VOL. 17, NO. 18, SEPTEMBER 2022 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

A SECURE AND HIGH CAPACITY PVD STEGANOGRAPHY SCHEME USING COMPRESSION, RSA AND QKD

Kalyan K[1], Nayanesh G[1], Puneeth G[1], Ravikumar CV[2] and Kalapraveen Bagadi[2]
[1]SENSE, Vellore Institute of Technology, Vellore, Tamil Nadu, India
[2]SENSE, Vellore Institute of Technology, Vellore, India
E-Mail: ravikumar.cv@vit.ac.in

ABSTRACT

In this paper we propose a highly secure and high capacity PVD steganography scheme in which LZW compression is used to increase embedding capacity and QKD is used to improve upon the security of the RSA public key. The compressed secret data is encrypted using RSA algorithm resulting an encrypted data which is embedded into the cover image using PVD algorithm. The scheme uses Hilbert fractal-based pixel traversal and selection method so as to increase the randomness of the embedding process. The scheme achieved an increase in the embedding capacity of 24.6% when compared with existing methods. The proposed scheme also achieves an average PSNR value of 40.96 dB at 4.99 bpp. The scheme is resilient to quantum computing and steganalysis like pixel difference histogram (PDH).

Keywords: QKD, RSA, LZW, PVD, PDH, hilbert fractal.

1. INTRODUCTION

In the present world, with the advent of new technologies it is always been a major concern to protect the privacy and integrity of the data. To protect the data over a communication channel an effective technique named Steganography emerged. Steganography methods can be mainly divided into two categories based on the domain used for hiding the data, they are spatial and frequency domain-based steganography [1].

Steganography methods can also be classified into three categories based on the type of encryption used in them: true steganography, private key steganography and public key steganography [2]. A true steganography scheme does not use any encryption for data protection so it is easily vulnerable to data purging. A private key steganography scheme makes use algorithms like Advanced encryption standard (AES) and Data encryption standard (DES) which use an identical key at both encryption and decryption. In a private key steganography scheme the system makes use of asymmetric key encryption standards like RSA and

ECC which use two different keys namely a public key for encryption and a private key for decryption. In any of the scheme there is always a need for some kind of key exchange mechanism like Diffie Hellman [3].

Now with the onboarding of new quantum computing technologies it has become much easier to break into public key encryption techniques like RSA and ECC using Shor's algorithm [4]. In this work we propose a safe steganography scheme that is resilient to quantum computing. The secret data that is to be hidden in the image is first compressed using LZW compression and then encrypted using RSA algorithm. The encrypted data is then embedded into the image using Hilbert fractal based PVD algorithm.

The new PVD algorithm ensures that the data is placed in a systematic order while appearing to be random for any attackers. The sender and receiver agree upon a symmetric quantum key that they exchange using QKD, this symmetric quantum key is then used to encrypt the public key for RSA algorithm. The structure of remaining paper is as follows: Section 2 describes the literature review, followed by Section 3 illustrating the proposed method, section 4 describes the Results and discussion and finally Section 5 describes the conclusion.

2. LITERATURE SURVEY

In the recent times, there has been a large amount of research relating to steganography, many researchers are trying to improve the embedding capacity and security of steganography schemes while preserving the image quality. In this section many such spatial domain-based steganography schemes are discussed.

C.-K. Chan *et al* [18] proposed an optimal pixel adjustment process (OPAP) with a simple LSB substitution technique. OPAP was proposed to reduce the visual artifacts in the image obtained by simple LSB substitution. The system achieves good hiding capacity but at an expenses of image quality and PSNR values. Also, LSB substitution techniques are easily detectable by RS steganalysis.

Kalaichelvi *et al*.'s [2] system uses double RSA algorithm to encrypt the secret data. The Canny edge detector is then used to divide the image into edge and non-edge pixels. The encrypted data is only saved in the edge pixels of the cover image using basic LSB substitution scheme. The authors achieved good PSNR values, however the capacity of the system is very less as the data is only being embedded into the edge pixels of the image. The revised RSA algorithm used in the work is also not resilient to quantum computing.

A.K Shuka *et a*l.'s [5] scheme uses modified PVD approach which is a combination of 2x2 PVD and LSB substitution. First the data is compressed using arithmetic coding to increase the embedding capacity by up to 22%. The compressed data is then encoded using AES algorithm and then embedded into the cover image using MPVD algorithm. The authors achieved an embedding capacity of 4 bpp (bits per pixel), however the PSNR values are around 36 db. The system is immune to

RS steganalysis, but immunity towards PDH analysis is not tested. The authors did not address a mechanism for the key exchange for the AES symmetric key.

Wu and Tsai [6] proposed the first PVD approach for effective hiding of data under a cover image. The system embeds the secret data based on the difference between two-pixel values, but the system suffers from FOBP (Fall out of boundary problem), there by yielding some wrong bits during extraction process.

Wu *et al* [7] proposed an improvement towards [3] by using PVD embedding and LSB substitution at the edge and smooth areas of the images thereby increasing the capacity of the system, this method also suffered from FOBP and the PSNR values are reduced compared to Wu and Tsai [6].

