Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

The clandestine image transmission scheme to prevent from the intruders

Saad Al-Mutairi [1], *, S. Manimurugan [2]

[1]Computer Science and Information Technology Faculty, Tabuk University, Tabuk, Saudi Arabia
[2]College of Computing and Informatics, Saudi Electronics University, Riyadh, Saudi Arabia

ARTICLE INFO

Article history:
Received 10 October 2016
Received in revised form
3 February 2017
Accepted 3 February 2017

Keywords:
Reverse matrix encode Alpha-encryption
Hill cipher
Secret image
Shuffle process

ABSTRACT

The main aim of this technique is to prevent the capture of secret data during the exchange/transfer between the source and destination, without the knowledge of intruder. A secret message is represented as an image, and the same image is encrypted by the Triple Encryption (TE) technique. This can be divided into three types of processes. These processes are Reverse Matrix (RM) encode, Alpha-Encryption (AE) and Hill Cipher (HC) encryption. In the RM process, the original secret image is hidden within a cover image. The encoded cover image pixels are converted as characters using a lookup table in the AE process. The encrypted cipher characters are once again encrypted in the HC encryption process. The main advantage of this proposed technique is that, to strengthen the proposed technique the pixels positions are interchanged in every stage. After the encryption, the cipher's data are sent to the destination for the reconstruction process. The Triple Decryption (TD) technique is divided into three processes of: Inverse Matrix (IM) decode, Alpha-Decryption (AD) and HC decryption. The received cipher data are decrypted by the HC decryption and AD processes. Finally, the decrypted data is decoded by the IM decode process. In this process, the original secret and cover images are obtained. In addition, to measure the performance of the proposed algorithm, the standard parameters are considered. The results show that the proposed algorithm provided a high Peak Signal to Noise Ratio (PSNR), good Correlation Coefficient (CC), minimum execution time and high CIA (Confidentiality, Integrity and Authentication) compared to existing encryption methods. Therefore, using this proposed method, we can obtain 100% of the original image and the secret data can be prevented from interception by intruders/third parties.

1. Introduction

Image/information encryption is a one of a number of important techniques where secret data can be transferred in a safe and secure manner via a public network. In connection with this, many authors have introduced different encryption methods to hide secret data within other data. In this section we would like to discuss the different methods and their performance proposed by a number of authors. Hu (2003) has proposed an image hiding scheme of hiding multiple grey-level images within another grey-level cover image. This method was introduced to reduce the volume of secret images. The vector quantization scheme was employed to encode the secret images (Hu, 2003). Wu and Tsai (2000) proposed a method to embed a secret image into a cover image. This method was based on the similarity among grey values of consecutive image pixels, as well as the variation of human visual insensitivity from smooth to contrastive.

Experiments found that the peak values of signal-to-noise ratios of the method were high and the resulting stego-images were imperceptible (Wu and Tsai, 2000). Haiping et al. (2013) proposed a novel method of a blind, colour image information-hiding algorithm based on grey prediction to hide the image. This algorithm compresses the secret image based on the improved grey prediction model and it chooses blocks of rich texture in the cover image as the embedding regions using DGRA (Double-dimension Grey Relational Analysis). After these processes, it adaptively embeds the compressed stream of secret bits into the DCT domain mid-frequency coefficients, which were decided by those blocks' DGCD (Double-Dimension Grey Correlation Degree) and HVS (Human Visual System).

* Corresponding Author.
Email Address: s.almutairi@ut.edu.sa (S. Al-Mutairi)

https://doi.org/10.21833/ijaas.2017.02.010
2313-626X/© 2017 The Authors. Published by IASE.

Experimental results show that, the proposed algorithm was robust against Gaussian noise and JPEG compression (Haiping et al., 2013). Zhang et al. (2016) introduced a reversible, lossless and combined data hiding schemes for cipher text images. In the lossless method the cipher text pixels were replaced with new values to embed the additional data into several of the least significant bit planes of cipher text pixels by multilayer wet paper coding (Zhang et al., 2016). Qian and Zhang (2016) proposed a data-hiding scheme where the content owner, using a stream cipher, encrypted the original image and the data-hider compresses a series of selected bits taken from the encrypted image to make room for the secret data. On the other side of the receiver, the secret bits could be extracted using an embedded key. Therefore, in the proposed scheme a key played a vital role between sender and receiver. Ma et al. (2013) proposed reversible data hiding (RDH) in encrypted images by reserving room before the encryption (Qian and Zhang, 2016; Ma et al., 2013). Zhou et al. (2016) proposed an image hiding method. This method was a reversible image data hiding scheme over the encrypted domain. Data embedding was achieved through a public key modulation mechanism, in which access to the secret encryption key was not needed. On the other end, the powerful two-class SVM classifier was designed to distinguish encrypted and non-encrypted image patches, allowing us to jointly decode the embedded message and the original image signal (Zhou et al., 2016).

