ISSN: 2278-4632
Vol-11 Issue-01 2021

A NEW SECURE MULTIPLE DATA HIDDING IN IMAGE ENCRYPTED TECHNIQUE BASED AES-CTR SCHEME IN SCRECT SHARING

Dr.V.S.R.Kumari
Principal (Professor)
Affiliated Department of Electronics and Communication Engineering
Sri mittapalli institute of technology for women /JNTUKakinada

Srinivasa Rao Chevala
Assistant Professor
Affiliated Department of Electronics and Communication Engineering
Sri mittapalli institute of technology for women /JNTUKakinada

SK.Ayesha
Assistant Professor
Affiliated Department of Electronics and Communication Engineering
Sri mittapalli institute of technology for women /JNTUKakinada

ABSTRACT: The existing models of reversible data hiding in encrypted images (RDH-EI) are based on single data-hider, where the original image cannot be reconstructed when the data-hider is damaged. To address this issue, this paper proposes a novel model with multiple data-hiders for RDH-EI based on secret sharing. It divides the original image into multiple different encrypted images with the same size of the original image and distributes them to multiple different data-hiders for data hiding. Each data-hider can independently embed data into the encrypted image to obtain the corresponding marked encrypted image. The original image can be losslessly recovered by collecting sufficient marked encrypted images from undamaged data-hiders when individual data-hiders are subjected to potential damage. This further protects the security of the original image. We provide four cases of the proposed model, namely, two joint cases and two separable cases. From the proposed model, we derive a separable RDH-EI method with high-capacity. Experimental results are presented to illustrate the effectiveness of the proposed method.

1.INTRODUCTION

Encryption and data hiding are two effective means of data protection. While the encryption techniques convert plaintext content into unreadable ciphertext, the data hiding techniques embed additional data into cover media by introducing slight modifications. In some distortion-unacceptable scenarios, data hiding may be performed with a key modulation manner. Although the terms have a same meaning in a set of previous references, we would distinguish them in this work

We say a data hiding method is lossless if the display of cover signal containing embedded data is same as that of original cover even though the cover data have been modified for data embedding. For example, in [1], the pixels with the most used color in a palette image are assigned to some unused color indices for carrying the additional data, and these indices are redirected to the most used color. This way, although the indices of these pixels are altered, the actual colors of the pixels are kept unchanged. On the other hand, we say a data hiding method is reversible if the original cover content can be perfectly recovered from the cover version containing embedded data even though a slight distortion has been introduced in data embedding procedure. A number of mechanisms, such as difference expansion [2], histogram shift [3] and lossless compression [4], have been employed to develop the reversible data hiding techniques for digital images. Recently, several good prediction approaches [5] and optimal transition probability under payload-distortion criterion [6, 7] have been introduced to improve the performance of reversible data hiding.

Combination of data hiding and encryption has been studied in recent years. In some works, data hiding and encryption are jointed with a simple manner. For example, a part of cover data is used for carrying additional data and the rest data are encrypted for privacy protection [8, 9]. Alternatively, the additional data are embedded into a data space that is invariable to encryption operations [10]. In another type of the works, data embedding is performed in encrypted domain, and an authorized receiver can recover the original plaintext cover image and extract the embedded data. This technique is termed as reversible data hiding in encrypted images (RDHEI). In some scenarios, for securely sharing secret images, a content owner may encrypt the images before transmission, and an inferior assistant or a channel administrator hopes to append some additional messages, such as the origin information, image notations or authentication data, within the encrypted images though he does not know the image content. For example, when medical images have been encrypted for protecting the patient privacy, a database administrator may aim to embed the personal information into the corresponding encrypted images. Here, it may be hopeful that the original content can be recovered without any error after decryption and retrieve of additional message at receiver side. In [11], the original image is encrypted by an exclusive-or operation with pseudo-random bits, and then the additional data are embedded by flipping a part of least significant bits (LSB) of encrypted image. By exploiting the spatial correlation in natural images, the embedded data and the original content can be retrieved at receiver side. The performance of RDHEI can be further improved by introducing an implementation order [12] or a flipping ratio [13]. In [14], each additional bit is embedded into a block of data encrypted by the Advanced Encryption Standard (AES). When a receiver decrypts the encrypted image containing additional data, however, the quality of decrypted image is significantly degraded due to the disturbance of additional data. In [15], the data-hider compresses the LSB of encrypted image to generate a sparse space for carrying the additional data. Since only the LSB is changed in the data embedding phase, the quality of directly decrypted image is satisfactory. Reversible data hiding schemes for encrypted JPEG images is also presented [16]. In [17], a sparse data space for accommodating additional data is directly created by compress the encrypted data. If the creation of sparse data space or the compression is implemented before encryption, a better performance can be achieved [18, 19]. While the additional data are embedded into encrypted images with symmetric cryptosystem in the above-mentioned RDHEI methods, a RDHEI method with public key cryptosystem is proposed in [20]. Although the computational complexity is higher, the establishment of secret key through a secure channel between the sender and the receiver is needless. With the method in [20], each pixel is divided into two parts: an even integer and a bit, and the two parts are encrypted using Paillier mechanism [21], respectively. Then, the ciphertext values of the second parts of two adjacent pixels are modified to accommodate an additional bit. Due to the homomorphic property of the cryptosystem, the embedded bit can be extracted by comparing the corresponding decrypted values on receiver side. In fact, the homomorphic property may be further exploited to implement signal processing in encrypted domain [22, 23, 24]. For recovering the original plaintext image, an inverse operation to retrieve the second part of each pixel in plaintext domain is required, and then two decrypted parts of each pixel should be reorganized as a pixel.