Yang *et al* [8] proposes an improvement towards Wu *et al* [7]. The system achieved better capacity but the PSNR values are reduced. The authors showed that the system is resilient towards RS steganalysis.

Khodaei *et al*.'s [9] method hides the data using a 1x3 block of pixels. The 3-LSB method is used on the mid pixel and the difference between the first and the third pixel with respect to the mid pixel is used for PVD embedding. The system achieved higher embedding capacity when compared to Wu *et al*.'s [7] method. The FOBP problem was still persistent in this work.

Lee *et al* [10] compressed the data image using JPEG2000 lossy compression technique and then divide the image into 2x2 blocks. The bottom left corner is considered as base pixel and difference between the other pixels is used to apply PVD algorithm. A residual value coding method is proposed to overcome the lossy JPEG 2000 compression and to enhance the quality of received stego image.

Hussain *et al* [11] suggested an improved version of Yang *et al.*'s [8] method. A 2x1 pixel block from the cover image is categorized into lower and higher texture areas using the pixel value difference, and data is embedded by LSB and PVD methods respectively. The pixels are further subjected to MPE embedding followed by PVD shift to increase the overall capacity.

Khashadarag *et al.* [12] in order to increase the capacity of the system compressed the secret data using LZW compression and the encoded it using XOR operation with pseudo random numbers. The data is divided into three parts for three colour planes of the image and then DFT and frequency hopping algorithm is applied. In the end all the three planes are combined to form the stego image.

In this study we present a comprehensive special domain-based steganography method combined with high-capacity lossless compression algorithm and secure public key encryption standards. Methods introduced by Wu and Tsai are modified and improved to avoid fall out of boundary problem while increasing the embedding capacity. QKD is introduced to improve upon the security of the public key exchange of the encryption algorithm RSA. The work also included a Hilbert fractal-based pixel traversal and selection method so as to increase the randomness of the embedding process. The motivations for this work have been to increase the capacity of existing steganography schemes while securing them against quantum computing.

3. PROPOSED SYSTEM

In this paper, we propose a system that consists of six major steps like Quantum key distribution, XOR encryption / decryption, LZW data compression / decompression, RSA encryption / decryption, KLSB data embedding / extraction and PVD data embedding / extraction.

The flow of the overall system is depicted as follows:

a) Figure-2 (left) shows the symmetric key generation and distribution using QKD.

b) Figure-1 shows the reception of RSA public key using QKD on left and the embedding process on right.

c) Figure-2 shows the extraction of data from stego image on the right.

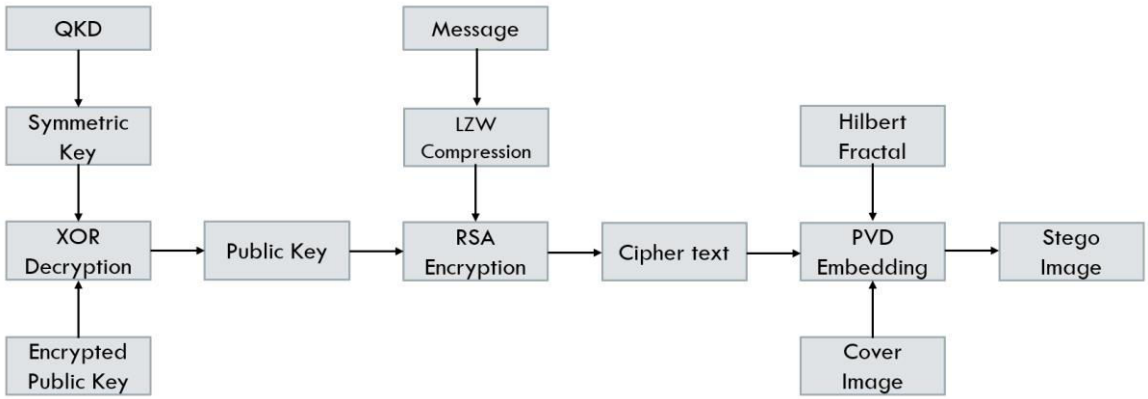

Figure-1. Transmitter side/sender side.

