High Capacity Reversible Data Hiding in Encrypted Transform Domain for Privacy Protection

Jeeva K A [1*], Sheeba V S [2]

[1] *Department of Electrical and Electronics Engineering*, Government Engineering College, Thrissur, Kerala, India.
[2] *Department of Electronics and Communication Engineering*, Government Engineering College, Kozhikode, Kerala, India.
*Corresponding author: jeevapadmakumar@gmail.com

***Abstract* - Reversible data hiding in the encrypted domain is emerging as a promising solution in secure data hiding applications. When implemented in the encrypted transform domain, it can provide high robustness along with security and privacy. Here we propose two different algorithms, Algorithm 1 and Algorithm 2, with high embedding capacity for reversible image data hiding in the encrypted discrete cosine transform (e-DCT) domain. Both algorithms use Paillier scheme for image encryption. Additive homomorphic property and probabilistic property of Paillier scheme are used to embed the secret data in the encrypted cover image. The Algorithm 1 allows hidden data extraction in plain-text domain whereas Algorithm2 supports the same in the encrypted domain. These algorithms outperform the existing algorithms in terms of the embedding capacity and provide higher PSNR for the same embedding strength. The hidden data, as well as the original cover, can be extracted without loss.**

Keywords - Reversible Data Hiding, Encrypted Transform Domain, Privacy Protection

I. INTRODUCTION

In the past two decades, the term Data hiding has drawn much attention from the research community. Using this technique, we can embed a secret code into a cover medium and the authorized user can extract it later for various applications. The cover medium mentioned here could be either audio, image or video. In primitive works, the quality of the cover medium was given least importance in comparison with the secret data endorsed during the retrieval process. But there are certain scenarios where the quality of cover medium is equally important as the secret code during the retrieval. Reversible data hiding (RDH) which is emerged as a solution to this problem, tries to recover both the cover medium and secret data without loss on the receiver side. In 1997 Barton proposed the first RDH algorithm for authentication of digital data [1]. Since then several algorithms have been reported in the literature which makes use of images as their cover medium [2-3]. RDH is still a hot research topic that finds challenging applications in remote healthcare monitoring, military communications, etc.

Depending on the domain in which secret data is embedded, RDH algorithms can be implemented either in the spatial domain or the frequency domain. Different techniques are used in literature that created additional space for embedding secret data. In compression based methods, a portion of the cover image is compressed to create vacant space for embedding secret data. LSB based data hiding and its variants are very popular in spatial domain schemes [4-5]. Frequency domain based compression techniques use discrete cosine transform (DCT), discrete wavelet transform (DWT), etc. for the same purposes [6-7]. Image histogram modification is another method used for embedding secret data. Here algorithm may use the histogram of the entire image or block-based histograms for modification [8-9]. Difference expansion (DE) based RDH is considered as a breakthrough in the reversible data hiding techniques [10]. The algorithm could attain an embedding capacity of 0.5 bit per pixel (bpp) at relatively low computational cost along with low distortion. DE technique has been later improved by many researchers [11-12].

Encryption plays a crucial role in RDH when privacy becomes a major concern. This privacy issue arises when the content owner depends on a third party for embedding the secret data. In encrypted domain RDH techniques (RDH-ED), the cover image is normally kept in the encrypted form for security reasons. Various combinations of data hiding and encryption schemes for embedding secret data in images for privacy-preserving applications were already reported. These algorithms reserve room for embedding secret data either before encryption [13-14] or after encryption [15-16]. In the former case, the cover image needs to be preprocessed before encryption to create vacant space for holding extra data. The algorithms reported in the latter case avoid this preprocessing but generally permit embedding of fewer numbers of additional bits. This is because the encryption process introduces randomness in the original cover leaving very little redundant space for inserting secret data. The aforesaid RDH-ED systems use various symmetric cryptosystems for protecting the cover image. In fact, public key cryptosystems with homomorphic properties are better suitable for privacy preserving applications as these cryptosystems allow secret data embedding in the encrypted cover image without giving any indication regarding the pictorial information of the cover image. They also avoid the need for key distribution which is a major drawback of

symmetric key systems and keeps them ideal for cloud based applications.

Many works are recently reported in RDH-ED using Paillier cryptosystem [17-21]. These algorithms introduce various techniques like histogram shifting of absolute differences [17], mirroring of cipher text groups [18], multilayer wet paper coding [19], etc. for secret data embedding. Even though these algorithms permit a lossless retrieval of cover image and secret data, the maximum embedding rate permitted is relatively small (<=0.5bpp) . In [20], authors proposed two algorithms for RDH-ED in the spatial domain with high embedding capacity using Paillier encryption. The first algorithm embeds secret data through expansion of encrypted values which on decryption will produce either an odd or even value depending on the secret binary value. The second algorithm uses the self-blinding property of encryption scheme to embed the secret data and permits extraction only in cipher domain. The embedding method proposed in [21] used histogram shifting. There are two different extraction methods proposed, called lossy and lossless, and only the lossy method provides a high embedding rate (1 bpp) at the expense of distortion to the recovered host image.

