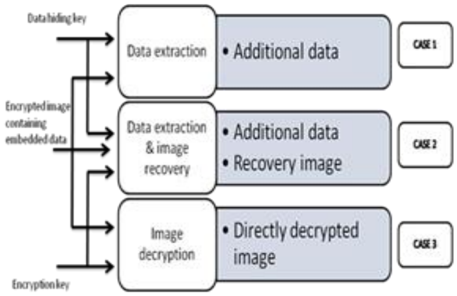

Fig. 2. Three cases at receiver side of the proposed separable scheme.

A. *Image encryption*

The Advanced Encryption Standard (AES) computer security standard is a symmetric block cipher that encrypts and decrypts 128-bit blocks of data. Standard key lengths of 128, 192, and 256 bits may be used. The algorithm consists of four stages that make up a round which is iterated 10 times for a 128-bit length key, 12 times for a 192-bit key, and 14 times for a 256-bit key. The first stage “SubBytes” transformation is a non-linear byte substitution for each byte of the block. The second stage “ShiftRows” transformation cyclically shifts (permutes) the bytes within the block. The third stage “MixColumns” transformation groups 4-bytes together forming 4-term polynomials and multiplies the polynomials with a fixed polynomial mod (x^4+1). The fourth stage “AddRoundKey” transformation adds the round key with the block of data. In most ciphers, the iterated transform (or round). Typically, in this structure, some of the bits of the intermediate state are transposed unchanged to another position (permutation). AES is composed of three distinct invertible transforms based on the Wide Trial Strategy design method. The Wide Trial Strategy design method provides resistance against linear and differential cryptanalysis. In the Wide Trail Strategy, every layer has its own function:

- The linear mixing layer: guarantees high diffusion over multiply rounds.
- The non-linear layer: parallel application of S-boxes that have the optimum worst-case non-linearity properties.
- The key addition layer: a simple XOR of the round key to the intermediate state.

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 and can be applied to blocks or streams.

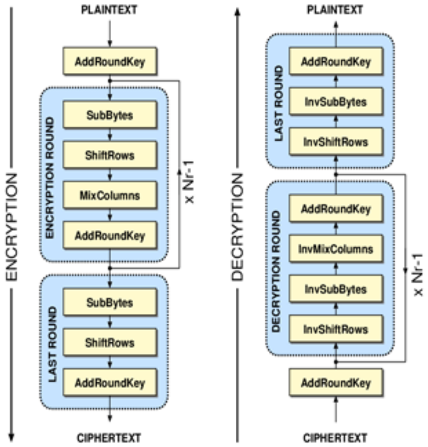

Fig. 3. Image encryption/ decryption process

To encipher a data block Xi in AES you first perform an Add Round Key step by XORing a sub key with the block. 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 embedding of data 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.

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 {X1,X2, ...,Xn} of bit plaintext blocks, each Xi is encrypted with the same secret key k producing the cipher text blocks {Y1, Y2, ..., Yn}, as described in the scheme. 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 cipher text. The next one is the ShiftRows step where the rows are cyclically shifted over different offsets. The next step is the Mix Columns, where each column is multiplied with a matrix over the Gallois Field, denoted as GF. 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 Yi, the AES performs an extra final routine that is composed of (SubBytes, ShiftRows and AddRoundKey) steps, as shown in Fig. 3

Table 1
Algorithm details

Algorithm	Key Length (Nk Words)	Block Size (Nb Words)	Number of rounds (Nr)
AES – 128	4	4	10
AES – 192	6	4	12
AES – 256	8	4	14

B. *Embedding of data*

In the embedding of data phase, some parameters are embedded into a small number of encrypted pixels, and the Least Significant Bit of the other encrypted pixels are compressed to create a space for accommodating the additional data and the original data at the positions occupied by the parameters. The detailed procedure is as follows.

According to a data-hiding key, the data-hider pseudo-randomly selects Np encrypted pixels that will be used to carry the parameters for data hiding. Here, Np is a small positive integer, for example, Np =20.The other $(N-Np$) encrypted pixels are pseudo-randomly permuted and divided into a number of groups, each of which contains L pixels. The permutation way is also determined by the data-hiding key.