This paper proposes a lossless, a reversible, and a combined data hiding schemes for public-key-encrypted images by exploiting the probabilistic and homomorphic properties of cryptosystems. With these schemes, the pixel division/reorganization is avoided and the encryption/decryption is performed on the cover pixels directly, so that the amount of encrypted data and the computational complexity are lowered. In the lossless scheme, due to the probabilistic property, although the data of encrypted image are modified for data embedding, a direct decryption can still result in the original plaintext image while the embedded data can be extracted in the encrypted domain. In the reversible scheme, a histogram shrink is realized before encryption so that the modification on encrypted image for data embedding does not cause any pixel oversaturation in plaintext domain. Although the data embedding on encrypted domain may result in a slight distortion in plaintext domain due to the homomorphic property, the embedded data can be extracted and the original content can be recovered from the directly decrypted image. Furthermore, the data embedding operations of the lossless and the reversible schemes can be simultaneously performed in an encrypted image. With the combined technique, a receiver may extract a part of embedded data before decryption, and extract another part of embedded data and recover the original plaintext image after decryption.

2.EXISTING SYSTEM

2.1 LOSS LESS DATA HIDING SCHEME

In this section, a lossless data hiding scheme for public-key-encrypted images is proposed. There are three parties in the scheme: an image provider, a data-hider, and a receiver. With a cryptosystem possessing probabilistic property, the image provider encrypts each pixel of the original plaintext image using the public key of the receiver, and a data-hider who does not know the original image can modify the ciphertext pixel-values to embed some additional data into the encrypted image by multi-layer wet paper coding under a condition that the decrypted values of new and original cipher-text pixel values must be same. When having the encrypted image containing the additional data, a receiver knowing the data hiding key may extract the embedded data, while a receiver with the private key of the cryptosystem may perform decryption to retrieve the original plaintext image. In other words, the embedded data can be extracted in the encrypted domain, and cannot be extracted after decryption since the decrypted image would be same as the original plaintext image due to the probabilistic property. That also means the data embedding does not affect the decryption of the plaintext image. The sketch of lossless data hiding scheme is shown in Figure 1.

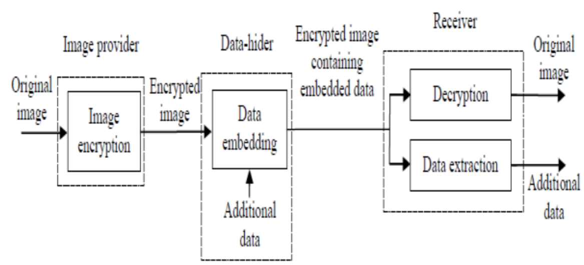


Figure 1. Sketch of lossless data hiding scheme for public-key-encrypted images

2.1.1. Image encryption

In this phase, the image provider encrypts a plaintext image using the public key of probabilistic cryptosystem p_k. For each pixel value m(i, j) where (i, j) indicates the pixel position, the image provider calculates its ciphertext value,

$$c(i,j) = E\left[p_k, m(i,j), r(i,j)\right] \quad (1)$$

where E is the encryption operation and r(i, j) is a random value. Then, the image provider collects the ciphertext values of all pixels to form an encrypted image.