RDH-ED in the transform domain is a research area which is yet to be explored. Even though RDH-ED in the spatial domain is a promising tool for secure signal processing, when implemented in the transform domain, it acquires more robustness against signal processing attacks. Researchers have already implemented some popular transforms for 1D and 2D signals in the cipher domain. Encrypted domain implementations of discrete Fourier transform (DFT), fast Fourier transform (FFT) and discrete cosine transform (DCT) for ID and 2D signals were already well investigated [22-24]. In [25], authors present the implementation of discrete wavelet transform in the encrypted domain. They also provide a novel method to reduce the data expansion through multiplicative inverse method (MIM). Only a few algorithms on data hiding techniques reported so far use encrypted transform domain implementations. Computation of Walsh-Hadamard transform (WHT) and fast Walsh-Hadamard transform for encrypted data are presented in [26] and also proposed a watermarking algorithm based on the same transform. The algorithm embeds 1-bit additional data in a block of size 8x8 through cardinal point concept. Another implementation of image watermarking uses a hybrid scheme of encrypted DCT and encrypted DWT [27]. The algorithm first performs DWT on the image to extract the LL band which is further divided into blocks of size m x m. Each block is divided into similar sub-blocks through image splitting, and DCT is applied on these sub-blocks. Selected mid-band coefficient of one sub-block is modified in terms of similar coefficients in another sub-block according to the nature of the watermark bit. These algorithms permitted blind data extraction either in plain text domain or encrypted domain. However, extraction of secret data utilizes the relationship of data in the watermarked positions with the unmodified data in its surrounding locations. This correlation based extraction makes these algorithms vulnerable to statistical attacks and imposes an upper limit on the payload capacity. The embedding capacity cannot be increased beyond 0.5 bpp here since modification of every coefficient demands at least another one to be kept unaltered for hidden data retrieval.

In this paper, we present two algorithms for robust image watermarking in the encrypted DCT domain with high embedding rate. Both algorithms use Paillier scheme for encryption. Algorithm 1 modifies e-DCT coefficients for embedding and facilitates hidden data extraction in plain text domain. Algorithm 2 uses self –blinding property for data embedding and permits data recovery only in the encrypted domain. Self- blinding property allows different cipher text representations for a single plaintext. Both algorithms outperform its predecessors in embedding capacity and provide a better PSNR for the same embedding strength. The algorithms ensure an uncorrelated encrypted cover input for embedding process which makes them robust to statistical attacks.

The remainder of the paper is organized as follows. Section 2 discusses preliminaries which include Paillier cryptosystem and implementation of DCT in the encrypted domain. The proposed algorithms are explained in Section 3. Simulation results and performance analysis are given in Section 4. Finally, Section 5 concludes the paper.

II. PRELIMINARIES

This section briefs the concepts of Paillier cryptosystem, homomorphism in public key cryptosystems and the implementation aspects of DCT in the encrypted domain.

A. Paillier Cryptosystem

Paillier cryptosystem is a public key cryptosystem with probabilistic property [28]. The key generation, encryption and decryption processes are shown below.

Key Generation:

Let p and q be two large prime numbers such that $N = pq$, $\lambda = \text{lcm}((p-1),(q-1))$ and $g \in Z^*_{N^2}$.

N divides the order of g so that the minimum value of g is $N+1$ and $\mu = (\text{L}(g^{\lambda} \bmod N^2))^{-1} \bmod N$.

where the function $\text{L}(x)$ is defined as $\text{L}(x) = \dfrac{x-1}{N}$.

(N, g) is the key for encryption and the key for decryption is (λ, μ).

Encryption:

Let m be the given message ($m \in Z_N^*$), *r* be a random number ($r \in Z_N^*$), and c be the corresponding cipher text. Then $c = g^m r^N \bmod N^2$. The parameter r is called the blinding variable whose value is chosen differently for each encryption. The role of r is to make the encryption scheme probabilistic, mapping same plaintext value to different cipher texts at different instances of time.

Decryption:

For the cipher text *c* ($c \in Z_{N^2}^*$), the corresponding plaintext m is recovered as:

$$m = \mathrm{L}(c^{\lambda} \bmod N^2)\mu(\bmod N).$$

B. Homomorphism in Public-Key Cryptosystems

Although the use of encryption algorithms avoids illegal access to digital data, it is not sufficient to prevent unauthorized access by an adversary and protect it along its lifetime. Once decrypted, the data lose its security and are susceptible to signal processing attacks. Many public key cryptosystems permit processing of signals in the encrypted domain. Homomorphism exhibited by these public-key cryptosystems enable a third party to hide a secret data into a cover without knowing what actually the cover is.

A homomorphism is a map between the elements of two algebraic systems. A cryptosystem is said to be homomorphic with respect to operation $(\circ)$, if there exists an operation $(\cdot)$ such that for two message inputs m_1 and m_2, we have:

$$D[[E(m_1)](\cdot)[E(m_2)]] = m_1(\circ)m_2 \bmod N \tag{1}$$

Where *E*[ ] and *D*[ ] denotes the encryption and decryption operation respectively. Thus additively homomorphic cryptosystems map an operation in the ciphertext domain to an addition in the plain text domain:

$$\text{i.e. } D[E[m_1](\cdot).E[m_2]] = m_1 + m_2 \tag{2}$$

$$\text{also } D[E[m]^k] = km \tag{3}$$

where k is a public integer. Most of the public key cryptosystems exhibit either additive or multiplicative homomorphism. Table 1 shows some of the popular public key cryptosystems and the type of homomorphism exhibited by them.

Paillier scheme exhibits additive homomorphism. Here a multiplication operation in cipher domain is analogous to an addition in plaintext domain.

Let c_1 and c_2 be two cipher texts in Paillier scheme for messages m_1 and m_2, then:

$$D[c_1 \cdot c_2] = D[g^{m_1} r_1^N . g^{m_2} r_2^N \bmod N^2]$$

$$= D[g^{m_1+m_2} r^N \bmod N^2 = m_1 + m_2$$

where $r = r_1 \cdot r_2$.

C. Implementation of DCT in encrypted domain (e-DCT)

In [24] Bianchi et al. investigated the implementation of DCT in the encrypted domain.

