Steganography Method Using Effective Combination of RSA Cryptography and Data Compression

Abhranil Bose
Electronics and Commnication Engineering, School of Electronics Engineering
Vellore Institute of Technology
Vellore 632014, Tamil Nadu, India
abhranil.bose2019@vitstudent.ac.in

Akshat Kumar
Electronics and Commnication Engineering, School of Electronics Engineering
Vellore Institute of Technology
Vellore 632014, Tamil Nadu, India
akshat.kumar2019@vitstudent.ac.in

Prof. Malaya Kumar Hota
Department of Communication Engineering, School of Electronics Engineering
Vellore Institute of Technology
Vellore 632014,Tamil Nadu, India
ORCID_ID: 0000-0002-3669-1611

Shubham Sherki
Electronics and Commnication Engineering, School of Electronics Engineering
Vellore Institute of Technology
Vellore 632014, Tamil Nadu, India
shubham.sherki2019@vitstudent.ac.in

2022 First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT) | 978-1-6654-3647-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICEEICT53079.2022.9768402

***Abstract*—In this modern era, technology is increasing at a rapid rate and making new developments. The growing use of the Internet amongst people around the world and availability of digital data and its sharing has driven researchers and professionals to pay particular attention to information security. Data security is of great need. The emergence of digital media has taken this to a complete new level. Users frequently need to store, send, or receive private information and the information needs to be protected against unauthorized access and attacks from intruder. The process of hiding data and information is known to be steganography it is done to provide information security. By using steganography, one can hide thousands of words even in an average sized image. Cryptography method enhances the security level. This proposed paper uses combination of RSA cryptography, data compression and Hash-LSB steganography technique to make the data secure. RSA cryptography algorithm is implemented as receiver will have to use private key to decrypt message because the confidential message has been encrypted using public key. The secret message is compressed using Huffman coding algorithm and the cover image is compressed using Discrete wavelet transform (DWT). The cover image lossy compression is to reduce the cover image's dimensions. Then Hash LSB algorithm is applied to the RSA encrypted data to put it into the cover image. The proposed algorithm is implemented in MATLAB. At last PSNR, SSIM, Compression ratio and MSE are calculated. Our proposed method is secure for digital data transmission and also reduces the time of sending data over the Internet.**

Keywords—Steganography, Cryptography, data compression, Huffman coding, DWT, Hash-LSB, LSB, MSE, PSNR, SSIM, RSA

I. INTRODUCTION

Data compression, encryption and hiding techniques are used individually as well as in combination of techniques in various fields and applications. These are currently being deployed so as to achieve multiple objectives such as to protect and enhance the security of information being transmitted, reduce bandwidth and complexity of hardware required for transmission of data, effectively hide secret information without getting attention from unwanted viewers etc. Hence, efficient algorithms to achieve these objectives are needed.

Secret messages can be sent by hiding it in a cover image so that apart from the sender and receiver so one else can obtain the secret information. Because of the simplicity and effectiveness of concealing data, LSB steganography technique is commonly used. Using LSB steganography technique the data can be hidden in such a way that a naked eye can't detect the hidden information present in the cover image. An improvised version of LSB steganography used i.e. Hash-LSB Steganography.

Compressing, compacting and removing the redundancy of data from an information without distorting the information safely is very essential. Different algorithms and methods are available to achieve this. Based on our requirement both lossy or lossless compression can be implemented. Since we want to convey the secret information to the receiver without corrupting or losing it we perform lossless text compression on text while on the cover image which is not of much importance to the receiver we perform lossy image compression. Huffman compression being simple and robust, is under lossless data compression that can be used for text compression. DWT compression is under lossy data compression which can be used to compress the size of cover image to achieve high compression ratio and degrades the quality of image insignificantly.

The process involving encryption of data using algorithms would provide an extra layer of security and safety so that for an unwanted recipient the information is meaningless and unreadable. For this we apply RSA Cryptography, it is based on the difficulty of factoring large integers.