Actually, the proposed scheme is capitable with various probabilistic public-key cryptosystems, such as Paillier [18] and Damgard-Jurik cryptosystems [25]. With Paillier cryptosystem [18], for two large primes p and q, calculate n = p·q, λ = lcm (p−1, q−1), where lcm means the least common multiple. Here, it should meet that gcd (n, (p−1)·(q−1)) = 1, where gcd means the greatest common divisor. The public key is composed of n and a randomly selected integer g in $Z^*_{n^2}$, while the private key is composed of λ and

$$\mu = \left(L\left(g^{\lambda} \bmod n^2\right)\right)^{-1} \bmod n \quad (2)$$

where

$$L(x) = \frac{(x-1)}{n} \quad (3)$$

In this case, (1) implies

$$c(i,j) = g^{m(i,j)} \cdot (r(i,j))^n \bmod n^2 \quad (4)$$

where r(i, j) is a random integer in Z^*_n. The plaintext pixel value can be obtained using the private key,

$$m(i,j) = L\left((c(i,j))^{\lambda} \bmod n^2\right) \cdot \mu \bmod n \quad (5)$$

As a generalization of Paillier cryptosystem, Damgard-Jurik cryptosystem [25] can be also used to encrypt the plaintext image. Here, the public key is composed of n and an element g in $Z^* n^{s+1}$ such that g = $(1+n)^j$.x mod n^{s+1} for a known j relatively prime to n and x belongs to a group isomorphic to Z^*_n, and we may choose d as the private key when meeting d mod n ∈ Z^*_n and d = 0 mod λ. Then, the encryption in (1) can be rewritten as

$$c(i,j) = g^{m(i,j)} \cdot (r(i,j))^{n^s} \bmod n^{s+1} \quad (6)$$

where r(i, j) is a random integer in $Z^* n^{s+1}$. By applying a recursive version of Paillier decryption, the plaintext value can be obtained from the ciphertext value using the private key. Note that, because of the probabilistic property of the two cryptosystems, the same gray values at different positions may correspond to different ciphertext values.

2.1.2 Data embedding

When having the encrypted image, the data-hider may embed some additional data into it in a lossless manner. The pixels in the encrypted image are reorganized as a sequence according to the data hiding key. For each encrypted pixel, the data-hider selects a random integer r'(i, j) in Z^*_n and calculates

$$c'(i,j) = c(i,j) \cdot (r'(i,j))^n \bmod n^2 \quad (7)$$

if Paillier cryptosystem is used for image encryption, while the data-hider selects a random integer r'(i, j) in $Z^* n^{s+1}$ and calculates

$$c'(i,j) = c(i,j) \cdot (r'(i,j))^{n^s} \bmod n^{s+1} \quad (8)$$

if Damgard-Jurik cryptosystem is used for image encryption. We denote the binary representations of c(i, j) and c'(i, j) as $b_k(i, j)$ and $b'_k(i, j)$, respectively,

$$b_k(i,j) = \left\lfloor \frac{c(i,j)}{2^{k-1}} \right\rfloor \bmod 2, \quad k = 1, 2, \ldots \quad (9)$$

$$b'_k(i,j) = \left\lfloor \frac{c'(i,j)}{2^{k-1}} \right\rfloor \bmod 2, \quad k = 1, 2, \ldots \quad (10)$$

Clearly, the probability of $b_k(i, j) = b'_k(i, j)$ (k = 1, 2, …) is 1/2. We also define the sets

$$S_1 = \{(i,j) \mid b_1(i,j) \neq b'_1(i,j)\}$$
$$S_2 = \{(i,j) \mid b_2(i,j) \neq b'_2(i,j), b_1(i,j) = b'_1(i,j)\}$$
$$\ldots$$
$$S_K = \{(i,j) \mid b_K(i,j) \neq b'_K(i,j), b_k(i,j) = b'_k(i,j), k = 1,2,\ldots,K-1\} \quad (11)$$

By viewing the k-th LSB of encrypted pixels as a wet paper channel (WPC) [26] and the k-th LSB in S_k as "dry" elements of the wet paper channel, the data-hider may employ the wet paper coding [26] to embed the additional data by replacing a part of c(i, j) with c'(i, j). The details will be given in the following.