The two-dimensional DCT for input $\mathbf{x}$ in the plain domain is defined as

$$\mathbf{y}_{pq} = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} \mathbf{x}_{ij}\, \mathrm{C}_M(i,p)\mathrm{C}_N(j,q) \tag{4}$$

where $0 \le p \le M-1$ and $0 \le q \le N-1$. The product of M and N gives the size of the input signal $\mathbf{x}$ and C represent the transformation coefficient function.

Equivalent representation in the encrypted domain can be defined as:

$$\mathbf{Y}_{pq} = \prod_{i=0}^{M-1}\prod_{j=0}^{N-1} \mathbf{X}_{ij}^{\mathrm{C}_M(i,p)\mathrm{C}_N(j,q)} \tag{5}$$

where $\mathbf{X}$ represents encrypted input and $\mathbf{Y}$ represents output coefficients in the encrypted domain.

Transformation coefficients C are here integer approximated.

TABLE I. HOMOMORPHISM EXHIBITED BY DIFFERENT ENCRYPTION SCHEMES

Name of crypto-system	Operation on encrypted domain	Equivalent operation in the plain domain
Paillier	Multiplication	Addition
RSA	Multiplication	Multiplication
Okamoto-Uchiyama	Multiplication	Addition
Damgard-Jurik	Multiplication	Addition
Goldwasser-Micali	Multiplication	XOR

III. REVERSIBLE DATA HIDING ALGORITHMS

The scenario considered here involves three different parties. The signal provider owns the cover which is a grey image. He/she encrypts this image with the receiver's public key and gives it to the data hider. The data hider either generates the secret data by himself or receives the binary encoded secret data from the signal provider. He performs the embedding operation on the encrypted DCT coefficients of the cover image using homomorphism and provides the result to the authorized receiver.

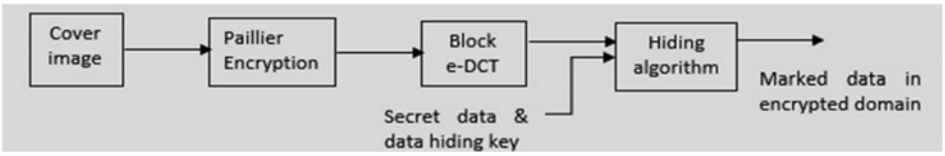


Fig. 1.A. Embedding in e-DCT domain

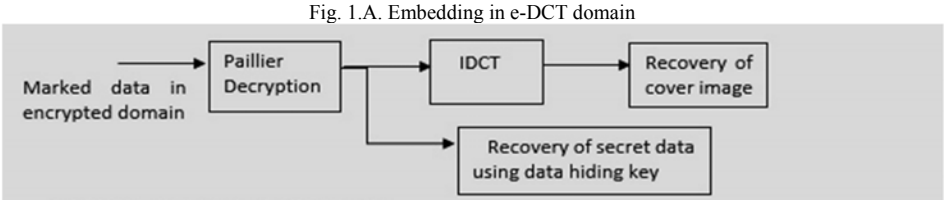


Fig. 1.B. Extraction in Plain text domain

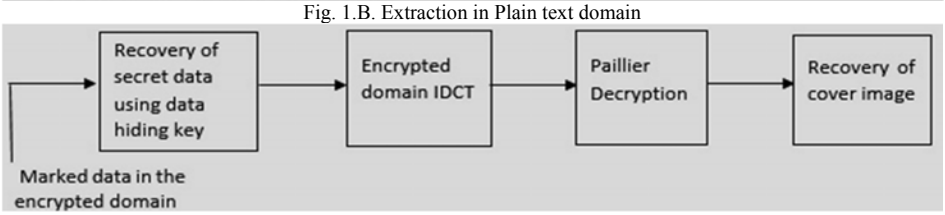


Fig. 1.C. Extraction in Encrypted Domain

The receiver extracts the secret data either in the plaintext domain (using Algorithm 1) or in the cipher domain (using Algorithm 2) according to the algorithm implemented. Figure 1 gives the block diagram representation of proposed algorithms for RDH-ED in the e-DCT domain. Algorithm 1 proposed here modifies e-DCT coefficients to hide the secret data, where coefficients are selected from high-frequency region towards low frequency in every block of size 8x8 as embedding rate increases. Algorithm 2 makes use of the self-blinding property of Paillier system to hide the data in every e-DCT coefficients.

A. Implementation of Algorithm 1

Embedding:

1. Encrypt the cover image using Paillier encryption with a key size of 1024 bits.
2. Divide the encrypted image into blocks of size 8 x 8 and apply two-dimensional e-DCT using equation (5).
3. Using the data hiding key, select the coefficients for modification in each block.
4. Perform data hiding by modifying the selected coefficients in each block using the following equation:

$$EF_{ijw} = \begin{cases} EF_{ij}^2 \cdot E[1] & \text{if } b = 1 \\ EF_{ij}^2 & \text{if } b = 0 \end{cases} \quad (6)$$

where $E[1]$ is the ciphertext for binary 1. EF_{ij} and EF_{ijw} represent the original and modified e-DCT coefficients of cover image and b is the secret bit for hiding.

In plain domain, this operation is equivalent to:

$f_{ijw} = 2 f_{ij} + b$ which generates either an odd value or even value of DCT coefficients for b=1 and b=0 respectively. f_{ij} and f_{ijw} represent original and modified DCT coefficients in plain domain.