A. RSA Encryption

There are 2 ways to encrypt your data, i.e. there are two types of cryptographic algorithm, broadly categorised as symmetric and asymmetric key algorithm. In a symmetric key algorithm the same key is used to encrypt and decrypt the data, while in the case of an asymmetric key algorithm, 2 different keys are used to encrypt and decrypt the data. The public key is distributed to all the senders while the private key is known

only to a specified user. RSA algorithm named after the last name of these MIT scholars who developed this concept. They are Ronald Rivest , Adi Shamir and Leonard Adleman.

In order to use this method i.e. to encrypt the data, the sender needs to choose 2 prime numbers, p and q, generally large and random to increase the complexity.

- Compute n, the product of the two prime numbers p and q ($n = p \times q$).
- Compute $\Phi(n) = (p-1)(q-1)$
- Then a number e is selected such that it is relatively prime to the $\Phi(n)$, also $1<e<\Phi(n)$
- Then a number d is selected such that $e \times d = 1 \bmod (\Phi(n))$, here mod stands for modular operation.
- (e , n) is the public key and this key is known to all.
- (d , n) is the private key and this key is known only to the authorised receiver. d , p , q , and Φ are kept secret.

B. *Hash LSB Steganography*

Steganography is the science of hiding information within an ordinary file/data in order to avoid detection by normal eyes. Using steganography, we can embed a secret message inside a piece of unsuspicious information and send it without anyone knowing of the existence of the secret message. This object is called the cover object or carrier object of the steganographic method. The type of the secret message to be hidden can be audio, image and video. After implementation of the steganography technique, the resultant output data file is known as stego-object.

One of the most well-known steganography techniques studied in the information hiding of digital data is Hash Least Significant Bit modification coding technique. Hash function is used in this technique to determine the position of LSB for hiding the confidential message. Hash function computes the positions of LSB of each RGB pixel; afterwards these text bits are embedded into these RGB pixels independently. In the beginning of this process a secret message is taken and it is inserted into the cover image. Then, the secret text is converted into binary bits. Further, the hash function would select the positions where the bits are to be placed and then 8 bits of message at a time would be embedded in the order of 3, 3, and 2 in the red, green and blue pixel of the cover image respectively.[3] This is continued until the entire message of bits gets embedded inside the cover image.

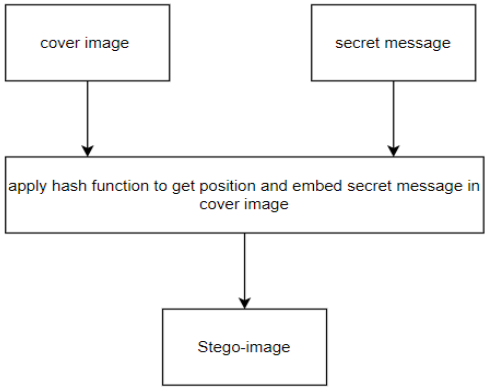

Fig. 1. H-LSB Extraction algorithm

In the decoding process, the hash function is used to detect the positions of the LSB's where the data bits were embedded. Then the bits are obtained from the same position and order as were previously embedded. Finally, the message obtained from the above step would be in binary which is then later converted into text, and thus, the secret text message is obtained.

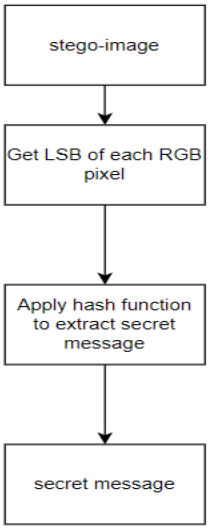

Fig. 2. H-LSB Decryption algorithm

C. *DWT Image Compression*

Discrete wavelet Transform(DWT) converts the image into a series of 2-D wavelets. These wavelets can be stored more efficiently than the pixels of an image. In 1-Dimensional DWT, the data is divided into two parts i.e. data containing high frequency components and data containing low frequency components.

In DWT various types of wavelets and filters can be implemented. One of them, 'Haar' wavelet, is the most simplest and commonly used wavelet. To implement lossy image compression using DWT, Image is decomposed into four sub bands i.e. LL-LH-HL-HH when 1-level decomposition of DWT is performed in image. These sub-bands are coefficients of DWT. Then a threshold is decided. The coefficients which are below this threshold are set to zero. The signal(data) component is accurately approximated to the following components: The approximation coefficients at a suitably chosen level and the detail coefficients from 1 to the chosen level.