For each pixel-group, collect the M least significant bits of the L pixels, and denote them as $B(k,1),B(k,2),..B(k,M.L)$ where k is a group index within $(1,(N-Np)/L$ and M is a positive integer less than 5. The data-hider also generates a matrix G sized, $(M.L-S)\times M.L$ which is composed of two parts

$$G=[I_{M.L-S}Q]$$

While the left part is an $(M.L-S)\times M.L-S$ identity matrix, the right part Q sized $(M.L-S)\times S$ is a pseudo-random binary matrix derived from the data-hiding key. Here, S is a small positive integer. Then, embed the values of the parameters M,L and S into the LSB of Np selected encrypted pixels. The permutation way is also determined by the data- sender may represent the values of M, L and S as 2, 14 and 4 bits, respectively, and replace the LSB of selected encrypted pixels with the 20 bits. In the following a total of $(N-Np).S/L$) bits made up of Np original LSB of selected encrypted pixels and (N- $N-Np.S/L-Np$ embedded in the pixel-group, For each group, calculate

$$\begin{bmatrix} B'(k,1) \\ B'(k,2) \\ : \\ : \\ B'(k,M.L-S) \end{bmatrix} = G.\begin{bmatrix} B(k,1) \\ B(k,2) \\ : \\ : \\ B(k,M.L-S) \end{bmatrix}_{(2)}$$

Where the arithmetic is modulo-2 by (2), $[B(k,1),B(k,2),..B(k,M.L)]$ are compressed as $(M.L-S)$ bits, and a sparse space is therefore available for data accommodation.

Let $[B'(k,M.L-S+1),B'(k,M.L-S+2),...B'(k(k,M.L)]$ of each group be the original LSB of selected encrypted pixels and the additional data to be embedded.

Replace them $B(k,1),B(k,2),..B(k,M.L)$ with the new $B'(k,1),B'(k,2),..B'(k,M.L)$ and put them into their original positions by an inverse permutation. At the same time, the (8-M) most significant bits (MSB) of encrypted pixels are kept unchanged. Since S bits are embedded into each pixel – group, the total $(N-Np).S/L$ bits can be accommodated in all groups. Clearly, the embedding rate, a ratio between the data amount of net payload and the total number of cover pixels, is $R=\frac{((N-Np).S/L-Np)}{N}\approx\frac{S}{L}$

1) *Image sharing*

Embedding of data to the image phase follow an image sharing process. The image is spited on equal shares and then the encrypted image will be sent to the receiver. The receiver can extract the image if they have all shares in hand. The data can be extracted from the image if receiver has valid decryption key

2) *Watermark embedding*

This is an optional phase. In this phase the watermark may give the identification of the provider. It is an optional phase. The embedding technique of watermark is given as follows:

- Assume that the size of the host image is 512×512. Host image is divided into small M×M blocks Z, block Z is divided into small M ×M blocks Y. If M=8 is used, the size of block Y is 8×8.

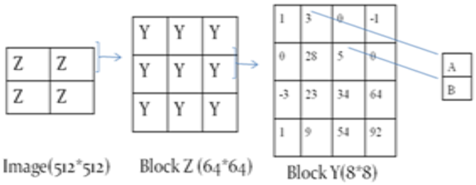

Fig. 4. Watermark embedding

- A number of pairs of coefficients (A, B) in block Y are chosen as A = a1, . . . , an, B = b1. . . bn based on a pseudo-random numbers, and mapping key that contains index of original chosen coefficients are kept.
- For embedding, two coefficient values (ai, bi) are modified by add parameter, which is a parameter for watermark strength. i=1,…,n.
- Continue the above process according to n. Each block Y is embedded 1 bit watermark and watermark length decides how many blocks Y is embedded.