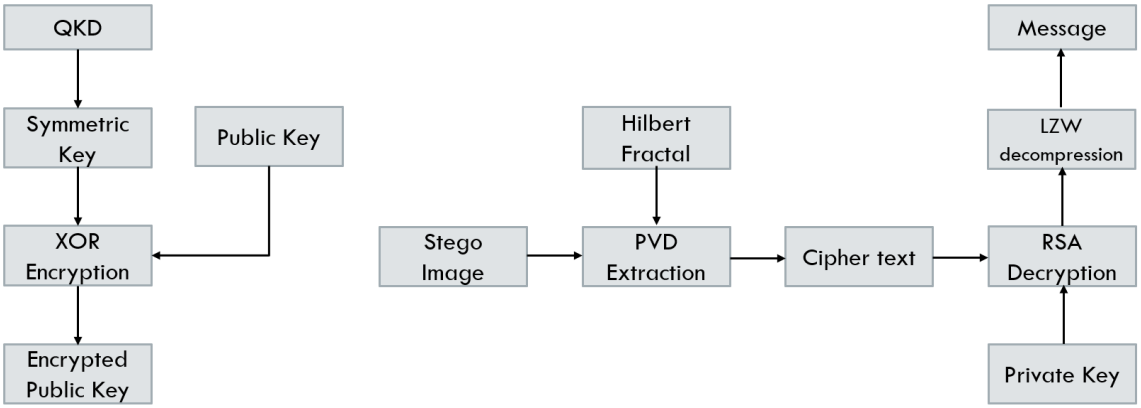

Figure-2. Receiver side.

3.1 Quantum Key Distribution

Quantum Key Distribution (QKD) is a symmetric key distribution technique which uses quantum mechanics to distribute a random secret key between two entities. Unlike classical key distribution like Diffie Hellman key exchangeQKD [3] can detect the presence of third party (Eavesdropper) who is trying gain knowledge about secret key. QKD is carried out by using a quantum channel and secured classical or public channel. The information is encoded into qubits rather than classical bits.

In this paper we are using a "Prepare and measure protocol" named BB84 [13]. This protocol uses the polarization of photons (as a qubit) to transmit information and an optical fiber as a quantum channel.

In this paper the simulation of Quantum computing for QKD is done using Quantum logic gates which are represented by unitary matrices and quantum states which are represented by "kets", from "bra-ket" notation

Table-1. Symmetric key transfer using QKD.

1. Alice generates random set of bits		**1**	**0**	**1**	**1**	**0**	**1**	**1**	**1**	**0**	**1**
		|1⟩	**|0⟩**	**|1⟩**	**|1⟩**	**|0⟩**	**|1⟩**	**|1⟩**	**|1⟩**	**|0⟩**	**|1⟩**
2. Alice generates random set of bases (either I or H)		H	I	I	H	H	H	I	I	H	H
Message sent over quantum channel		|-⟩	|0⟩	|1⟩	|-⟩	|+⟩	|-⟩	|1⟩	|1⟩	|+⟩	|-⟩
3. Bob generates random set of bases (either I or H)		I	I	H	H	H	I	I	H	H	H
▪ Message after passing through bob bases		|-⟩	|0⟩	|-⟩	|1⟩	|0⟩	|-⟩	|1⟩	|+⟩	|0⟩	|1⟩
▪ Message after measurement by bob		1	0	0	1	0	0	1	1	0	1
4. Alice and Bob share their bases though public channel	Alice Bases	H	I	I	H	H	H	I	I	H	H
	Bob Bases	I	I	H	H	H	I	I	H	H	H
5.Discard bits where the Alice bases and the Bob bases are different	Alice Key		0		1	0		1		0	1
	Bob Key		0		1	0		1		0	1
6.Share a sample of the key and discard the sample key (Ex: sample key 1st,4th and 5th bits)	Alice Key				1	0					1
	Bob Key				1	0					1

A single qubit is represented as vector shown in (1).

Figure-3 illustrates the qubit in the quantum plane.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{bmatrix}\alpha\\ \beta\end{bmatrix} \quad (1)$$

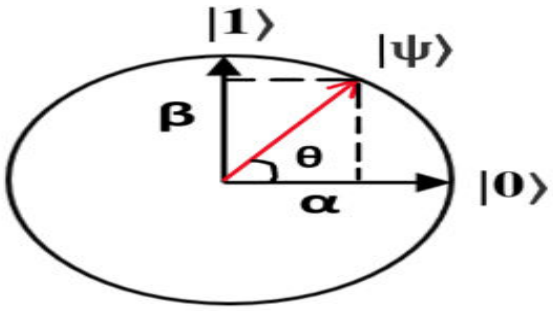

Figure-3. Qubit as a vector in quantum plane.

where α and β are the complex probability amplitudes of the single qubit. |0⟩and |1⟩states are the basis of the quantum plane. The representations for |0⟩and |1⟩states are shown in (2) and (3) respectively. |0⟩is considered as a horizontal polarized photon and |1⟩as a vertically polarized photon. When this qubit is measured the quantum state collapses to either |0⟩or |1⟩state. The probabilities of the qubit to collapse to |0⟩or |1⟩state is calculated using (4) and (5) respectively.

$$|0\rangle = \begin{bmatrix}1\\ 0\end{bmatrix} \quad (2)$$

$$|1\rangle = \begin{bmatrix}0\\ 1\end{bmatrix} \quad (3)$$

$$P(|0\rangle) = |\alpha|^2 \text{ ,where } |\alpha| = \cos\theta \quad (4)$$

$$P(|1\rangle) = |\beta|^2 \text{ ,where } |\beta| = \sin\theta \quad (5)$$

For producing a quantum superposition state, we need to use Hadamard gate. The Hadamard gate converts the qubits |0⟩and|1⟩into quantum superposition states |+⟩ and |-⟩. The superposition states |+⟩ and |-⟩ are shown as vectors in (6) and (7) respectively.