Qin et al. (2014) have proposed a joint data-hiding and compression scheme for digital images using side match vector quantization (SMVQ) and image in painting. They claimed that the experimental results were good (Qin et al., 2014). Renza (2016) proposed a method for image hiding in an IEEE transaction. His work was based on digital image watermarking. Renza (2016) proposed an algorithm that consists of an insertion over an image of a text string, previously modified by permutation, using a random key and an OVSF (Orthogonal Variable Spreading Factor) generator. The insertion was made in the wavelet domain and it uses QIM (Quantization Index Modulation). The robustness of the proposed algorithm was evaluated by several attacks on the marked image (Renza, 2016).

Nikolaidis (2015) has proposed a data hiding technique in JPEG images. This technique was the modification of zero quantized coefficients in each image block, in contrast to most previously proposed methods, which also affects the non-zero coefficients and/or the quantization tables. Both embedding and extraction were performed on a per-block basis, without the need of a pre-process for the whole image (Nikolaidis, 2015).

Li et al. (2015) proposed a reversible data hiding of encrypted images. Their proposed work was the partition of the encrypted image into two sets; only one set was used for data embedding. The full embedding strategy was employed. The corresponding new fluctuation measurement was designed for the full embedding strategy (Li et al., 2015). Zhang and Zhang (2014) introduced semantic image compression with a hiding technique. The compression creates a compact image by gathering part of the pixels in the original image, and it estimates errors of the remaining pixels. After the process, a compressed image was produced by embedding the estimated errors into the compact image using data hiding techniques (Zhang and Zhang, 2014).

Kwon (2014) and Wu et al. (2015) proposed an innovative image hiding technique in 2014 and 2015. Kwon (2014) proposed a technique of a modified transmission map based on the HMRF (hidden Markov random field) and EM (expectation-maximization) algorithm. The experimental results confirmed that the proposed algorithm was superior to conventional algorithms in image haze removal. Wu et al. (2015) proposed the contrast of a host image to improve its visual quality. The highest two bins in the histogram are selected for data embedding; histogram equalization could be performed by repeating the process (Kwon, 2014; Wu et al., 2015). Lee et al. (2014) also proposed a lossless data hiding scheme to achieve the goal of hiding secret data into vector quantization (VQ)-compressed images that could be lossless reconstructed after the secret data was extracted in the decoder (Lee et al., 2014).

Ishimaru et al. (2014) extended the previous work of hard-wall imaging, which was related to the historical problem of “Poisson Spot” and “Anti-Podal point” (Ishimaru et al., 2014). Xiao and Chen (2014) introduced a separable data hiding scheme for an encrypted image based on compressive sensing. The encoding and decoding were dependent on a key (Xiao and Chen, 2014). Cao and Kot (2013) proposed an image hiding technique using EAG (Edge Adaptive Grid). They also stated that, the proposed method supported state-of-the-art hybrid authentication, which integrates data hiding and modern cryptographic techniques (Cao and Kot, 2013).

In the above statements, many authors have proven different image encryption techniques. However, each method has its own merits and demerits. In this paper, we have proposed a novel secret image encryption technique for secure transmission without the knowledge of intruders/third parties. The entire work of this paper has been divided into seven sections. Section 1 is about the literature review of various conventional image hiding techniques and encryptions. Sections 2 and 3 describe proposed encryption and decryption techniques. Section 4 considers the experimental results, while the conclusion is discussed in section 5. Sections 6 and 7 detail the acknowledgement and references.

2. Proposed triple encryption to encrypt a secret image

In section we outlined the many encryption algorithms that different authors have proposed to encrypt a secret image. Those proposed encryptions are for different applications and situations. However, this paper has proposed the new encryption technique of Triple Encryption (TE) to encrypt a secret image so it can be transmitted over a public network (Almutairi and Manimurugan, 2016).

