A New Secure Data Hiding AES-CTR Key Modulation

L. Malliga[1], D. Lakshi Sri Kavya[2], G. Jahnavi[2], G. Vanajakshi[2], E. Rachana[2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of Electronics and Communication Engineering
[1,2]Malla Reddy Engineering College for Women, Maisammaguda, Hyderabad, Telangana, India

ABSTRACT: This paper proposes a novel reversible image data hiding scheme over encrypted domain. Data embedding is achieved through a public key modulation mechanism, in which access to the secret encryption key is not needed. At the decoder side, a powerful two-class SVM classifier is designed to distinguish encrypted and non encrypted image patches, allowing us to jointly decode the embedded message and the original image signal. Compared with the state-of-the-art methods, the proposed approach provides higher embedding capacity and is able to perfectly reconstruct the original image as well as the embedded message. Extensive experimental results are provided to validate the superior performance of our scheme.

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

As described in Sections 3 and 4, 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. On the other hand, the data extraction procedures of the two schemes are very different. With the lossless scheme, data embedding does not affect the plaintext content and data extraction is also performed in encrypted domain. With the reversible scheme, there is slight distortion in directly decrypted image caused by 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 decryption, while the additional data embedded by the reversible scheme cannot extracted before decryption. In this section, we combine the lossless and reversible schemes to construct a new scheme, in which data extraction in either of the two domains is feasible. That means the additional data for various purposes may be embedded into an encrypted image, and a part of the additional data can be extracted before decryption and another part can be extracted after decryption.

In the combined scheme, the image provider performs histogram shrink and image encryption as described in Subsection 3.A. When having the encrypted image, the data-hider may embed the first part of additional data using the method described in Subsection 3.B. Denoting the ciphertext pixel values containing the first part of additional data as c'(i, j), the data-hider calculates

$$c''(i,j)=c'(i,j)\cdot\left(r''(i,j)\right)^{n} \bmod n^{2} \qquad (36)$$

or

$$c''(i,j)=c'(i,j)\cdot\left(r''(i,j)\right)^{n^{s}} \bmod n^{s+1} \qquad (37)$$

where r''(i, j) are randomly selected in Z^*_n or Z^* n^{s+1} for Paillier and Damgard-Jurik cryptosystems, respectively. Then, he may employ wet paper coding in several LSB-planes of ciphertext pixel values to embed the second part of additional data by replacing a part of c'(i, j) with c''(i, j). In other words, the method described in Subsection 2.B is used to embed the second part of additional data. On receiver side, the receiver firstly extracts the second part of additional data from the LSB-planes of encrypted domain. Then, after 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 Subsection 3.C. The sketch of the combined scheme is shown in Figure 3. Note that, since the reversibly embedded data should be extracted in the plaintext domain and the lossless embedding does not affect the decrypted result, the lossless embedding should implemented after the reversible embedding in the combined scheme.

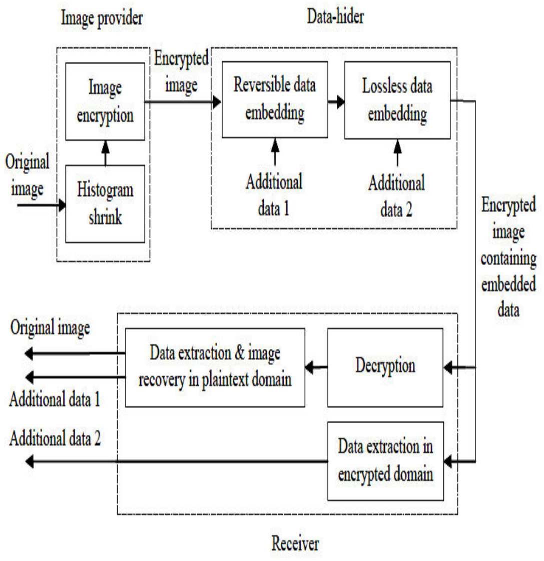


Figure 3. Sketch of combined scheme

Four gray images sized 512×512, Lena, Man, Plane and Crowd, shown in Figure 4, and 50 natural gray images sized 1920×2560, which contain landscape and people, were used as the original plaintext covers in the experiment. With the lossless scheme, all pixels in the cover images were firstly encrypted using Paillier cryptosystem, and then the additional data were embedded into the LSB-planes of ciphertext pixel-values using multi-layer wet paper coding as in Subsection 2.B. Table 1 lists the average value of embedding rates when K LSB-planes were used for carrying the additional data in the 54 encrypted images. In fact, the average embedding rate is very close to $(1-1/2^{k})$. On receiver side, the embedded data can be extracted from the encrypted domain. Also, the original plaintext images can be retrieved by direct decryption. In other word, when the decryption was performed on the encrypted images containing additional data, the original plaintext images were obtained.

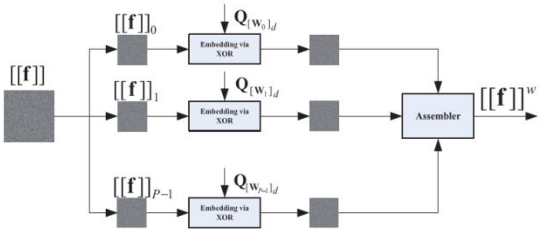
3. Proposed system

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 $b_{i,j,0}$,$b_{i,j,1}$,…,$b_{i,j,7}$ 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

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

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:

