International Journal of Research
Available at https://edupediapublications.org/journals

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 09
August 2017

Improved PSNR with Encryption and Data hiding Using Histogram shrink and Wet paper coding

M.Hemalatha[1], Mr.K.Rajasekhar[2]

Research Associate, Dept. of Electronics and Communication Engineering, UCEK (A), JNTUK, Kakinada,

Andhra Pradesh, India[1]

Assistant Professor, Dept. of Electronics and Communication Engineering, UCEK (A), JNTUK, Kakinada,

Andhra Pradesh, India[2]

***Abstract*:**

Now a day's security is the main object in a digitalized world. To make our data secure we need encryption which adds key and some randomly generated values to the respected image that means Encryption disguises the content of the message. Whereas steganography disguise the existence of the message. However additional security will be obtained if we use encryption along with the steganography. Encryption algorithm which we used here is a modificated version of RSA algorithm which is proposed by paillier and the random values are added by PN sequence generator values to the respected pixels. The datahiding proposed here is, we need to split the image into different layers and add the data which we want to hide. The method which we proposed here is merging of lossless and reversible data hiding, whereas the key used here is symmetric and the data hiding at the transitter uses the reversible and at the receiver the lossless method is used.

***Keywords*:**

Encryption, Datahiding,PSNR, Paillier system.

1. INTRODUCTION

In our daily life internet is the source of transfer of data. Which makes our busy life easy. The internet and multimedia make digitalized life simple but due to unauthorized access or hacking we may face some problems. so we need secure data transfer ,Which is possible with encryption and datahiding .If we want to send a data and an image at a time without wasting another channel we can easily send the message along with the image .

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. Encryption Helps you Move to the Cloud Everyone is concerned about moving sensitive data to the cloud, and many organizations perceive that the cloud is not as safe as their own data center. If your data is present in the cloud storage, it's not only possible that strangers might see it, but your data could be sitting on the same storage as your competitors.

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. 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 the cover data is used for carrying extra data and the remaining data are encrypted for privacy protection.

Alternatively, the additional data are embedded into a data space that is invariable to encryption operations. 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 named as reversible data hiding in encrypted images. 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. when medical images are 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.

This present 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. In these schemes, the pixel division or reorganization is avoided and the encryption or decryption is performed on the cover pixels directly, so that the amount of the encrypted data and computational complexity are lowered.

With 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 and the embedded data can be going to 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.PREVIOUS WORK

Reversible data hiding scheme for encrypted image: This work proposes a reversible data hiding scheme for encrypted image. After completion of encrypting the entire data of an uncompressed image by a stream cipher, the additional data can be embedded into the image by modifying a small proportion of the encrypted data. With the encrypted image containing additional data, one may be firstly decrypt it using the encryption key, and the decrypted version is similar to original image. In accordance with the data-hiding key, with the aid of spatial correlation in natural image, the embedded data can be successfully extracted and the original image can also be perfectly recovered.

Reversible data hiding is the technique to add additional message into some distortion-unacceptable cover media, such as military and medical images, with in a reversible manner so that the original cover content can be perfectly restored after the extraction of the hidden message. In the encryption phase, the exclusive-or operation of the original bits and pseudo-random bits are calculated as,

$$B_{i,j,k} = b_{i,j,k} \oplus r_{i,j,k}$$

$$b_{i,j,k} = \left[\frac{p_{i,j}}{2^k} \right] \mathrm{mod}\, 2, k = 0,1,........7$$

$$p_{i,j} = \sum_{u=0}^{7} b_{i,j,k} \bullet 2^k$$

In decryption the operation is reverse ex-or is

$$b^1_{i,j,k} = r_{i,j,k} \oplus B^1_{i,j,k}$$
$$= r_{i,j,k} \oplus \overline{B}_{i,j,k}$$
$$= r_{i,j,k} \oplus \overline{b}_{i,j,k} \overline{\oplus} \overline{r}_{i,j,k}$$
$$= \overline{b}_{i,j,k},$$
$$K=0,1,2.$$

The mean square error is

$$E_A = \frac{1}{8} \sum_{u=0}^{7} \left[u - (7-u) \right]^2 = 21$$

The PSNR is

$$PSNR = 10 \cdot \log_{10} \frac{255^2}{\frac{E_A}{2}} = 37.9dB.$$

The PSNR of this method is low and Zhang's work did not fulfill the pixels in calculating the smoothness of the each block and was not consider the pixel correlations in the border of the neighboring blocks. These two issues can reduces the correctness of data extraction

Separable Reversible Data Hiding in Encrypted Images: This work proposes a novel scheme for separable reversible data hiding in encrypted images. In this first phase of operation, a content owner encrypts the original uncompressed image using an encryption key. Then, a data-hider may compress the least significant bits of the encrypted image using a data-hiding key to create a space to accommodate some of the additional data. With an encrypted image containing additional data, if a receiver has the data-hiding key, he can able to extract the additional data though he does not know the content of the image content, which makes the system more secure than the previous methods.