The algorithm of DWT compression are as follows:

- The image is decomposed by DWT into N levels using a suitable wavelet(filter).
- A threshold for each level is selected, then by soft or hard thresholding is applied on to the detailed coefficients.
- The image is then reconstructed using original approximation coefficients of level N and the altered detailed coefficients from level 1 to N using Inverse DWT.

D. *Huffman Coding*

Huffman coding is implemented to achieve lossless data compression. It converts input characters into variable-length codes. The length of codes assigned to a character depends on the number of occurrences of the character in a text. The most occurring character has the smallest code length and the least occurring character has the largest code. These codes are assigned to the characters in such a way that none of the code set to a character is not the prefix of code to any other character's code. This method creates a Huffman Tree. Then after, it assigns codes to the characters transverse to the Huffman Tree. Huffman Tree uses the bottom-up approach.

The Huffman coding of a text is found as follows

- The probabilities of each character is found, then these characters are arranged in descending order with their probability values. This is the last level of the Huffman tree.
- Then, the sum of two smaller probabilities is computed and all remaining probabilities are kept under it. This is the second last level of the Huffman tree. Then subsequent lower levels are built using this same process until we get one probability which is level 0.
- Then the edges of the Huffman tree are assigned 0 and 1.

II. Proposed Algorithm

The proposed algorithm in our case is encrypting our secret message using RSA method combined with Huffman lossless data compression, while using DWT lossy data compression method on the cover image in order to reduce the information's bit in steganography. Two key processes are used in embedding and extraction of the secret message.

A. *Embedding Algorithm*

- First the secret message is processed and then two suitable prime numbers are taken as per the RSA algorithm.
- The encrypted message and two keys (private and public key) are generated.
- Next, the encrypted message is compressed using the Huffman Coding.
- Next, the cover image is decomposed using DWT in order to decompose the image into four sub-bands which are LL, LH, HL, and HH.
- Using the Hash-LSB technique, an encrypted message is embedded in the cover image producing a compressed stego-image.
- The last step calculates MSE, PSNR and SSIM.

B. *Extracting Algorithm*

- The encrypted message is extracted from the stego-image using the Hash-LSB method.
- Then, the extracted message is decompressed using the Huffman Decoding algorithm.
- Finally, the retrieved encrypted message is decrypted using the RSA algorithm with the private key.

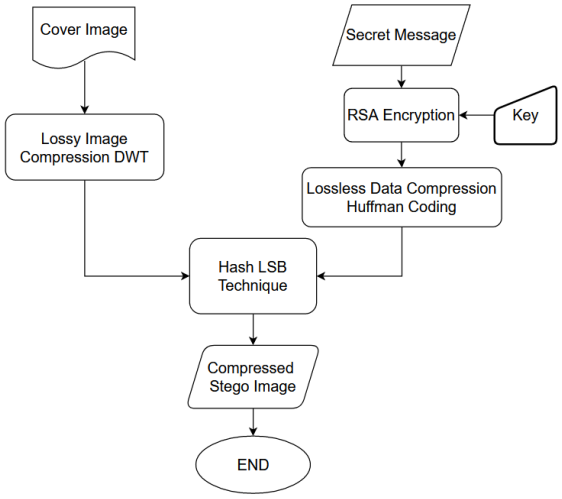

Fig. 3. Embedding algorithm

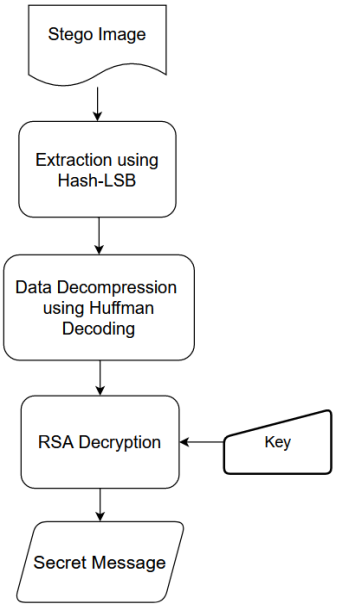

Fig. 4. Extracting algorithm

III. Performance Analysis

There are Several criteria that can be used to evaluate the performance of the proposed algorithm, some of the parameter are mentioned below:

A. *Mean Squared Error (MSE)*

MSE helps in determining the difference between the compressed image data and the original image data. It is used mainly to analyze the quality of our image. It should be as less as possible where if it is 0, it means that the compressed and the original image are similar.

$$MSE = \frac{1}{MxN}\sum_{X=1}^{M}\sum_{Y=1}^{N}(f(X,Y) - f_0(X,Y))^2 \quad (1)$$

Where f (x, y) is the original input image, f0 (x, y) is compressed image, and M, N are the dimensions of the images.

B. Peak Signal to Noise Ratio (PSNR)

This is the ratio between the signal strength and the noise that appears in the signals. It all depends on the quality of the image. The higher the PSNR, the higher the image quality. It depends on the MSE of the selected image.

$$PSNR = 10\log_{10}\left(\frac{MAX^2}{MSE}\right) \quad (2)$$

Where MAX^2 is 255^2 is the maximum intensity of pixels in the ideal image and MSE is mean squared error.

C. Structural Similarity Index (SSIM)

SSIM measures the perceptual difference between two identical images. The Structural Similarity Index (SSIM) is a perceptual metric that quantifies image quality degradation caused by processing such as data compression or by losses in data transmission.

$$SSIM(x,y) = \frac{(2\mu_x\mu_x+C_1).(2\sigma_x\sigma_y+C_2)}{(\mu_x^2+\mu_y^2+C_1).(\sigma_x^2+\sigma_y^2+C_2)} \quad (3)$$

where μx and μy are defined by illuminating each image in the x and y directions, σx and σy are the standard deviations and the contrast estimation of the signal, and C1 and C2 are very small constants.

IV. Experimental Results

Both the proposed methodology and the methodology presented in [1] has been implemented in MATLAB R2018a software and both the result with a given fixed message length. PSNR, MSE and SSIM have been computed for each methodology for different cover images and tabulated below.

In this paper, the we have used standard color image of several formats (like jpeg, png, tiff, and bmp) to hide a text in cover image hence a high-quality stego-image is created.

A. Proposed Algorithm Result

TABLE I , TABLE II , TABLE III described below depicts the MSE, PSNR, SSIM values computed for five cover images after our proposed algorithm was implemented on each of those cover images with concealing message length of 30, 250, 1000 characters respectively.

TABLE I. MESSAGE LENGTH OF 30 CHARACTERS

Cover Image	MSE	PSNR	SSIM
lena.tif	0.0383	62.3039	1
peppers.tiff	0.0366	62.4911	1
baboon.jpg	0.0236	64.4005	1
tulips.png	0.0249	64.1656	0.9999
Average	0.0308	63.3403	1

TABLE II. MESSAGE LENGTH OF 30 CHARACTERS

Cover Image	MSE	PSNR	SSIM
lena.tif	0.3686	52.4648	0.9999
pepper.tiff	0.3732	52.4100	0.9999
baboon.jpg	0.2475	54.1942	0.9999
tulips.png	0.2479	54.1877	0.9995
Average	0.3093	53.3141	0.9998

TABLE III. MESSAGE LENGTH OF 30 CHARACTERS

Cover Image	MSE	PSNR	SSIM
lena.tif	0.0383	62.3039	1
peppers.tiff	0.0366	62.4911	1
baboon.jpg	0.0236	64.4005	1
tulips.png	0.0249	64.1656	0.9999
Average	0.0308	63.3403	1

B. Existing Algorithm in [1] Result

TABLE IV, TABLE V, TABLE VI described below depicts the MSE, PSNR, SSIM values computed for five cover images after existing algorithm in [1] was implemented on each of those cover images with concealing message length of 30, 250, 1000 characters respectively.