Extraction:

1. Perform Paillier decryption of modified e-DCT coefficients to compute DCT coefficients in plain domain (f_{ijw})
2. Extract hidden data in block each using data hiding key as

$$b = f_{ijw} \bmod 2 \quad (7)$$

3. To reconstruct the cover, original DCT values are computed through equation

$$f_{ij} = \left\lfloor \frac{f_{ijw}}{2} \right\rfloor \quad (8)$$

where symbol $\lfloor . \rfloor$ represents floor operation.

4. Apply inverse DCT in plaintext domain on f_{ij} values to reveal the original gray values of the cover medium
5. Perform inverse DCT in plaintext domain on f_{ijw} values to reveal the cover image with embedded hidden data.

B. Implementation of Algorithm 2

Algorithm 2 exploits the self-blinding property of Paillier cryptosystem to hide the data in encrypted transform domain coefficients. This property helps to change one cipher text to another without actually changing its decryption value. In

this paper, it is used for suitably tailoring odd/even nature of e-DCT coefficients to hold the secret data.

The following equation explains the property of self-blinding:

If $E[m]$ represents a cipher in Paillier system, then:

$$E[m]\cdot(r^N \bmod N^2) \bmod N^2 = E[m]\cdot r^N \bmod N^2 \quad (9)$$

$$\text{and } D[E[m]\cdot r^N] = m \bmod N \quad (10)$$

Embedding:

1. Encrypt the cover image using Paillier encryption with a key size of at least 1024 bits.
2. Divide the encrypted image into blocks of size 8x8 and apply two-dimensional e-DCT using equation (5).
3. Select the coefficients to be modified in each block using the data hiding key as EF_{ij}
4. To embed a binary bit b into a chosen coefficient, perform self-blinding on EF_{ij} as:

$EF_{ij} = EF_{ij}\cdot r^N \bmod N^2$ for different values of r until it satisfies the condition:

$$b = EF_{ij} \bmod 2 \quad (11)$$

Extraction:

1. Identify the modified e-DCT coefficients in each block using the data hiding key.
2. Extract secret data bit b from e-DCT coefficients using equation (11).
3. Perform inverse e-DCT followed by Paillier decryption to reconstruct the cover image.

IV. RESULTS AND DISCUSSIONS

In this section, we first analyze the security of encryption scheme towards statistical attacks, discuss the performance of the proposed schemes and compare the results with recently reported algorithms for RDH. Popular gray image Lena of size 128 x 128 is used as the cover image. While implementing the Paillier cryptosystem, the length of the key is chosen as 1024 bits to ensure practical security. Secret data is a sequence of randomly chosen binary values. The DCT coefficients for embedding secret data in each block were selected from the high-frequency region towards low-frequency region as the payload increases. The algorithms have been implemented in C++ using the GNU Multi-Precision library and the NTL library for processing of integers of arbitrary length

.

A. Security of Encryption Scheme

Many of the already implemented RDH algorithms in the encrypted domain or encrypted transform domain make use of the correlation between adjacent pixels or adjacent blocks for extraction of hidden data. It means that, in these encryption schemes, encrypted pixel values or encrypted transform coefficients retain correlation even after encryption as appears in natural images. In a deterministic cryptosystem like RSA, encryption effectively makes a one-to-one mapping since each pixel value will be uniquely converted to another fixed value in the expanded domain. This situation will favour a cryptanalyst to understand the information content of the original image through histogram analysis, irrespective of the large key size used. A cryptosystem provides robustness against statistical attacks if it can provide a low correlation between adjacent pixels. In Paillier cryptosystem, being probabilistic in nature, different cipher text representations are possible for a plaintext using the same public key as the blinding parameter is selected randomly during each encryption. Decryption of these cipher texts with the corresponding private key generates the original plaintext. Thus Paillier cryptosystem achieves semantic security.

To analyse the statistical security of encrypted data, the commonly accepted methods are the computation of correlation coefficient, histogram analysis and entropy calculation.

The correlation coefficient of adjacent pixels in any given direction is computed using the following equation.

$$(X,Y)_{CORR} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{X_i - \overline{X}}{\sigma_X}\right)\left(\frac{Y_i - \overline{Y}}{\sigma_Y}\right) \quad (12)$$

where N denotes the number of tuples (X_i, Y_i), $\overline{X}$ and $\overline{Y}$ represent the mean σ_X and σ_Y represent the standard deviation of X_i and Y_i respectively in the given direction. A low value for correlation indicates high randomness present in the data.

Figure 2 shows the cover image and its encrypted counterparts used for the implementation and analysis of algorithms presented in this paper and Table 2 provides the correlation coefficient computed for these images in different directions.

TABLE II. HORIZONTAL AND VERTICAL CORRELATION IN THE PLAINTEXT AND THE ENCRYPTED DOMAIN

$(X,Y)_{CORR}$	Original Image	Encrypted Image	e-DCT Coefficients
Horizontal	0.9505	-0.0035	0.0077
Vertical	0.8937	-0.0040	-0.0090

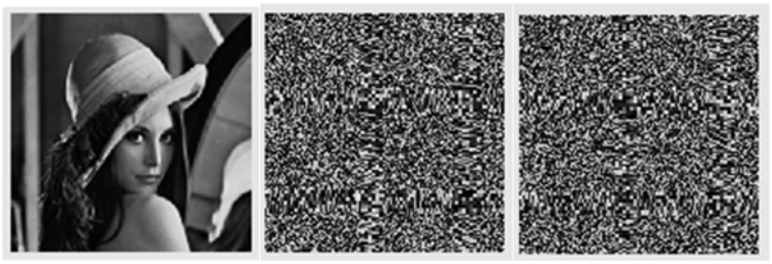
Figure 3 plots the histogram of the cover image and encrypted cover image. It is seen that for the encrypted image, the histogram gives a nearly uniform distribution which ensures the robustness of the data towards histogram based attacks.