If the receiver has the encryption key, he can able to decrypt the received data to obtain the image similar to the original image, but he cannot extract the additional data.

If the receiver has both data-hiding key and encryption key, Disadvantages: He can able to extract the additional data and recover the original content of input without any error by exploiting the spatial correlation in natural image when the amount of additional data is not too large.

3. LOSSLESS DATA HIDING SCHEME

In this section, the lossless data hiding scheme for public-key-encrypted images is proposed. There are three stage in the scheme they are an image provider, a data-hider, and a receiver. With a cryptosystem possessing the probabilistic property, the image provider can encrypts each pixel of the original plaintext image using the public key of the receiver, and the data-hider who does not know the original image.

The data-hider 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.

The receiver with the private key of the cryptosystem may perform decryption to retrieve the original plaintext image. That means the embedded data can be extracted in an encrypted domain, and cannot be extracted after the decryption since the decrypted image would be as same as the original plaintext image due to probabilistic property. That means the data embedding does not affect the decryption of the plaintext image. The process of lossless data hiding scheme is shown in below Figure.

4. REVERSIBLE DATA HIDING SCHEME

This proposes a reversible data hiding scheme for public-key-encrypted images. In the reversible scheme, a preprocessing is employed to shrink the image histogram, and then each pixel is encrypted with additive homomorphic cryptosystem by the image provider. When having the encrypted image, the data-hider modifies the ciphertext pixel values to embed a bit-sequence generated from the additional data and error-correction codes.

Due this homomorphic property, the modification in the encrypted domain will result in slight increase or decrease on plaintext pixel values, implying that a decryption can be implemented to obtain an image similar to the original plaintext image on receiver side. Because of the histogram shrink before the encryption, the data embedding operation does not cause any overflow or the underflow in the directly decrypted image.

Then, the original plaintext image can also be recovered and the embedded additional data can also be extracted from directly decrypted image.

The data-extraction and the content-recovery of the reversible scheme are performed in plaintext domain, while the data extraction of the previous lossless scheme is performed in the encrypted domain and the content recovery is need less. The process of reversible data hiding scheme is given in Figure. the lossless and reversible schemes are as shown in figure1 and 2

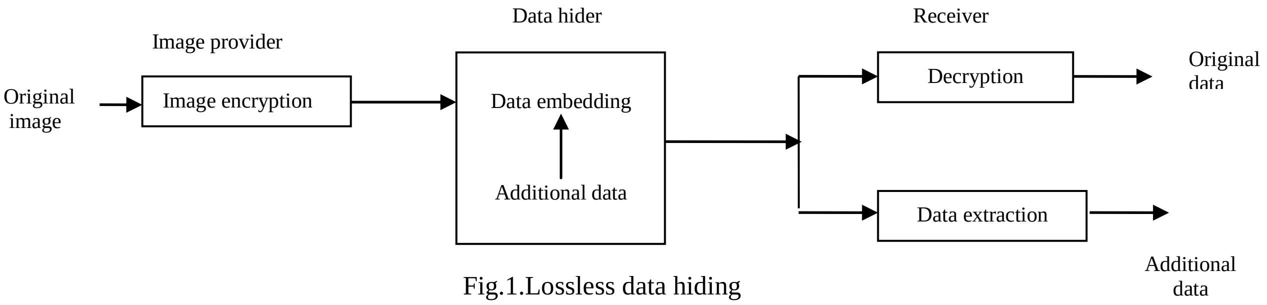

Fig.1.Lossless data hiding

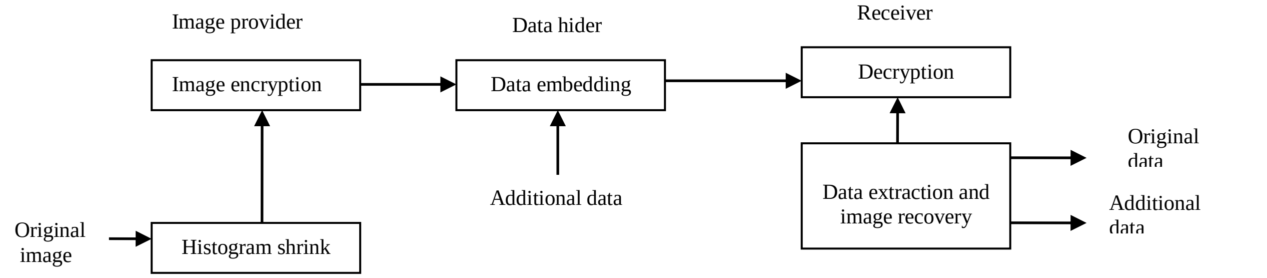

Fig.2.Reversible Data hiding

5. PROPOSED COMBINED DATA HIDING

Histogram shrink and image encryption:

In the reversible scheme, a small integer δ shared by the image provider, the data-hider and the receiver will be used, and its value will be discussed later. Denote the number of pixels in the original plaintext image with gray value *v* as h_v, implying

$$\sum_{v=0}^{255} h_v = N$$

where *N* is the number of all pixels in the image. The image provider collects the pixels with gray values in [0, δ+1], and represent their values as a binary stream BS1. When an efficient lossless source coding is used, the length of BS1.

$$l_1 \approx \sum_{v=0}^{\delta+1} h_v \cdot H\left(\frac{h_0}{\sum_{v=0}^{\delta+1} h_v}, \frac{h_1}{\sum_{v=0}^{\delta+1} h_v}, \ldots\ldots\ldots\ldots \frac{h_{\delta+1}}{\sum_{v=0}^{\delta+1} h_v} \right)$$

Where *H*(·) is the entropy function. The image provider also collects the pixels with gray values in [255−δ, 255], and represent their values as a binary stream BS2 with a length *l*2. Similarly.

$$l_2 \approx \sum_{v=255-\delta}^{255} h_v \cdot H\left(\frac{h_{255-\delta}}{\sum_{v=255-\delta}^{255} h_v}, \frac{h_{255-\delta+1}}{\sum_{v=255-\delta}^{255} h_v}, \ldots\ldots \frac{h_{255}}{\sum_{v=255-\delta}^{255} h_v} \right)$$

Then, the gray values of all pixels are enforced into [δ+1, 255−δ],as shown

$$m_S(i,j) = \begin{cases} 255-\delta, if m(i,j) \geq 255-\delta \\ m(i,j), if \delta+1 < m(i,j) < 255-\delta \\ \delta+1, if m(i,j) \leq \delta+1 \end{cases}$$

Denoting the new histogram as *h'*$_v$, as below

$$h'_v = \begin{cases} 0, v \leq \delta \\ \sum_{v=0}^{\delta+1} h_v, v = \delta+1 \\ h_v, \delta+1 < v < 255-\delta \\ \sum_{v=255-\delta}^{255} h_v, v = 255-\delta \\ 0, v > 255-\delta \end{cases}$$

Then the image provider may finds out the peak of the new histogram by the below equation,

$$V = \arg\max_{\delta+1 \leq v \leq 255-\delta} h'_v$$

The image provider also divides all pixels into two sets: the first set including (*N*−8) pixels and the second set including the rest 8 pixels, and maps each bit of BS1, BS2 and the LSB of pixels in the second set to a pixel in the first set with gray value *V*. Since the gray values close to extreme black/white are rare, there is

$$h'_V \geq l_1 + l_2 + 16$$

when δ is not too large. In this case, the mapping operation is feasible. Here, 8 pixels in the second set cannot be used to carry BS1/BS2 since their LSB should be used to carry the value of *V*, while 8 pixels in the first set cannot be used to carry BS1/BS2 since their LSB should be used to carry the original LSB of the second set. So, a total of 16 pixels cannot be used for carrying BS1/BS2. That is the reason that there is a value 16 in (22). The experimental result on 1000 natural images shows (22) is always right when δ is less than 15. So, we recommend the parameter δ < 15. Then, a histogram shift operation is made,

$$m_T(i,j) = \begin{cases} m_S(i,j), \\ if m_s(i,j) > V \\ V, if m_S(i,j) = V \\ and the corresponding bit is 0 \\ V-1, if m_s \\ (i,j) = V and the corresponding bit is 1 \\ m_S(i,j)-1, \\ if m_s(i,j) < V \end{cases}$$

In the other words, BS1, BS2 and the LSB of pixels in second set are carried by the pixels in first set. After this, the image provider represents the value of *V* as 8 bits and maps them to the pixels in the second set in a one-to-one manner. Then, the values of pixels in the second set are modified as follows,

$$m_T(i,j) = \begin{cases} m_S(i,j), \\ if LSB of m_S(i,j) is same as corres bit \\ m_S(i,j)-1, \\ if LSB of m_S(i,j) differs from corres bit \end{cases}$$