Considering the first LSB, if c(i, j) are replaced with c'(i, j), the first LSB in S_1 would be flipped and the rest first LSB would be unchanged. So, the first LSB of the encrypted pixels can be regarded as a WPC, which includes changeable (dry) elements and unchangeable (wet) elements. In other words, the first LSB in S_1 are dry elements and the rest first LSB are wet positions. By using the wet paper coding [26], one can represent on average N_d bits by only flipping a part of dry elements where N_d is the number of dry elements. In this scenario, the data-hider may flip the dry elements by replacing c(i, j) with c'(i, j). Denoting the number of pixels in the image as N, the data-hider may embed on average N/2 bits in the first LSB-layer using wet paper coding. Considering the second LSB (SLSB) layer, we call the SLSB in S_2 as dry elements and the rest SLSB as wet elements. Note that the first LSB of ciphertext pixels in S_1 have been determined by replacing c(i, j) with c'(i, j) or keeping c(i, j) unchanged in the first LSB-layer embedding, meaning that the SLSB in S_1 are unchangeable in the second layer. Then, the data-hider may flip a part of SLSB in S_2 by replacing c(i, j) with c'(i, j) to embed on average N/4 bits using wet paper coding.

Similarly, in the k-th LSB layer, the data-hider may flip a part of k-th LSB in S_k to embed on average $N/2^k$ bits. When the data embedding is implemented in K layers, the total $N\cdot(1-1/2^k)$ bits, on average, are embedded. That implies the embedding rate, a ratio between the number of embedded bits and the number of pixels in cover image, is approximately $(1-1/2^k)$. That implies the upper bound of the embedding rate is 1 bit per pixel. The next subsection will show that, although a part of c(i, j) is replaced with c'(i, j), the original plaintext image can still be obtained by decryption.

2.1.3 Data extraction and image decryption

After receiving an encrypted image containing the additional data, if the receiver knows the data-hiding key, he may calculate the k-th LSB of encrypted pixels, and then extract the embedded data from the K LSB-layers using wet paper coding. On the other hand, if the receiver knows the private key of the used cryptosystem, he may perform decryption to obtain the original plaintext image. When Paillier cryptosystem is used, Equation (4) implies

$$c(i,j)=g^{m(i,j)}\cdot\left(r(i,j)\right)^{n}+\alpha\cdot n^{2} \qquad (12)$$

where α is an integer. By substituting (12) into (7), there is

$$c'(i,j)=g^{m(i,j)}\cdot\left(r(i,j)\cdot r'(i,j)\right)^{n} \bmod n^{2} \quad (13)$$

Since r(i, j)·r'(i, j) can be viewed as another random integer in Z^*_n, the decryption on c'(i, j) will result in the plaintext value,

$$m(i,j)=L\left(\left(c'(i,j)\right)^{\lambda} \bmod n^{2}\right)\cdot\mu \bmod n \quad (14)$$

Similarly, when Damgard-Jurik cryptosystem is used,

$$c'(i,j)=g^{m(i,j)}\cdot\left(r(i,j)\cdot r'(i,j)\right)^{n^{s}} \bmod n^{s+1} \quad (15)$$

The decryption on c'(i, j) will also result in the plaintext value. In other words, the replacement of ciphertext pixel values for data embedding does not affect the decryption result.

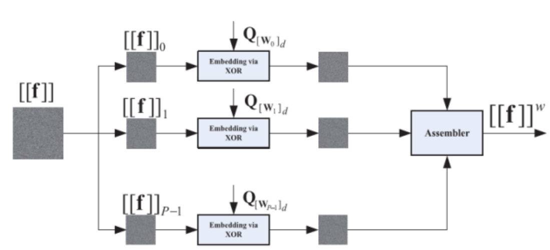
3. Proposed system

Instead of considering dedicated encryption algorithms tailored to the scenario of encrypted-domain data hiding, we here stick to the conventional stream cipher applied in the standard format. That is, the cipher text is generated by bitwise XORing the plaintext with the key stream. If not otherwise specified, the widely used stream cipher AES in the CTR mode (AES-CTR) is assumed. The resulting data hiding paradigm over encrypted domain could be more practically useful because of two reasons:

1. Stream cipher used in the standard format (e.g.,AES-CTR) is still one of the most popular and reliable encryption tools, due to its provable security and high software/hardware implementation efficiency. It may not be easy, or even infeasible, to persuade customers to adopt new encryption algorithms that have not been thoroughly evaluated.
2. Large amounts of data have already been encrypted using stream cipher in a standard way.