Entropy is another statistical measure of randomness which characterizes the texture of an image. It is a scalar quantity and is measured for a grey image as:

$$E = -\sum_{i=1}^{n} p(X_i)\log_2 p(X_i) \quad (13)$$

Where p denotes the histogram counts.

The calculation using equation (13) shows that the entropy of the cover image used is 7.3883 whereas for the encrypted cover image, it is 7.9599. It is observed that for the encrypted version, the entropy value closely approaches its ideal value 8 implying negligible leakage of texture information. The above analysis proves that the Paillier scheme implemented here can effectively eliminate the correlation and makes the encrypted data secure against statistical attacks.



-a. –b. –c.

Fig. 2. Cover image and its encrypted counterparts (a) cover image 'Lena' (b) Encrypted Lena (c) e-DCT of Lena

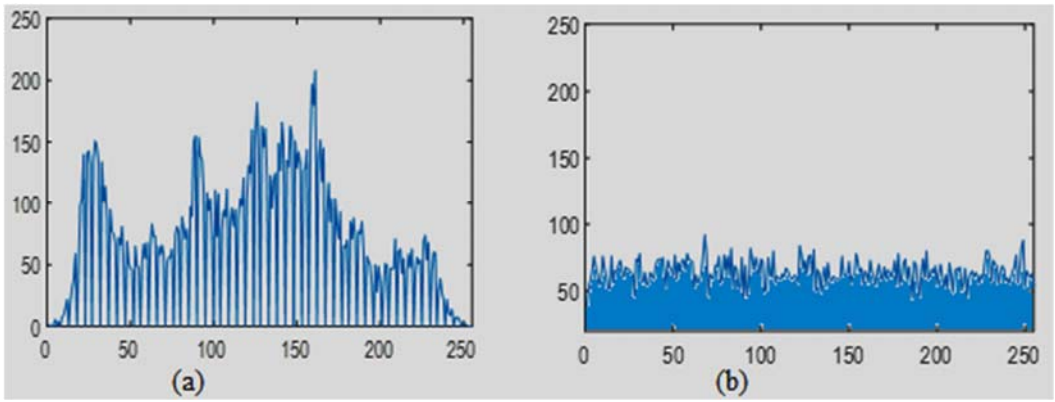


Fig. 3. Histogram of (a) cover image (b) encrypted cover image

B. Performance of Algorithm

The quality of cover retrieved and embedding capacity are the two major parameters that define the effectiveness of a data hiding algorithm. For data hiding algorithms proposed here, the embedding capacity is either one bpp or very close to it. Homomorphic embedding avoids the disclosure of sensitive cover data to an untrusted embedder. Original cover data can be extracted only by the authorized receiver using his private key without any loss, which provides enough robustness. The bit error ratio (BER) is zero for the extraction of hidden secret data in both cases in their respective domains of extraction.

We use the well-known peak-signal-to-noise ratio (PSNR) measure and bit error rate (BER) to evaluate the quality of the embedded images and the robustness of the algorithms. Figure 4 compares the embedding capacity of proposed algorithms with the algorithms reported in encrypted transform domains [26, 27] and Figure 5 compares their PSNR vs. embedding strength performance. In both cases, the proposed algorithms clearly dominate the performance of the aforementioned algorithms. Figure 6 shows the perceptual quality of the retrieved marked images for various embedding strength using Algorithm 1 and compares it with the results of the algorithm proposed in [26] and [27]. It is obvious that for algorithm 2, the PSNR is infinity in all cases.

Performance of the algorithms towards common attacks has been evaluated in this paper. Assuming that embedding algorithm is secure and a brute-force attack is practically impossible with the large key size used, the effect of attacks that are familiar in plain text domain on the retrieved images are analysed. Random noise is applied by shuffling the encrypted pixel values of the embedded cover at random locations. Salt and pepper noise is also applied by replacing selected encrypted pixel values of the cover by randomly chosen encrypted values of 0 and 255. A cropping attack in the encrypted domain can be done by replacing the selected block with encrypted zero values.

It is clear that all the above alterations affect the quality of the retrieved cover and induce errors in the secret data retrieved. Figure 7 analyses the effect of these operations on the embedded cover image for different capacity using Algorithm 1. For Algorithm 2, the embedding procedure

does not bring any errors in the recovered image. Thus embedding capacity does not play any role in the computation of PSNR or BER in the noise-free condition. Any attempts to change the encrypted pixel values of the embedded image will directly reflect on the retrieved cover and produce a similar effect as if it is done on the original cover image in the plaintext domain.