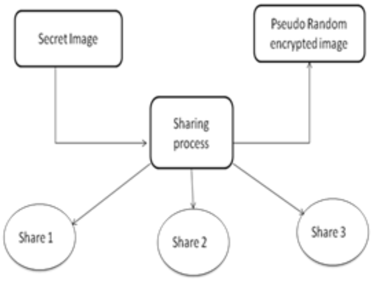

Fig. 5. Image sharing process

C. Watermark extraction

This phase is backward process of above process. The watermark extraction process is discussed as follows. First, choose pseudo-random numbers and mapping key to assign two pixels (ai, bi) for n pairs from each block and modified value of the assigned pixels after embedded watermark. For extraction, choose the same pairs, according tithe function, the watermark is extracted. Product Provider decrypts the extracted watermark by chaotic sequence and can restore the user's message.

D. Data extraction and image recovery

In this phase, we will consider the three cases that a receiver has only the embedding key, only the encryption key, and both the embedding key and encryption keys, respectively. With an encrypted image containing embedded data, if the receiver has only the embedding key, he may first obtain the values of the parameters M,L and S from the LSB of the Np selected encrypted pixels. Then, the receiver permutes and divides the other (N-Np). Pixel s into(N-Np)/L groups and extracts the S embedded bits from the M LSB-planes of each group. When having the total (N-Np). S/L extracted bits, the receiver can divide them into Np original LSB of selected encrypted pixels and (N-Np).S/L-Np additional bits. Note that because of the pseudo-random pixel selection and permutation, any attacker without the embedding key cannot obtain the parameter values and the pixel-groups, therefore cannot extract the embedded data. Furthermore, although the receiver having the data-hiding key can successfully extract the embedded data, he cannot get any information about the original image content.

3. Experimental results

The test image Lena sized 512 x512 shown in Fig. 9(a) was used as the original image in the experiment. After image encryption, the eight encrypted bits of each pixel are converted into a gray value to generate an encrypted image shown in Fig. 9(b). Then, we M=3, L=128 and S=2to embed 4.4 x103additional bits into the encrypted image. The encrypted image containing the embedded data is shown in Fig. 9(c). With an encrypted image containing embedded data, we could extract the data using the embedding key. If we directly decrypted the encrypted image containing embedded data using the encryption key, the value of PSNR in the decrypted image was 39.0 dB, which verifies the theoretical value 39.1 dB calculated.