Hence, due the implementation of the AES-CTR algorithm it can be told the RIDH technique takes place over an encrypted domain.

Encryption Block Diagram:



3.1 Input Image Initialization:

In this module, we initialize the given image (i.e.) get the input image from user by using the keyword 'uigetfile'. This contains only the pathname and filename. To read the image filename, we used 'imread' command.

3.2 Image Encryption:

Assume the original image with a size of N1XN2 is in uncompressed format and each pixel with gray value falling into [0, 255] is represented by 8 bits. Denote the bits of a pixel as $b_{i,j,0},b_{i,j,1},\ldots,b_{i,j,7}$ where 1<=i<=N1 and1<=j<=N2, the gray value as, and the number of pixels as N(N=N1XN2).That implies In encryption phase, the exclusive-or results of the original bits and pseudo- random bits are calculated. When stream cipher is employed, the encrypted image is generated by

$$[[\mathrm{f}]] = \mathrm{Enc}(\mathrm{f}, K) = \mathrm{f} \oplus \mathrm{K}$$

Where **f** and [[**f**]] denote the original and the encrypted images, respectively. Here, **K** denotes the key stream generated using the secret encryption key *K*

3.3 Key Modulation:

the key management functions Instead of considering dedicated encryption algorithms tailored to the scenario of encrypted-domain data hiding, we here stick to the conventional stream cipher applied in the standard format. That is, the cipher text is generated by bitwise XOR the plaintext with the key stream. Find the public key $Q[W_i]_d$ associated with W_i, where the index $[W_i]_d$ is the decimal representation of W_i For instance, when n = 3 and W_i = 010, the corresponding public key is Q2. Embed the length-*n* message bits $W_i$into the *i* th block via

$$[[\mathbf{f}]]_i^w = [[\mathbf{f}]]_i \oplus \mathbf{Q}_{[\mathbf{W}_i]_d}$$

Decryption Block Diagram:

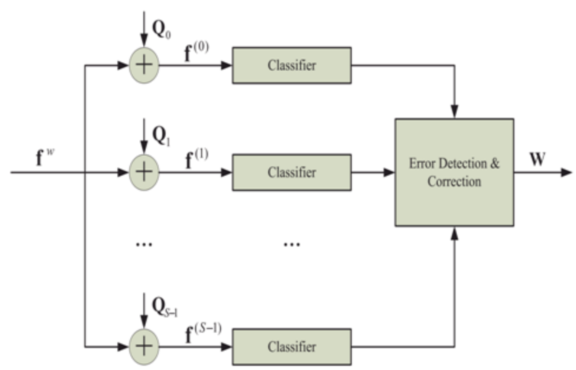


Fig5.2: Schematic of data extraction

3.4 Data Extraction and Image Recovery:

The decoder in the data center has the decryption key *K* and attempts to recover both the embedded message and theoriginal image simultaneously from [[**f**]]*w*, which is assumedto be perfectly received without any distortions. Note that thisassumption is made in almost all the existing RIDH methods.Due to the

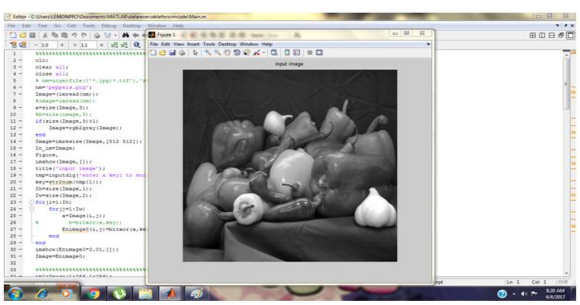
ISSN: 2278-4632
Vol-11 Issue-01 2021

interchangeable property of XOR operations, theany attacker without the data-hiding 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. decoder first XORs [[**f**]]*w* with the encryption key stream **K** and obtains