That means that the value of *V* is embedded into LSB of the second set. This way, all the pixel values must fall into [δ, 255−δ]. At last, the image provider will encrypts all the pixels using a public key cryptosystem with the additive homomorphic property, such as the Paillier and the Damgard-Jurik cryptosystems. When Paillier cryptosystem is used, the ciphertext Pixel is as shown by the below equation.

$$c(i,j) = g^{m_T(i,j)}.(r(i,j))^n.\mathrm{mod}\, n^2$$

the decrypted pixel values in Set B are $m_T(i,j)$ since their ciphertext values are unchanged in the data embedding phase. When δ is small, the decrypted image is perceptually as similar as to the original plaintext image. Then, the receiver with data-hiding key can extract the embedded data from directly decrypted image. He can able to estimates the pixel values in Set A using their neighbors as average

$$\overline{m_T}(i,j) = \frac{m_T(i-1,j)+m_T(i,j-1)+m_T(i+1,j)+m_T(i,j+1)}{4}$$

and can obtain an estimated version of the coded bit-sequence by comparing the decrypted and estimated pixel values in Set A. That means that the coded bit is estimated as 0 if

$\overline{m_T}(i,j) > m'(i,j)$or as 1 if $\overline{m_T}(i,j) \le m'(i,j)$.

With the estimate of the coded bit-sequence, then the receiver may employ error-correction method to retrieve the original coded bit-sequence and the embedded additional data. with a larger δ, the error rate in the estimate of coded bits would be lower, so that more additional data can be embedded when ensuring successful error correction and data extraction. That means, a smaller δ would result in a higher error rate in the estimate of coded bits, so that the error correction may be un successful when excessive of payload is embedded. That means the embedding capacity of reversible data hiding scheme is dependent on the value of the δ. After retrieving original coded bit-sequence and the embedded additional data, the original plaintext image may be recovered. For the pixels in Set A, $m_T(i,j)$ are retrieved according to the coded bit-sequence as

$$m_T(i,j) = \begin{cases} m'(i,j) - \delta, \\ ifthecorrespondingbitis1 \\ m'(i,j) + \delta, \\ ifthecorrespondingbitis0 \end{cases}$$

For the pixels in Set B, as mentioned above, $m_T(i,j)$ are just $m'(i,j)$. Then, divides all $m_T(i,j)$ into two sets: the first one including (*N*−8) pixels and the second one including the rest 8 pixels. The receiver may obtain the value of *V* from the LSB in the second set, and retrieve $m_S(i,j)$ of the first set

$$m_S(i,j) = \begin{cases} m_T(i,j), if m_T(i,j) > V \\ V, if m_T(i,j) = V or V-1 \\ m_T(i,j)+1, if m_T(i,j) < V-1 \end{cases}$$

Meanwhile, the receiver extracts a bit 0 from a pixel with $m_T(i,j)$ = *V* and a bit 1 from a pixel with $m_T(i,j)$ = *V*−1. After decomposing the extracted data into BS1, BS2 and the LSB of $m_S(i,j)$ in the second set, the receiver retrieves $m_S(i,j)$ of the second set,

$$m_S(i,j) = \begin{cases} m_T(i,j), \\ ifLSBofm_S(i,j)andm_T(i,j)aresame \\ m_T(i,j)+1, \\ ifLSBofm_S(i,j)andm_T(i,j)aredifferent \end{cases}$$

Collect all pixels with $m_S(i,j)$ = δ+1, and, according to BS1, recover their original values within [0, δ+1]. Similarly, the original values of pixels with $m_S(i,j)$ = 255−δ are recovered within [255−δ, 255] according to BS2. This way, the original plaintext image is recovered.

In above sections, a lossless and a reversible data hiding schemes for public-key-encrypted images are proposed. In both of the two schemes, the data embedding operations are performed in encrypted domain. The data extraction procedure of the two schemes are entirely different. With the help of the lossless scheme, data embedding does not affect the plaintext content and data extraction is also be performed in the encrypted domain.

With the help of an reversible scheme, there is a slight distortion in directly decrypted image caused by the data embedding, and data extraction and image recovery must be performed in plaintext domain.

That implies, on receiver side, the additional data embedded by the lossless scheme cannot be extracted after the decryption, and the additional data embedded by the reversible scheme cannot extracted before decryption.

Later we combine the lossless and reversible schemes to construct the new scheme, in which the data extraction in either of the two domains is feasible. That means the additional data for various purposes may be embedded into the encrypted image, and the part of the additional data can also be extracted before the decryption and another part can be extracted after decryption.