TABLE IV. MESSAGE LENGTH OF 30 CHARACTERS

Cover Image	MSE	PSNR	SSIM
lena.tif	0.4145	51.9551	0.9998
peppers.tiff	0.4085	52.0194	0.9998
baboon.jpg	0.4177	51.9223	0.9997
tulips.png	0.3880	52.2423	0.9992
Average	0.4072	52.0348	0.9996

TABLE V. MESSAGE LENGTH OF 30 CHARACTERS

Cover Image	MSE	PSNR	SSIM
lena.tif	0.5497	50.7292	0.9998
pepper.tiff	0.5440	50.7744	0.9998
baboon.jpg	0.5558	50.6819	0.9997
tulips.png	0.4770	51.3457	0.9991
Average	0.5316	50.8828	0.9996

TABLE VI. MESSAGE LENGTH OF 30 CHARACTERS

Cover Image	MSE	PSNR	SSIM
lena.tif	0.9601	48.3077	0.9997
pepper.tiff	0.9627	48.2959	0.9996
baboon.jpg	0.9928	48.1622	0.9996
tulips.png	0.7643	49.2981	0.9987
Average	0.9200	48.5410	0.9994

C. Comparison Between Existing Algorithm [1] and Proposed Algorithm

TABLE VII. TABULATED COMPARISON BETWEEN EXISTING ALGORITHM [1] AND PROPOSED ALGORITHM

Cover Image	Message length (characters)	MSE	PSNR	SSIM
In [1]	30	0.4072	52.0348	0.9996
Proposed	**30**	**0.0308**	**63.3403**	**1**
In [1]	250	0.5316	50.8828	0.9996
Proposed	**250**	**0.3093**	**53.3141**	**0.9998**
In [1]	**1000**	**0.9200**	**48.5410**	**0.9994**
Proposed	1000	1.1616	47.5668	0.9993

Even after using lossy image compression technique (DWT compression), the stego-image produced still looks similar to original cover image. RSA cryptography gives extra security to our secret message to be embedded to cover image.

From TABLE VII it is clear that our proposed algorithm works very well for small (30) and moderate (250) character length messages i.e. it achieves a higher PSNR, higher SSIM and lower MSE values compared to existing algorithm in [1]. This implies that we obtain a higher quality stego images and higher level of similarity exists between cover and stego images. However, for a very large character length message, it fails to achieve a better PSNR, SSIM and MSE values than in [1].

Hence, our proposed algorithm can be used for small to moderate character length messages to produce high quality stego image.

V. Conclusion

The experimental result indicates that the proposed mechanism has higher PSNR and SSIM as well as lower MSE i.e. it gives better performance especially in visual quality when the message length is small to moderate in comparison to other existing methods but for large message length, [1] is proven to be the best algorithm. Our algorithm which uses a combination of RSA cryptography, DWT compression and Huffman coding is already proven in [1] to be very effective in providing high security, accepting durability against attacks and high storage capacity.

References

[1] O. F. A. Wahab, A. A. M. Khalaf, A. I. Hussein and H. F. A. Hamed, "Hiding Data Using Efficient Combination of RSA Cryptography, and Compression Steganography Techniques," in IEEE Access, vol. 9, pp. 31805-31815, 2021, doi: 10.1109/ACCESS.2021.3060317.

[2] R.L. Rivest, A. Shamir, and L.M. Adleman, "A Method for Obtaining Digital Signature and Public-key Cryptosystems," Communications of the ACM, Vol. 21, No. 2, pp. 120–126, 1978.

[3] Ms.Shridevishetti, Mrs.Anuja S, 2015, A Secure Image Steganography based on RSA Algorithm and Hash-LSB Technique, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) ICESMART – 2015 (Volume 3 – Issue 19)

[4] Mozammel, M & Chowdhury, Hoque & Khatun, Amina. (2012). Image Compression Using Discrete Wavelet Transform. International Journal of Computer Science Issues. 9.

[5] M. H. Abood, "An efficient image cryptography using hash-LSB steganography with RC4 and pixel shuffling encryption algorithms," 2017 Annual Conference on New Trends in Information & Communications Technology Applications (NTICT), 2017, pp. 86-90, doi: 10.1109/NTICT.2017.7976154.