The TE technique can be classified into three processes. In the first process, the original secret information $\sum_{i=0,j=0}^{m,n} A_{i,j}$ is hidden inside the cover image $\sum_{i=0,j=0}^{m,n} C_{i,j}$ using the Reverse Matrix (RM) encode process. Hiding an original secret image $\sum_{i=0,j=0}^{m,n} A_{i,j}$ into a cover image $\sum_{i=0,j=0}^{m,n} C_{i,j}$ is called encoded image $\sum_{i=0,j=0}^{m,n} D_{i,j}$ in Eq. 1. The encoded image $\sum_{i=0,j=0}^{m,n} D_{i,j}$ is encrypted by the Alpha-Encryption (AE) process using a lookup table. This AE is purely based on the substitution. The encrypted data of the AE is once again encrypted by the Hill Cipher (HC) encryption process, as illustrated in Fig. 1. Finally, the encrypted data is sent to the receiver/authenticated person for the purpose of reconstruction.

$$\sum_{i=0,j=0}^{m,n} A_{i,j} \bowtie \sum_{i=0,j=0}^{m,n} C_{i,j} = \sum_{i=0,j=0}^{m,n} D_{i,j} \qquad (1)$$

2.1. RM Encode process for secret text image

In this encode process, the original secret text image $\sum_{i=0,j=0}^{m,n} A_{i,j}$ and cover image $\sum_{i=0,j=0}^{m,n} C_{i,j}$ are considered as an input. Both images are divided into 4x4 equal subbands in Figs. 2 and 3 (Almutairi and Manimurugan, 2016).

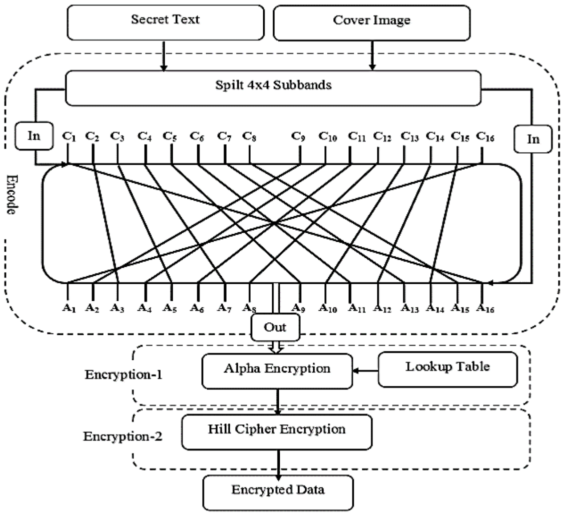


Fig. 1: The triple encryption process

After this process, secret image $\sum_{i=0,j=0}^{m,n} A_{i,j}$ is segregated into 16 subbands. The divided subbands are

$$\sum_{i=0,j=0}^{m,n} A_{i,j}^1, \sum_{i=0,j=0}^{m,n} A_{i,j}^2, \sum_{i=0,j=0}^{m,n} A_{i,j}^3 \ldots \sum_{i=0,j=0}^{m,n} A_{i,j}^{16}.$$

Similarly, the cover image $\sum_{i=0,j=0}^{m,n} C_{i,j}$ is split into

$$\sum_{i=0,j=0}^{m,n} C_{i,j}^1, \sum_{i=0,j=0}^{m,n} C_{i,j}^2, \sum_{i=0,j=0}^{m,n} C_{i,j}^3 \ldots \sum_{i=0,j=0}^{m,n} C_{i,j}^{16}$$

subbands in Eqs. 2 and 3.

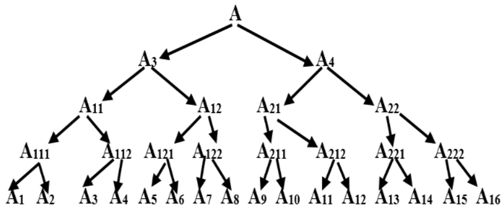


Fig. 2: Divided secret image subbands

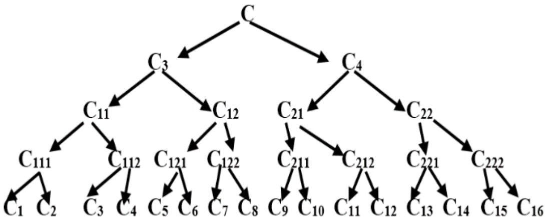


Fig. 3: Divided cover image subbands

The divided secret subbands are encoded into cover image subbands based on the different combinations. This is illustrated in Fig. 4.