The combined scheme process to be as, the image provider performs histogram shrink and image encryption. When we are having the encrypted image, the data-hider may embed the first part of additional data using the method described in the above. By denoting the ciphertext pixel values containing the first part of additional data as *c'*(*i*, *j*), the data-hider calculates

$$c''(i,j) = c'(i,j).(r''(i,j))^n \bmod n^2$$

Where *r''*(*i*, *j*) are randomly selected in *Z*n* or for the Paillier cryptosystem. Then, he may able to employ wet paper coding in several LSB-planes of ciphertext pixel values to embed the second part of the additional data by replacing a part of *c'*(*i*, *j*) with *c''*(*i*, *j*). That means the method described in lossless is used to embedded the second part of the additional data

At receiver side, the receiver firstly extracts the second part of additional data from the LSB-planes of encrypted domain. Then after the decryption with his private key, he extracts the first part of additional data and recovers the original plaintext image from the directly decrypted image as described in the reversible scheme.

The process of the combined scheme is shown in Figure Note that, since the reversibly embedded data should also be extracted in plaintext domain and the lossless embedding does not affect the decrypted result, the lossless embedding should be implemented after the reversible embedding in the combined scheme.

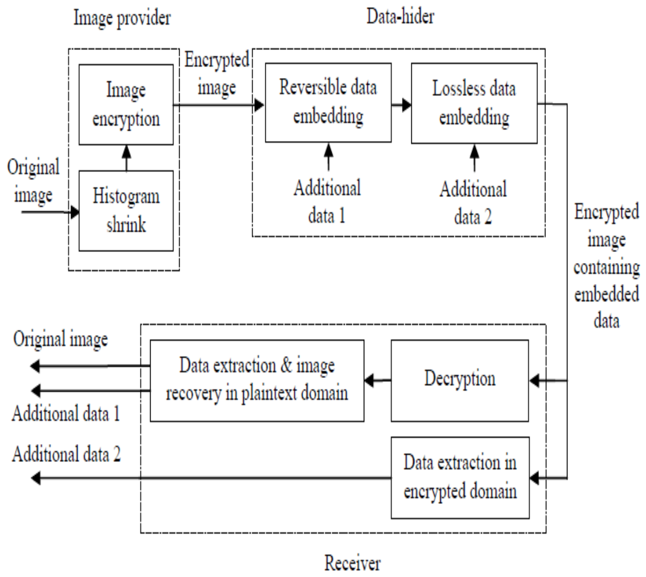

Fig.3. Combined data hiding scheme

6. EXPERIMENTAL RESULTS

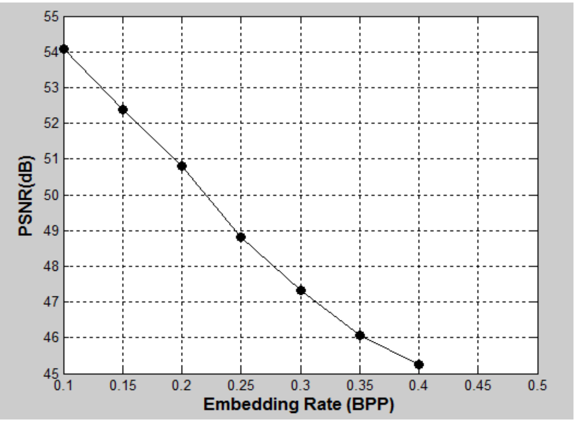

Fig.4.Experimental results

7. CONCLUSION

The combined scheme of lossless and reversible data hiding scheme improved a PSNR value and symmetric key provides the data hiding security. The PSNR with the asymmetric key is 43.03dB for leena image, but for symmetric key the PSNR is increased to 54 dB. But to get noiseless image we will use water marking

8. REFERENCES

[1] 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.

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

[3] 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. .

[4] 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.

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

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

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

[8] 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.

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

[10] Y.-C.Chen, C.-W. Shiu, and G. Horng, “Encrypted Signal-Based Reversible Data Hiding with PublicKey Cryptosystem,” *Journal of Visual Communication and Image Representation*, 25, pp. 1164-1170, 2014.

[11] P.Paillier, “Public-Key Cryptosystems Based on Composite Degree Residuosity Classes,” *Proceeding of the Advances Cryptology, EUROCRYPT’99, LNCS*, 1592, pp. 223-238, 1999.

[12] 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.

[13] 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.

[14] P.Zheng, and J.Huang, “Discrete Wavelet Transform and DataExpansion Reduction in Homomorphic Encrypted Domain,” *IEEE Trans. Image Processing*, 22(6), pp. 2455-2468, 2013.

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

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