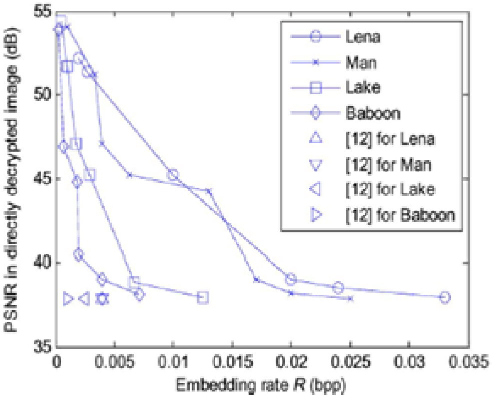

Fig. 6. Rate-PSNR proposed scheme

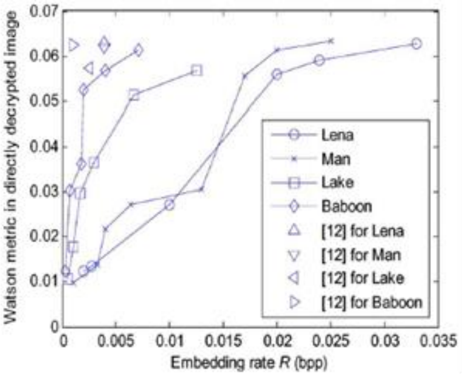

Fig. 7. Rate-Watson metric proposed scheme

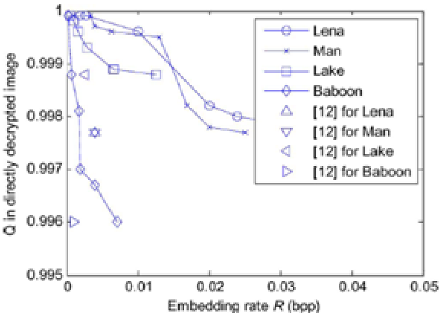

Fig. 8. Comparison between the proposed schemes and data hiding method

Fig. 9 (a). Original image

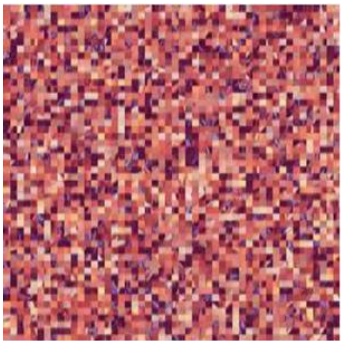

Fig. 9 (b). Encrypted version

The directly decrypted image is given as Fig. 9 (d). By using both the data hiding and the encryption keys, the embedded data could be successfully extracted and the original image could be perfectly recovered from the encrypted image containing embedded data. Decrypted Images and Pulse to Signal Noise Ratio in recovered images when different M, L and S were used for images Lena and Man. The embedding rate is dependent on S and L, and the larger Sand the smaller correspond to a higher embedding rate. On the other hand, the smaller the values of M and S, the quality of directly decrypted image is better since more data in encrypted image are not changed by data embedding. Here, the large M, L and the small S are helpful to the perfect content recovery since more cover data and less possible solutions are involved in the recovery procedure. It shows the rate-distortion curves of the four images, Man, Lake and Baboon.

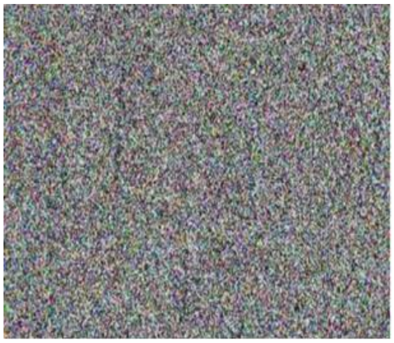

Fig. 9 (c). Encrypted image containing embedded data

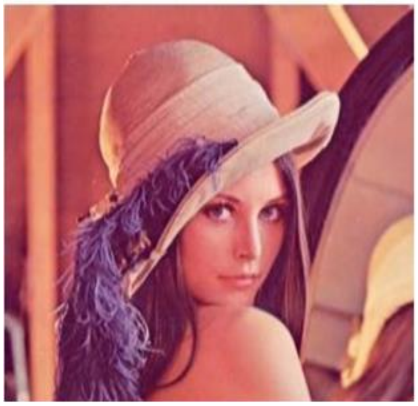

Fig. 9 (d). Decrypted image

Here, three quality metrics were used to measure the distortion in directly decrypted image: PSNR, the Watson metric and a universal quality index Q. In these figures, while the abscissa represents the embedding rate, the ordinate is the values of PSNR, Watson metric or quality index Q. While PSNR simply indicates the energy of distortion caused by data hiding, the Watson metric is designed by using characteristics of the human visual system and measures the total perceptual error, which is takes into account three factors: contrast sensitivity, luminance masking and contrast masking. Additionally, the quality index Q works in distortion and contrast distortion. Higher PSNR, lower Watson metric or higher Q means better quality. In these figures, while the abscissa represents the embedding rate, the ordinate is the values of PSNR, Watson metric or quality index Q. The curves are derived from different M, L and S under a condition that the original content can be perfectly recovered using the data hiding and encryption keys. Since the spatial correlation is exploited for the content recovery, the rate-distortion performance in a smoother image is better. The performance of the non-separable method is also given in Figs. 6-8. It can be seen that the performance of the proposed separable scheme is significantly better than that existing system.