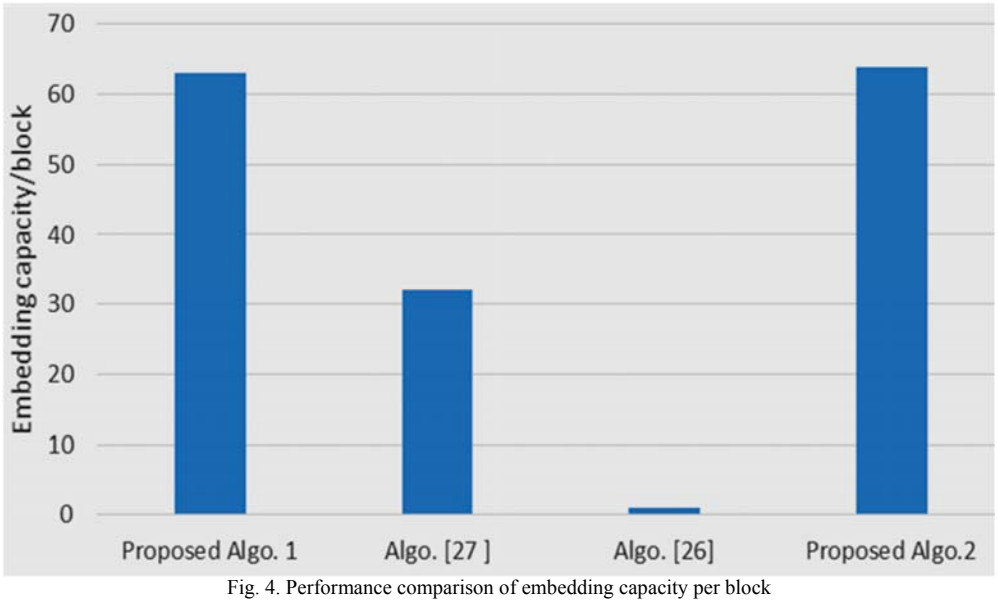


Fig. 4. Performance comparison of embedding capacity per block

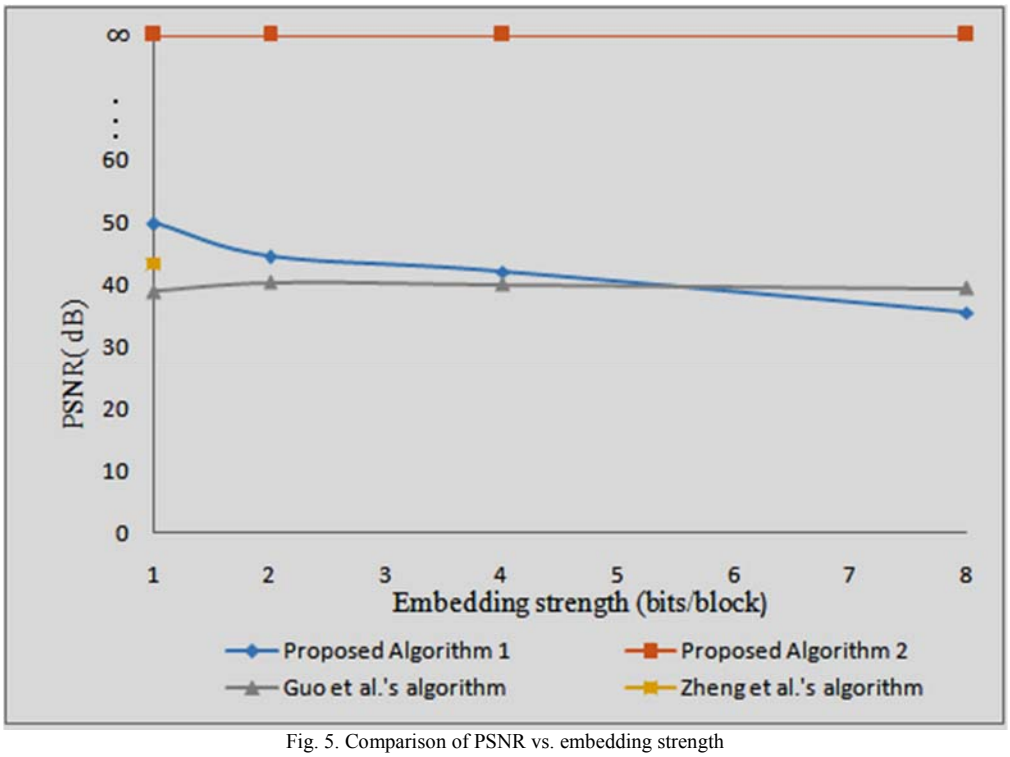


Fig. 5. Comparison of PSNR vs. embedding strength

Fig. 6. Watermarked images with Algorithm 2 for different embedding rate: (a) 1/64 bpp (b) 1/32 bpp (c) 1/16 bpp (d) 1/8 bpp (e) 1/4 bpp (f) 1/2 bpp (g) 3/4 bpp (h) 63/64 bpp

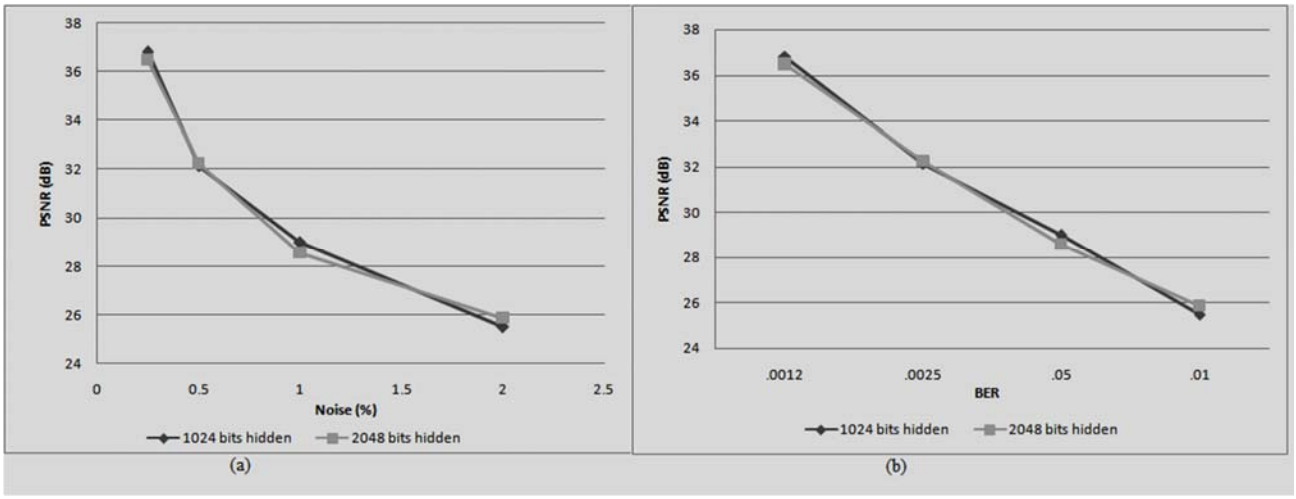


Fig. 7.a Effect of random noise on embedded cover with different embedding strength (a) Noise vs. PSNR performance (b) BER vs. PSNR performance.