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 1\end{bmatrix} \quad (6)$$

$$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ -1\end{bmatrix} \quad (7)$$

The unitary matrix of Hadamard gates is given in (8).

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \quad (8)$$

Table-1 depicts an example of symmetric key exchange using QKD between two parties Alice and Bob. As a result of following the procedure for, m random bits yield a n bit symmetric key, (n < m).

3.2 Text Compressor/Decompressor

To increase the embedding capacity of the system, we use LZW algorithm [14] for text compression and decompression.

3.2.1 LZW compression

The compressor takes in a large text file as input and compresses it to a sequence of numbers, wherein each number is a unique identification in the dictionary. The algorithm for compression is shown below.

String dictionary: Initialize the dictionary with all single character strings;
S: First character from input;
while *input stream is non empty* do
C: next character from input;
if *String dictionary has S + C* then
S = S + C;
else
output the dictionary index for S.
add S + C as new entry in the String dictionary.
S = C.
end
end
output the dictionary index for S;

3.2.2 LZW De-compression

The decompression block takes in a sequence of numbers and maps them to the corresponding characters in the dictionary. The algorithm for de-compression is depicted below.

String dictionary: Initialize the dictionary with all single character strings;
old: First dictionary index from input;
while *input stream is non empty* do
new: next dictionary index from input;
if *String dictionary has new* then
S = translation of new;
else
S = translation of new;
S = S + C.
end
output the string S;
assign the first character of string S to C;
add old + C to the String dictionary;
old = new;
end

3.3 RSA Encryption/Decryption

In this article we use the classical RSA algorithm [15] for encrypting and decrypting the compressed data from LZW compressor. This has three stages key generation, encryption and decryption.

3.3.1 Key generation

The receiver shall generate two keys namely the public and the private key. The public key is encrypted using a symmetric key with XOR encryption and is send to the transmitter. The private key is kept secret by the receiver, so as to use it for decryption. The process of generation of public key and private key is described below.

Select two distinctive and large prime number 'a', 'b' and calculate the values of 'n', ψ(n), 'e', 'd' as in (9), (10), (11) and (12) respectively.

$$n = a * b \quad (9)$$

$$\psi(n) = (a-1) * (b-1) \quad (10)$$

$$\gcd(\psi(n), e) = 1 \text{ and } 1 < e < \psi(n) \quad (11)$$

$$d * e \equiv 1 \bmod \psi(n) \quad (12)$$

The Public and Private keys are {n, e} and {n, d} respectively.

3.3.2 Encryption

The sender takes the result from LZW compressor and encrypts it using the public key. The algorithm uses modular exponentiation method for encryption. The cipher 'c' is calculated using (13).

$$c = m^e \pmod n \quad (13)$$

Where cipher is 'c', message is 'm' and public key is {n, e}.

3.3.3 Decryption

The receiver takes the result from LZW de-compressor and decrypts it using the secret private key. The same modular exponentiation method is used for decryption. The message 'm' is calculated using (14).

$$m = c^d \pmod n \quad (14)$$

Where cipher is 'c', decrypted message is 'm' and secret private key is {n, d}.

Table-2. Range table for PVD.

Range	[0,15]	[16,31]	[32,63]	[64,127]	[128,255]
Bits (t)	3	4	5	6	7

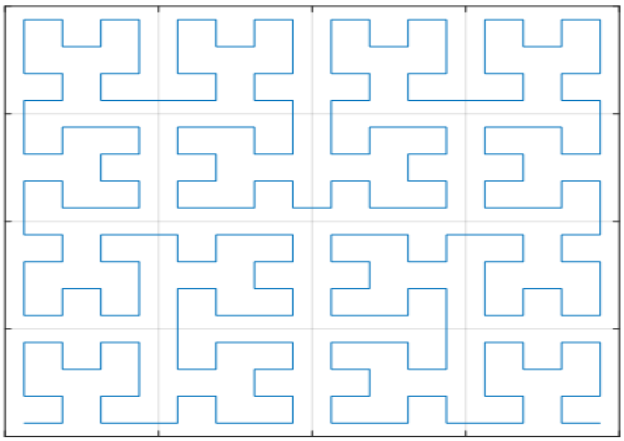

Figure-4. Hilbert fractal order.

3.4 Proposed PVD Algorithm

The cover image is first divided into non intersecting blocks of size 1x2 pixels. The selection and traversal of these blocks is given by a Hilbert fractal. The order of Hilbert fractal is given in the equation (15)

$$n = \lceil \log_2 w \rceil \quad (15)$$

Where n is the Hilbert order and width of the image is denoted by 'w'. Figure-2 shows the traversals in second and third order Hilbert fractals [16]. The fractal starts from the bottom left corner of the image and traverses through all the pixels in a single-color plane.