4. Conclusion

In this paper, a novel scheme for visual cryptography data hiding in encrypted image is proposed, which consists of image encryption, data embedding and data-extraction/image-recovery phases. In the first phase, the content owner encrypts the original uncompressed image using an encryption key. A sharing process is also conducted for sharing the image to number of users who uses the system. Each share will be transmitted to the receivers. Only those who got all share a reveal the image properly. Although a data-hider does not know the original content, he can compress the least significant bits of the encrypted image using a data-hiding key to create a sparse space to accommodate the additional data. With an encrypted image containing additional data, the receiver may extract the additional data using only the data-hiding key, or obtain an image similar to the original one using only the encryption key. When the receiver has both of the keys, he can extract the additional data and recover the original content without any error by exploiting the spatial correlation in natural image if the amount of additional data is not too large. If the lossless compression method in or is used for the encrypted image containing embedded data, the additional data can be still extracted and the original content can be also recovered since the lossless compression does not change the content of the encrypted image containing embedded data.

Acknowledgments

I must thank, first and foremost my Internal Guide Mrs. B. Hemalatha M.E., Assistant Professor, Department of Computer Science and Engineering and Project Coordinator Mrs. M Bharanishree M.E., Assistant Professor, Department of

Computer Science and Engineering without whose guidance and patience this dissertation would not be possible. I wish to record my thanks to Mrs. A. Kanchana M.E., (Ph.D) Head of the Department, Department of Computer Science and Engineering, Project Panel Members, Professors of the Department of Computer Science and Engineering for their consistent encouragement and ideas.

References

[1] P. Comesana, L. Perez-Freire, and F. Perez-Gonzalez, “Fundamentals of data hiding security and their application to spread-spectrum analysis,” in *Proc. 7th Inf. Hiding orkshop (IH 2005), Lectures Notes In Computer Science, Springer-Verlag*, Barcelona, Spain, Jun. 2005, vol.3727, pp. 146–160.

[2] F. Cayre, C. Fontaine, and T. Furon, “Watermarking security: Theoryand practice,” *IEEE Trans. Signal Process.*, vol. 53, no. 10, pp.3976–3987, Oct. 2005.

[3] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, “Secure spreadspectrum watermarking for multimedia,” *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1673–1687, Dec. 1997.

[4] P. Moulin and A. Ivanovic, “The zero-rate spread-spectrum watermarking game,” *IEEE Trans. Signal Process.*, vol. 51, no. 4, pp.1098–1117, Apr. 2003.

[5] H. S. Malvar and D. A. F. Florencio, “Improved spread spectrum: A new modulation technique for robust watermarking,” *IEEE Trans. Signal Process.*, vol. 51, no. 4, pp. 898–905, Apr. 2003.

[6] B. Mathon, P. Bas, F. Cayre, and B. Macq, Optimization of natural watermarking using transportation theory,” in *Proc. 11th ACM Workshopon Multimedia and Security (MM&Sec’2009)*, Princeton, NJ, Sep.009, pp. 33–38.

[7] P. Bas and F. Cayre, “Achieving subspace or key security for woa using natural or circular Watermarking,” in *Proc. 8th ACM Workshop Multimedia and Security* Geneva, Switzerland, Sep. 2006, pp. 80–88.

[8] F. Cayre and P. Bas, “Kerckhoffs-based embedding security classes for’woa data hiding,” *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 1, pp.’1–15, Mar. 2008.

[9] J. Cao, J. Huang, and J. Ni, “A new spread-spectrum watermarking scheme to achieve a trade-off between security and robustness,” in*Proc. 12th Inf. Hiding Workshop,Lectures Notes in Computer Science, Springer-Verlag*, Calgary, AB, Canada, Jun. 2010, vol.6387, pp. 262–276.*Acad. Sci. URSS* vol. 37, pp. 199–201, 1942.

[10] M. Johnson, P. Ishwar, V. M. Prabhakaran, D. Schonberg, and K.Ramchandran, “On compressing encrypted data,” *IEEE Trans. Signal Process*., vol. 52, no. 10, pp. 2992–3006, Oct. 2004.

[11] X. Zhang, "Separable Reversible Data Hiding in Encrypted Image," in *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 826-832, April 2012.

[12] X. Zhang, “Reversible data hiding in encrypted image,” *IEEE Signal Process. Lett.*, vol. 18, no. 4, pp. 255–258, Apr. 2011.