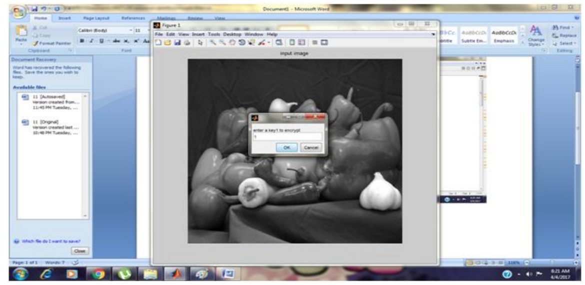
$$\mathbf{f}^{w} = [[\mathbf{f}]]^{w} \oplus \mathbf{K}$$

The resulting **f***w*is then partitioned into a series of non overlapping Blocks fw*i*'s of size $M \times N$, similar to the operation conducted at the embedding stage.

Input Image:

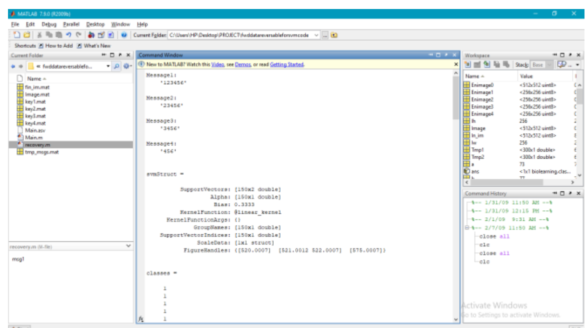


Enter the Key For Encryption:

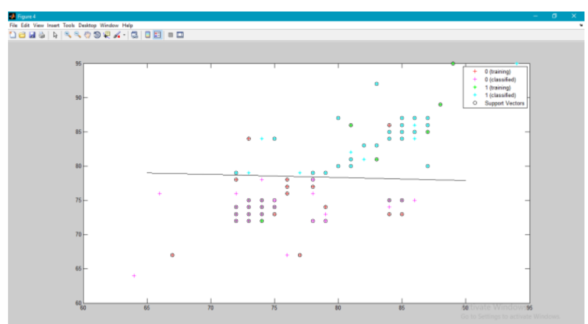


FINAL EMBEDEED IMAGE

DATA RETRIVAL



SVM CLASSIFICATION



CONCLUSION AND FUTURE SCOPE

8.1 Conclusion:

ISSN: 2278-4632
Vol-11 Issue-01 2021

In this paper, we design a secure RIDH scheme operated over the encrypted domain. We suggest a public key modulation mechanism, which allows us to embed the data via simple XOR operations, without the need of accessing the secret encryption key. At the decoder side, we propose to use a powerful two-class SVM classifier to discriminate encrypted and non-encrypted image patches, enabling us to jointly decode the embedded message and the original image signal perfectly. We have also performed extensive experiments to validate the superior embedding performance of our proposed RIDH method over encrypted domain.

8.2 Future Scope:

- Micro electronics intends to use this work as part of larger projects such as smart metering in power systems and in data communication.

REFERENCES

[1] N. A. Saleh, H. N. Boghdad, S. I. Shaheen, A. M. Darwish, "High Capacity Lossless Data Embedding Technique for Palette Images Based on Histogram Analysis," Digital Signal Processing, 20, pp. 1629−1636, 2010.

[2] J. Tian, "Reversible Data Embedding Using a Difference Expansion," IEEE Trans. on Circuits and Systems for Video Technology, 13(8), pp. 890−896, 2003.

[3] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible Data Hiding," IEEE Trans. on Circuits and Systems for Video Technology, 16(3), pp. 354−362, 2006.

[4] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless Generalized-LSB Data Embedding," IEEE Trans. on Image Processing, 14(2), pp. 253–266, 2005.

[5] X. Hu, W. Zhang, X. Li, and N. Yu, "Minimum Rate Prediction and Optimized Histograms Modification for Reversible Data Hiding," IEEE Trans. on Information Forensics and Security, 10(3), pp. 653-664, 2015.

[6] X. Zhang, "Reversible Data Hiding with Optimal Value Transfer," IEEE Trans. on Multimedia, 15(2), 316−325, 2013.