The embedding procedure of this system is depicted below. The difference between the two pixels denoted by 'd' is calculated using (16).

$$d = |g_1 - g_2| \quad (16)$$

Table-2 is the PVD range table; the pixel difference d matches to a range R_i in the range table. Now l and t are calculated where l is the lower bound of R_i with t bit embedding capacity. Then t bits of the input stream of secret data is converted into decimal number denoted by 's'. The new difference d' is calculated using (17) and a new value m is calculated using (18)

$$d' = l + s \quad (17)$$

$$m = | d' - d | \quad (18)$$

The new modified values of the pixels are calculated using (19)

$$(g_1^{'},g_2^{'})=\begin{cases}\left(g_1+\left\lceil\frac{m}{2}\right\rceil,g_2-\left\lfloor\frac{m}{2}\right\rfloor\right)_{\text{if } g_1\geq g_2 \text{ and } d^{'}>d}\\ \left(g_1-\left\lceil\frac{m}{2}\right\rceil,g_2+\left\lfloor\frac{m}{2}\right\rfloor\right)_{\text{if } g_1\geq g_2 \text{ and } d^{'}\leq d}\\ \left(g_1-\left\lceil\frac{m}{2}\right\rceil,g_2+\left\lceil\frac{m}{2}\right\rceil\right)_{\text{if } g_1<g_2 \text{ and } d^{'}>d}\\ \left(g_1+\left\lceil\frac{m}{2}\right\rceil,g_2-\left\lfloor\frac{m}{2}\right\rfloor\right)_{\text{if } g_1<g_2 \text{ and } d^{'}\leq d}\end{cases} \quad (19)$$

In order to avoid the FOPB (Fall out of boundary problem) the modified pixels are again subjected to the equation (20)

$$(g_1', g_2')=\begin{cases}\left(255, g_2'-g_1^{'}+255\right)_{\text{if } g_1'>255}\\ \left(g_1'- g_2'+255,255\right)_{\text{if } g_2'>255}\\ \left(0, g_2'-g_1^{'}\right)_{\text{if } g_1'<0}\\ \left(g_1'-g_2^{'}, 0\right)_{\text{if } g_2'<0}\end{cases} \quad (20)$$

The secret message is extracted from the stego image as described below. The stego image is divided into block of 2 pixels each using Hilbert fractal same as in the embedding process.

The difference between the two pixels denoted by 'd' is calculated using (21)

$$d=\left|g_1'-g_2'\right| \quad (21)$$

The difference 'd' is used to calculate the values of 'l' and 't' from the range table. Equation (22) is used to calculate the secret data 's' present in the stego block.

$$s = d - l \quad (22)$$

The s in decimal is then converted to t bit binary and is appended to the output stream.

4. RESULTS AND DISCUSSIONS

This section describes the results of the proposed method. MATLAB and Python are used to test and implement our system. The images for testing the system are available at [19].

The images are in grayscale with a size of 512x 512 pixels. Figure-7 shows three cover images (Lena, Pepper and Boat) and their stego images respectively in two rows. Figure-5 shows the (Pixel Difference Histogram) PDH analysis of above three images. The plot for number of bits in key vs the probability of Eave's dropper being undetected in QKD is illustrated in Figure-6. The un-detection probability is given by (23) as follows

$$P(\text{un-detection})=0.75^n \quad (23)$$

If we use a 100-bit symmetric key obtained from QKD then the probability of Eave's dropper being undetected is 3.2072×10^{-13} , which is nearly zero i.e., the probability of detecting the Eave's dropper is close to 1.

Table-3. Performance metrics of proposed system.

Image	Entropy	NPCR	PSNR	MSE	Q	Capacity(bits)	Bpp
Lena	7.451443	0.761707	42.064787	4.042049	0.999122	1273296	4.857239
Mandril	7.374285	0.815571	36.076562	16.048119	0.995553	1528992	5.832642
Pepper	7.601174	0.765827	40.990180	5.176811	0.999111	1275488	4.865601
Jet	6.736727	0.752949	40.650830	5.597542	0.998707	1288952	4.916962
Tank	6.402966	0.769512	42.367051	3.770294	0.997439	1272408	4.853851
Truck	6.585865	0.765163	42.479981	3.673519	0.997509	1271536	4.850525
Airplane	5.475328	0.748657	42.761283	3.443119	0.996517	1242016	4.737915
Boat	7.218194	0.773438	40.269269	6.111580	0.998603	1312528	5.006897
Average	6.855748	0.769103	40.957493	5.982879	0.997820	1308152	4.990204

Table-4. Comparision of existing schemes with our proposed scheme.