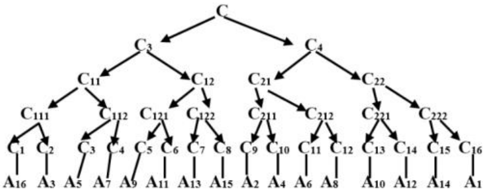

C
C3
C4
C11
C12
C21
C22
C111
C112
C121
C122
C211
C212
C221
C222
C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C12 C13 C14 C15 C16
A16 A3 A5 A7 A9 A11 A13 A15 A2 A4 A6 A8 A10 A12 A14 A1

Fig. 4: Different subbands combinations for the encode process

$$\sum_{i=0,j=0}^{m,n} A_{i,j} = \sum_{i=0,j=0}^{m,n} A_{i,j}^1 \oplus \sum_{i=0,j=0}^{m,n} A_{i,j}^2 \oplus \ldots \oplus \sum_{i=0,j=0}^{m,n} A_{i,j}^{16} \qquad (2)$$

$$\sum_{i=0,j=0}^{m,n} C_{i,j} = \sum_{i=0,j=0}^{m,n} C_{i,j}^1 \oplus \sum_{i=0,j=0}^{m,n} C_{i,j}^2 \oplus \ldots \oplus \sum_{i=0,j=0}^{m,n} C_{i,j}^{16} \qquad (3)$$

During the encode process, every combination of subbands pixels $(\sum_{i=0,j=0}^{m,n} A_{i,j}, \sum_{i=0,j=0}^{m,n} C_{i,j})$ are converted into 8-bit binary values. The least significant bits of the converted cover image subbands (last two bits), are replaced by secret image subbands bits in reverse order. This is given in Eq. 4.

$$\sum_{i=0,j=0}^{m,n} D_{i,j}^1 = \uplus lsp\left\|\sum_{i=0,j=0}^{m,n} C_{i,j}^1 \oplus \sum_{i=0,j=0}^{m,n} A_{i,j}^{16}\right\| \qquad (4)$$

The same process is continued for other combinations in Eqs. 5, 6 and 7. In addition, the different encoded images $\sum_{i=0,j=0}^{m,n} D_{i,j}^1$, $\sum_{i=0,j=0}^{m,n} D_{i,j}^2$, $\sum_{i=0,j=0}^{m,n} D_{i,j}^3$ .... $\sum_{i=0,j=0}^{m,n} D_{i,j}^{16}$ are merged together as an image $\sum_{i=0,j=0}^{m,n} D_{i,j}$ in Eqs. 8, 9 and 10.

$$\sum_{i=0,j=0}^{m,n} D_{i,j}^2 = \uplus lsp\left\|\sum_{i=0,j=0}^{m,n} C_{i,j}^2 \oplus \sum_{i=0,j=0}^{m,n} A_{i,j}^3\right\| \qquad (5)$$

$$\sum_{i=0,j=0}^{m,n} D_{i,j}^3 = \uplus lsp\left\|\sum_{i=0,j=0}^{m,n} C_{i,j}^3 \oplus \sum_{i=0,j=0}^{m,n} A_{i,j}^5\right\| \qquad (6)$$

$$\sum_{i=0,j=0}^{m,n} D_{i,j}^{16} = \uplus lsp\left\|\sum_{i=0,j=0}^{m,n} C_{i,j}^{16} \oplus \sum_{i=0,j=0}^{m,n} A_{i,j}^1\right\| \qquad (7)$$

$$\sum_{i=0,j=0}^{m,n} D_{i,j} = \sum_{i=0,j=0}^{m,n} D_{i,j}^1 \oplus \sum_{i=0,j=0}^{m,n} D_{i,j}^2 \oplus \sum_{i=0,j=0}^{m,n} D_{i,j}^3 \ldots \sum_{i=0,j=0}^{m,n} D_{i,j}^{16} \qquad (8)$$

$$\sum_{i=0,j=0}^{m,n} D_{i,j} = \uplus lsp\left\|\sum_{i=0,j=0}^{m,n} C_{i,j}^1 \oplus \sum_{i=0,j=0}^{m,n} A_{i,j}^{16}\right\| \bowtie \uplus lsp\left\|\sum_{i=0,j=0}^{m,n} C_{i,j}^2 \oplus \sum_{i=0,j=0}^{m,n} A_{i,j}^3\right\|$$