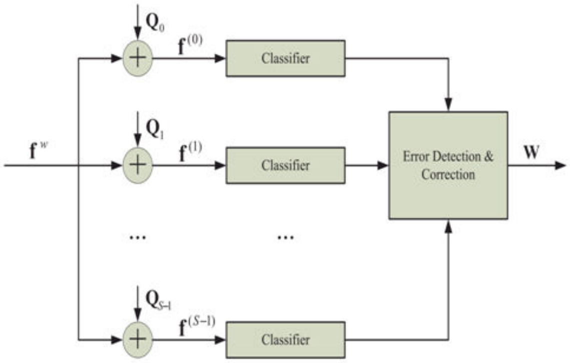


Fig5.2: Schematic of data extraction

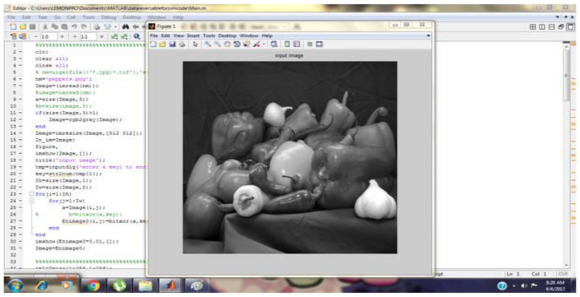
3.4 Data Extraction and Image Recovery:

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 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

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

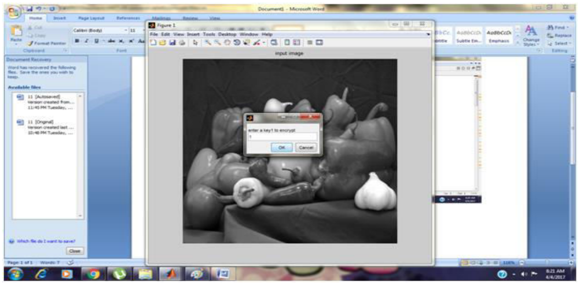
The resulting **f***w*is then partitioned into a series of non overlapping Blocks fw*i*'s of size *M* × *N*, similar to the operation conducted at the embedding stage.

Input Image:


Editor - C:\Users\LEMONPRO\Documents\MATLAB\datareversableforsvmcode\Main.m
Figure 1
input image

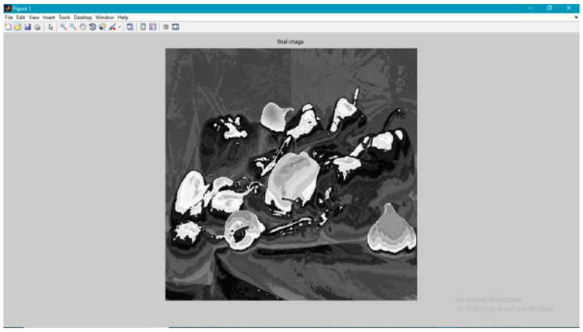
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc;
clear all;
close all;
% nm=uigetfile({'*.jpg;*.tif'},'
nm='peppers.png';
Image=(imread(nm));
%image=imread(nm);
a=size(Image,3);
%b=size(image,3);
if(size(Image,3)>1)
    Image=rgb2gray(Image);
end
Image=imresize(Image,[512 512]);
In_im=Image;
figure,
imshow(Image,[]);
title('input image');
tmp=inputdlg('enter a key1 to en
key=str2num(tmp{1});
Ih=size(Image,1);
Iw=size(Image,2);
for(i=1:Ih)
    for(j=1:Iw)
        a=Image(i,j);
%         b=bitxor(a,key);
        Enimage0(i,j)=bitxor(a,ke
    end
end
imshow(Enimage0*0.01,[]);
Image=Enimage0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Enter the Key For Encryption:

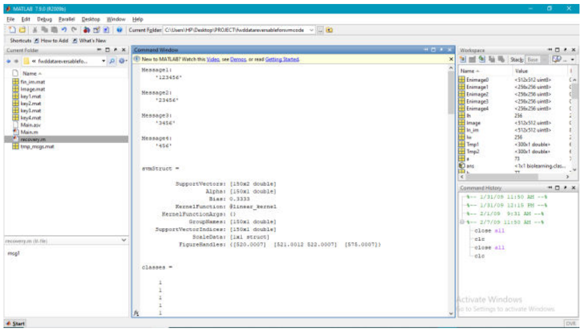


enter a key1 to encrypt

FINAL EMBEDEED IMAGE

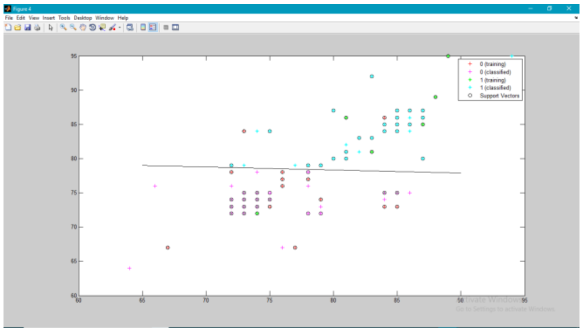


DATA RETRIVAL


MATLAB 7.9.0 (R2009b)
File Edit Debug Parallel Desktop Window Help
Current Folder: C:\Users\HP\Desktop\PROJECT\fwddatareversableforsvmcode
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Workspace
Command History

```
Message1:
    '123456'

Message2:
    '23456'

Message3:
    '3456'

Message4:
    '456'


svmStruct =

          SupportVectors: [150x2 double]
                   Alpha: [150x1 double]
                    Bias: 0.3333
          KernelFunction: @linear_kernel
      KernelFunctionArgs: {}
              GroupNames: [150x1 double]
    SupportVectorIndices: [150x1 double]
               ScaleData: [1x1 struct]
           FigureHandles: {[520.0007]  [521.0012 522.0007]  [575.0007]}


classes =

     1
     1
     1
     1
     1
```

SVM CLASSIFICATION



CONCLUSION AND FUTURE SCOPE

8.1 Conclusion:

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.