Image	Hussain *et al.* [11]			A.K. Shukla *et al.* [5]			Proposed method		
	Capacity (bits)	Bits/pixel (bpp)	PSNR (dB)	Capacity (bits)	Bits/pixel (bpp)	PSNR (dB)	Capacity (bits)	Bits/pixel (bpp)	PSNR (dB)
Lena	800673	3.05	35.76	994403	3.79	37.32	1273296	4.86	42.06
Mandril	825881	3.15	33.57	1131700	4.32	33.18	1528992	5.83	36.08
Pepper	798636	3.04	35.64	992187	3.78	37.47	1275488	4.87	40.99
Jet	795304	3.03	35.99	1003598	3.83	36.83	1288952	4.92	40.65
Tank	865093	3.30	35.84	107066 1	4.08	36.70	1272408	4.85	42.37
Truck	865094	3.30	35.92	1071526	4.09	36.65	1271536	4.85	42.48
Airplane	865093	3.30	34.07	103826 1	3.96	37.38	1242016	4.74	42.76
Boat	865094	3.30	34.03	1091587	4.16	35.51	1312528	5.01	40.27
Average	835108	3.18	35.10	1049240	4.00	36.38	1308152	4.99	40.96

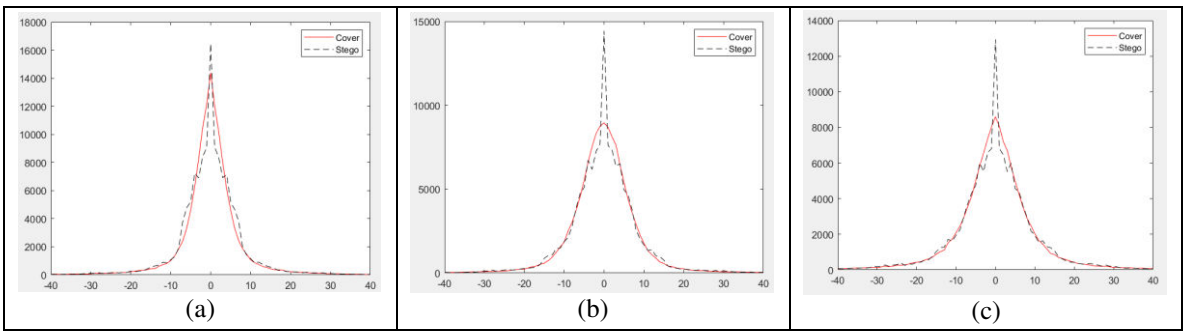

Figure-5. PDH analysis for proposed system: (a) Lean (b) Pepper (c) Boat.

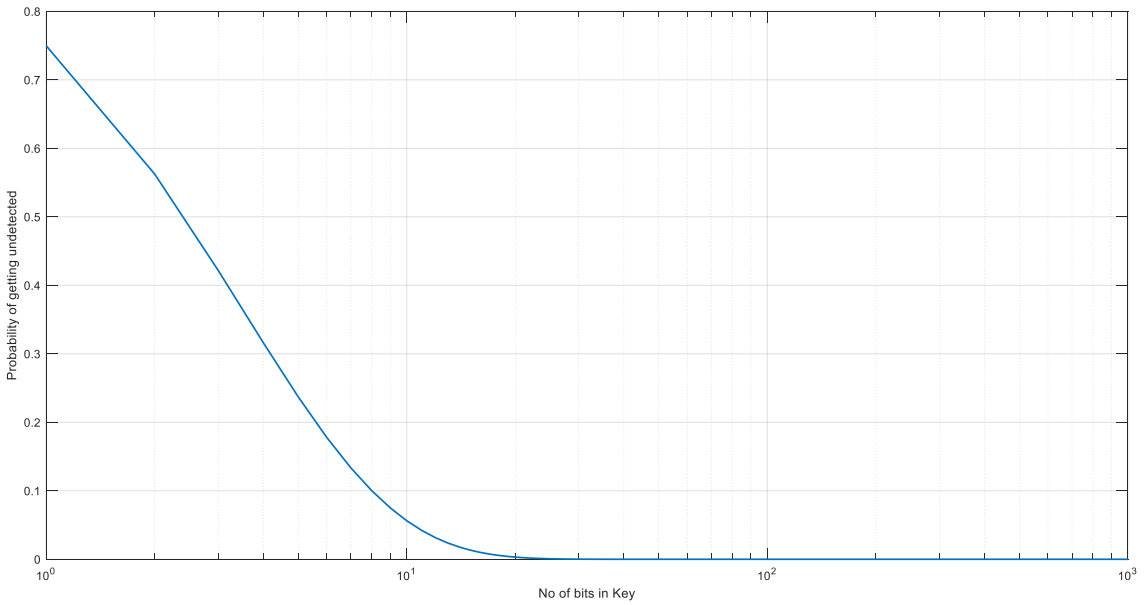

Figure-6. Number of bits in key vs Probability of undetected in QKD.