[7] W. Zhang, X. Hu, X. Li, and N. Yu, "Optimal Transition Probability of Reversible Data Hiding for General Distortion Metrics and Its Applications," IEEE Trans. on Image Processing, 24(1), pp. 294-304, 2015.

[8] S. Lian, Z. Liu, Z. Ren, and H. Wang, "Commutative Encryption and Watermarking in Video Compression," IEEE Trans. on Circuits and Systems for Video Technology, 17(6), pp. 774−778, 2007.

[9] M. Cancellaro, F. Battisti, M. Carli, G. Boato, F. G. B. Natale, and A. Neri, "A Commutative Digital Image Watermarking and Encryption Method in the Tree Structured Haar Transform Domain," Signal Processing: Image Communication, 26(1), pp. 1−12, 2011.

[10] X. Zhang, "Commutative Reversible Data Hiding and Encryption," Security and Communication Networks, 6, pp. 1396−1403, 2013.

[11] X. Zhang, "Reversible Data Hiding in Encrypted Image," IEEE Signal Processing Letters, 18(4), pp. 255−258, 2011.

[12] W. Hong, T.-S. Chen, and H.-Y. Wu, "An Improved Reversible Data Hiding in Encrypted Images Using Side Match," IEEE Signal Processing Letters, 19(4), pp. 199−202, 2012.

[13] J. Yu, G. Zhu, X. Li, and J. Yang, "An Improved Algorithm for Reversible Data Hiding in Encrypted Image," Proceeding of the 11th International Workshop on Digital-Forensics Watermark (IWDW 2012), Shanghai, China, Oct. 31-Nov. 02, 2012, Lecture Notes in Computer Science, 7809, pp. 358-367, 2013.

[14] W. Puech, M. Chaumont, and O. Strauss, "A Reversible Data Hiding Method for Encrypted Images," Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, Proc. SPIE, 6819, 2008.

[15] X. Zhang, "Separable Reversible Data Hiding in Encrypted Image," IEEE Trans. Information Forensics & Security, 7(2), pp. 526−532, 2012.

[16] Z. Qian, X. Zhang, and S. Wang, "Reversible Data Hiding in Encrypted JPEG Bitstream," IEEE Trans. on Multimedia, 16(5), pp. 1486−1491, 2014.

[17] M. S. A. Karim, and K. Wong, "Universal Data Embedding in Encrypted Domain," Signal Processing, 94, pp. 174-182, 2014.

[18] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible Data Hiding in Encrypted Images by Reserving Room Before Encryption," IEEE Trans. Information Forensics & Security, 8(3), pp. 553-562, 2013.

[19] W. Zhang, K. Ma, and N. Yu, "Reversibility Improved Data Hiding in Encrypted Images," Signal Processing, 94, pp. 118-127, 2014.

[20] Y.-C. Chen, C.-W. Shiu, and G. Horng, "Encrypted Signal-Based Reversible Data Hiding with Public Key Cryptosystem," Journal of Visual Communication and Image Representation, 25, pp. 1164-1170, 2014.

[21] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," Proceeding of the Advances Cryptology, EUROCRYPT'99, LNCS, 1592, pp. 223-238, 1999.

[22] T. Bianchi, A. Piva, and M. Barni, "On the Implementation of the Discrete Fourier Transform in the Encrypted Domain," IEEE Trans. Information Forensics and Security, 4(1), pp. 86–97, 2009.

[23] T. Bianchi, A. Piva, and M. Barni, "Composite Signal Representation for Fast and Storage-Efficient Processing of Encrypted Signals," IEEE Trans. Information Forensics and Security, 5(1), pp. 180–187, 2010.

[24] P. Zheng, and J. Huang, "Discrete Wavelet Transform and Data Expansion Reduction in Homomorphic Encrypted Domain," IEEE Trans. Image Processing, 22(6), pp. 2455-2468, 2013.

[25] I. Damgård, and M. Jurik, "A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System," Public Key Cryptography, pp. 119-136, 2001.

[26] J. Fridrich, M. Goljan, P. Lisonek, and D. Soukal, "Writing on Wet Paper," IEEE Trans. Signal Processing, 53(10), pp. 3923-3935, 2005.