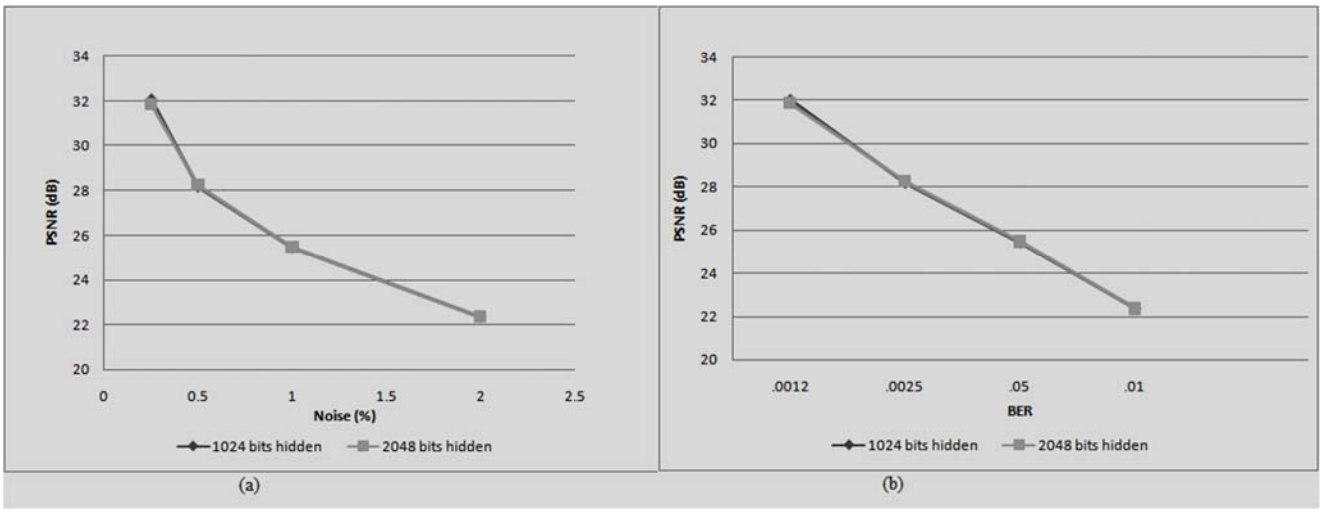


Fig. 7.b Effect of salt and pepper noise embedded cover with different embedding strength (a) Noise vs. PSNR performance (b) BER vs. PSNR performance

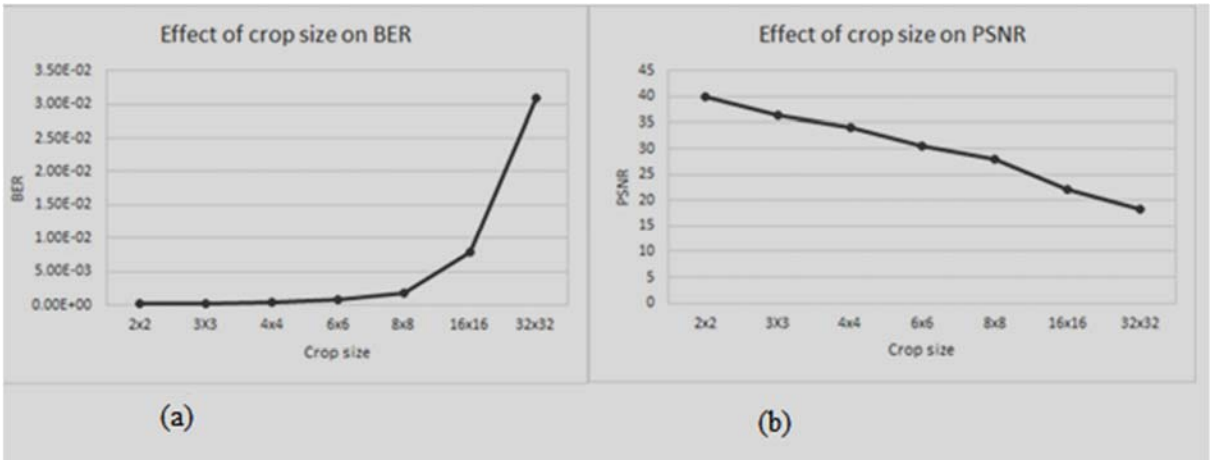


Fig. 7.c Effect of cropping on 256 bits embedded image (a) crop size vs. BER performance (b) crop size vs. PSNR performance

V. CONCLUSION

Reversible data hiding in the encrypted DCT domain using homomorphic encryption has been investigated here. The proposed algorithms make use of Paillier encryption with a key size minimum 1024 bits to ensure data security. Data embedding is performed in the transform domain to improve the robustness. The first algorithm supports hidden data extraction after image decryption while the second algorithm makes it possible in the cipher domain itself. In both cases, the original cover image and the hidden data can be retrieved without loss. Low inter-pixel correlation in encrypted domain ensures statistical attacks difficult. The noise analysis in the encrypted domain proves that the algorithms can survive common attacks. Both these algorithms provide high embedding strength with acceptable PSNR and are ideal in a scenario where an untrusted third party performs embedding process.