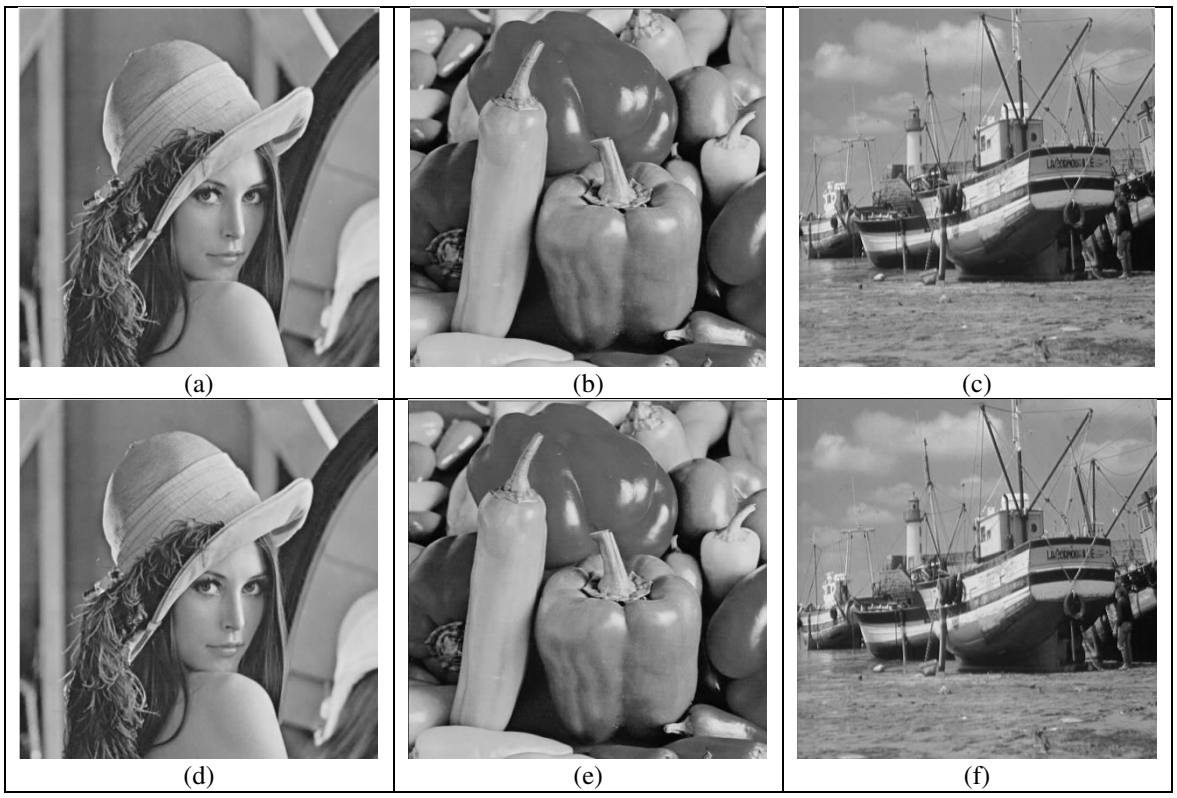

Figure-7. Cover Images: (a)Lena (b)Pepper (c)Boat; Stego Images: (d)Lena (e)Pepper(f)Boat.

Table-3 describes the performance metrics of the system for different cover images. The secret data is a text file of size 2MB and taken from [17]. The comparison of performance metrics like Capacity (bits), Bits/pixel and PSNR of our system with Hussain *et al*. [11] and A. K. Shukla *et al.* [5] is depicted in Table 4. The proposed work achieved better PSNR values while embedding the cover image with more data. The proposed scheme has an average capacity 1308152 bits which is 473044 bits more than Hussain *et al.*'s. [11] and 258912 bits more than A.K. Shukla *et al*.'s. [5]. An average bits/ pixel value of 4.99 is achieved by our system which is 1.81 bits more than Hussain *et al*.'s. [11] and 0.99 bits more than A.K. Shukla *et al.* [5]. A PSNR (average) value of 40.96 dB is achieved by the proposed scheme which is 5.86 dB more than Hussain *et al*.'s. [11] and 4.58 dB more than A.K. Shukla et al.'s [5].

5. CONCLUSIONS

In this paper we proposed a highly secure and high capacity PVD steganography scheme in which LZW compression is used to increase embedding capacity and QKD is used to secure key distribution of RSA key. QKD can detect the presence of third party (Eavesdropper) who is trying gain knowledge about secret key.

PVD and LZW together achieved an increase in the embedding capacity of about 24.6% when compared with existing methods. The system is resilient to quantum computing and steganalysis like PDH. Thus, the proposed system achieves higher security and capacity than existing schemes.

REFERENCES

[1] Hussain M., Wahab A. W., Idris Y. I., Ho A. T. S. and amp Jung K.-H. 2018. Image steganography in spatial domain: A survey. Signal Processing: Image Communication, 65, 46-66. https://doi.org/10.1016/j.image.2018.03.012

[2] Kalaichelvi V., Meenakshi P., Vimala Devi P., Manikandan H., Venkateswari P. and Swaminathan S. 2020. A stable image steganography: A novel approach based on modified RSA algorithm and 2-4 least significant bit (LSB) technique. Journal of Ambient Intelligence and Humanized Computing, 12(7): 7235-7243. https://doi.org/10.1007/s12652-020-02398-w

[3] Nan Li. 2010. Research on Diffie-Hellman Key Exchange protocol. 2010 2nd International Conference on Computer Engineering and Technology. https://doi.org/10.1109/iccet.2010.5485276

[4] Bhatia V. and amp Ramkumar K. R. 2020. An efficient quantum computing technique for cracking RSA using Shor's algorithm. 2020 IEEE 5th International Conference on Computing

Communication and Automation (ICCCA). https://doi.org/10.1109/iccca49541.2020.9250806

[5] Shukla A. K., Singh A., Singh B. and Kumar A. 2018. A secure and high-capacity data-hiding method using compression, encryption and optimized pixel value differencing. IEEE Access, 6, 51130-51139. https://doi.org/10.1109/access.2018.2868192

[6] D.-C. Wu and W.-H. Tsai. 2003. A steganographic method for images by pixel-value differencing. Pattern Recognition. Lett. 24: 1613-1626.

[7] H.-C. Wu, N.-I. Wu, C.-S. Tsai and M.-S. Hwang. 2005. Image steganographic scheme based on pixel-value differencing and LSB replacement methods. IEE Proc.-Vis., Image Signal Process. 152(5): 611-615.

[8] C.-H. Yang, C.-Y. Weng, S.-J. Wang and H.-M. Sun. 2010. 'Varied PVD+LSBevading detection programs to spatial domain in data embedding systems. J. Syst. Softw. 83(10): 1635-1643.

[9] M. Khodaei and K. Faez. 2012. New adaptive steganographic method usingleast significant- bit substitution and pixel-value differencing. IET Image Process. 6(6): 677-686.

[10] Lee Y. P., Lee J. C., Chen W. K., Chang K. C., Su J. and Chang C. P. 2012. High-payload image hiding with quality recovery using tri-waypixel-value differencing. Information Sciences. 191, 214-225.

[11] M. Hussain, A. W. A. Wahab, N. Javed, and K.-H. Jung. 2018. Recursiveinformation hiding scheme through LSB, PVD shift, and MPE. IETETech. Rev. 35(1): 53-63.

[12] Khashandarag A. S., Navin A. H., Mirnia M. K. and amp Agha Mohammadi H. H. 2011. An optimized color image steganography using LFSR and DFT Techniques. Advanced Research on Computer Education, Simulation and Modeling, 247–253. https://doi.org/10.1007/978-3-642-21802-6_40

[13] Nurhadi A. I. and amp Syambas N. R. 2018. Quantum key distribution (QKD) protocols: A survey. 2018 4th International Conference on Wireless and Telematics (ICWT).https://doi.org/10.1109/icwt.2018.8527822

[14] Bharti A., Deep V. and amp Choudhary D. R. 2001. LZW Data Compression: Optimizations & Review. IETE Journal of Education, 42(1-4), https://doi.org/10.1080/09747338.2001.11415744

[15] Rivest R. L., Shamir A. and amp Adleman L. 1978. A method for obtaining digital signatures and public-key cryptosystems. https://doi.org/10.21236/ada606588

[16] Zhang X., Wang L., Zhou Z. Niu Y. 2019. A chaos-based image encryption technique utilizing Hilbert curves and H-fractals. IEEE Access, https://doi.org/10.1109/access.2019.2921309

[17] Lorem Ipsum. Lorem Ipsum - All the facts - Lipsum generator. (n.d.). Retrieved April 9, 2022, from https://www.lipsum.com/

[18] C.-K. Chan and L.-M. Cheng. 2004. Hiding data in images by simple LSB substitution. Pattern Recognition. 37: 469-474.

[19] Ravikumar CV Kalapraveen B. 2016. Performance analysis of HSRP in layer3 for corporate networks. Indian Journal of Science and Technology. 9(20).

[20] Ravikumar C. V., Saranya K. C. 2016. Improoving Interference alignment of Gaussian MIMO x channel and Gaussian MIMO z channel. International Journal of Applied Engineering and Research. 11(9).

[21] Kalapraveen. 2016. Ravikumar CV-Performance analysis of ipv4 to ipv6 transition methods. Indian Journal of Science & Technology. 9(20).

[22] Jayaprabath, Ravi Kumar, C.V. and Rahul Varma C., 2019. Performance analysis of interior and exterior routing protocols. Asian Research Publishing Network (ARPN).

[23] Md. Imaudin, Ravi Kumar, C.V. and Kalapraveen B. 2020. Signal detection in MC-CDMA system using ELM receiver to mitigate MAI and non-linear distrortion. Asian Research Publishing Network (ARPN).

[24] C.V. Ravikumar, Kala Praveen Bagadi. 2017. Receiver design using artificial Neural Network for signal detection in MC-CDMA system. International Journal of Intelligent Engineering & Systems.

[25] Jayaprabath, Rahul Varma C. 2019. Performance analysis of interior and exterior routing protocols. Asian Research Publishing Network (ARPN).

[26] Ravi Kumar, C.V. and Kalapraveen B. 2019. Design of Multilayer Perceptron Receiver for MC-CDMA system to Mitigate Multiple Access Interference and Non-linear Distortion. Neural Computing and Applications. 31(S-2): 1263-1273.