REFERENCES

[1] Barton J.M. 'Method, and apparatus for embedding authentication information within digital data', U.S. Patent 5 646 997 ,Jul. 8, 1997.

[2] Fridrich, J., Soukal, D.: 'Matrix embedding for large payloads', IEEE Trans. Inf. Secur. Forensics, 2006, 1, (3), 390-394.

[3] Munuera, C.: 'Steganography and error-correcting codes', Signal Process., 2007, 87, (6), 1528-1533.

[4] Celik, M. U., Sharma, G.,Tekalp, A. M., et al.: 'Lossless generalized-LSB data embedding', IEEE Trans. Image Process, 2005, 14, (2), 253-266.

[5] Yang, C. H.,Weng, C.Y., Wang, S., et al.: 'Adaptive data hiding in edge areas of images with spatial LSB domain systems', IEEE Trans. Info. Forensics and Secure, 2008,3, (3), 488-497.

[6] Parah, S. A., Sheikh, J. A., Loan, N. A., et al.: 'Robust and blind watermarking technique in DCT domain using inter-block coefficient differencing. Digital Signal Processing', 2016, 53, 11-24, 2016.

[7] Kamstra, L., Heijmans, H. J. A. M.: 'Reversible data embedding into images using wavelet technique and sorting', IEEE Trans. Image Process., 2005, 14, (12), 2082-2090.

[8] Li, X, Zhang, W., Gui, X., et al.: 'Efficient Reversible Data Hiding Based on Multiple Histogram Modification', IEEE Trans. Inf. Forensics Secur., 2015,10, (9), 2016–2027.

[9] Chen, H., Ni, J., Hong, W., et al.: 'Reversible data hiding with contrast enhancement using adaptive histogram shifting and pixel value ordering', Signal Process. Image Commun., 2016, 46, 1-16.

[10] Tian, J.: 'Reversible Data Embedding Using a Difference Expansion', IEEE Trans. Circuits and Syst. Video Techno.l, 2003, 13, 890-896.

[11] Dragoi, L.-C., Coltuc, D.: 'On local prediction based reversible watermarking', IEEE Trans. Image Process., 2015, 24, (4), 1244-1246.

[12] Ou, B., Li, X., Wang, J.: 'High-fidelity reversible data hiding based on pixel-value-ordering and Pairwise prediction error expansion', J. Vis. Commun. Image Represent., 2016,39, 12-23.

[13] Shiu, C.-W., Chen Y.-C., Hong, W.: 'Encrypted image-based reversible data hiding with public key cryptography from difference expansion', Signal Process. Image Commun., 2015,39, 226-233.

[14] Cao, X., Du, L., Wei, X., et al.: 'High capacity reversible data hiding in encrypted images by patch-level sparse representation', IEEE Trans. Cybern., 2016, 46, (5), 1132-1143.

[15] Liao, X., Shu, C.: 'Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels', J. Vis. Commun. Image Represent., 2015, 28, 21-27.

[16] Zhou, J., Sun, W., Dong, L., et al.: 'Secure reversible image data hiding over encrypted domain via key modulation. IEEE Trans. Circuits. Syst. Video Technol., 2016, 26, (3), 441-452.

[17] Zheng, S., Li, D., Hu, D., et al.: 'Lossless data hiding algorithm for encrypted Images with high capacity', Multimedia Tools Appl., 2016.

[18] Wu, H.-T., Cheung Y.-m., Huang, J.: 'Reversible data hiding in Paillier cryptosystem', J. Vis. Commun. Image Represent, 2016,40, 765-771.

[19] Rama Thulasi, P., Vijayalaskshmi, C. " Design and analysis of clinical trials using statistical techniques- A review", (2018) International Journal of Pharmaceutical Research, 10 (3), pp. 541-546.

[20] Xu, D., Wang, R.: 'Separable and error-free reversible data hiding in encrypted images', Signal Process., 2016, 123, 9-21.

[21] Zhang, X., Long, J., Wang, Z., et al.: 'Lossless and Reversible Data Hiding in Encrypted Images with Public-Key Cryptography', IEEE Trans. Circuits Syst. Video Technol., 2016, 25, (2), 1622-1631.

[22] Bianchi, T., Piva, A., Barni, M.: 'Comparison of different FFT implementations in the encrypted Domain', In Proc. EURASIP EURASIPCO, 2008.

[23] Bianchi, T., Piva, A., Barni, M.: 'On the implementation of discrete Fourier transform in the encrypted domain. IEEE Trans. Inform. Forensics and Security, 2009, 4, (1), 86-97.
[24] Bianchi, T., Piva, A., Barni, M.: 'Encrypted domain DCT based on homomorphic cryptosystems', EURASIP J. Inform. Security, 2009.
[25] Zheng, P., Huang, J.: 'Discrete wavelet transform and data expansion reduction in homomorphic encrypted domain', IEEE Trans. Image Processing, vol. 22, no. 6, pp.2455-2468, 2013.
[26] Zheng, P., Huang, J.: 'Walsh-Hadamard transform in the homomorphic encrypted domain and its application in image water marking', *In* Information Hiding, Springer, 2013, 240-254.
[27] Guo, J., Zheng, P., Huang, J.: 'Secure Watermarking scheme against watermark attacks in the encrypted domain', J. Vis. Commun. Image Represent, 2015, 30, 125-135.
[28] Paillier, P.: 'Public-key cryptosystems based on composite degree residuosity classes. Adv. Cryptology-EUROCRYPT99, Springer, 1999, 223-238.