$$\ldots \bowtie \uplus lsp \| \sum_{i=0,j=0}^{m,n} C_{i,j}^{16} \oplus \sum_{i=0,j=0}^{m,n} A_{i,j}^{1} \| \qquad (9)$$

$$\sum_{i=0,j=0}^{m,n} D_{i,j} = \uplus lsp \| \sum_{i=0,j=0}^{m,n} C_{i,j} \oplus \sum_{i=0,j=0}^{m,n} A_{i,j} \| \qquad (10)$$

2.2. The alpha- encryption (AE) for encoded image

In the Alpha Encryption (AE) process, the encoded image is divided into 4x4 equal parts $\sum_{i=0,j=0}^{m,n} A_{i,j}^{1}$ … $\sum_{i=0,j=0}^{m,n} A_{i,j}^{8}$ and $\sum_{i=0,j=0}^{m,n} B_{i,j}^{1}$ … $\sum_{i=0,j=0}^{m,n} B_{i,j}^{8}$ in Eqs. 11-14. The segregated parts are shuffled within the image itself based on odd and even numbers. The shuffled parts of the pixels are converted into corresponding alphabetic characters using a lookup table in Table 1 and Fig. 5. The converted characters and header information are written in a file. In result, the file e contains cipher text and header information about the cipher text. The detailed AE process steps are given below (Almutairi and Manimurugan, 2016):

- Strat loop
- Read input pixel Xi
- If the pixel value is < 4 digits
- Open a new file F
- Read the first 2 digits and find the corresponding character from table
- Read last digit value and find the corresponding character from the table.
- Both characters are written in the file
- Close file F
- End if
- Do until last pixel X_n
- End loop
- Create a header information
- Add header information in same file F
- Send the F to receiver

Table 1: Lookup table

0	1	2	3	4	5	6	7	8	9
A	B	C	D	E	F	G	H	I	J
10	11	12	13	14	15	16	17	18	19
K	L	M	N	O	P	Q	R	S	T
20	21	22	23	24	25	00	01	02	03
U	V	W	X	Y	Z	A	B	C	D
04	05	06	07	08	09				
E	F	G	H	I	J				

$$\sum_{i=0,j=0}^{m,n} D_{i,j} = \sum_{i=0,j=0}^{m,n} A_{i,j} \oplus \sum_{i=0,j=0}^{m,n} B_{i,j} \qquad (11)$$

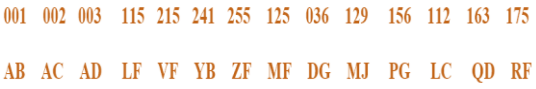
$$\sum_{i=0,j=0}^{m,n} A_{i,j} = \sum_{i=0,j=0}^{m,n} A_{i,j}^{1} \oplus \sum_{i=0,j=0}^{m,n} A_{i,j}^{2} \ldots \sum_{i=0,j=0}^{m,n} A_{i,j}^{8} \qquad (12)$$

$$\sum_{i=0,j=0}^{m,n} B_{i,j} = \sum_{i=0,j=0}^{m,n} B_{i,j}^{1} \oplus \sum_{i=0,j=0}^{m,n} B_{i,j}^{2} \ldots \sum_{i=0,j=0}^{m,n} B_{i,j}^{8} \qquad (13)$$

$$\sum_{i=0,j=0}^{m,n} D_{i,j} = \sum_{i=0,j=0}^{m,n} A_{i,j}^{1} \ldots \sum_{i=0,j=0}^{m,n} A_{i,j}^{8} \oplus \sum_{i=0,j=0}^{m,n} B_{i,j}^{1} \ldots \sum_{i=0,j=0}^{m,n} B_{i,j}^{8} \qquad (14)$$

As an example, the encoded image pixels values are given in Fig. 5. As per the AE algorithm, the first pixel X_1 value is 001 and the values of the first two digits of the same is 00. So, the corresponding character of 00 is 'A' in Table 1. The last digit is 1 and its corresponding character is 'B'. Therefore, the plain text value of '001' is converted into the cipher text of 'AB'.

The same process is continued up to the last pixel. The different segregated parts of the AE process are given in Eqs. 15-18. The ∐ denotes a reference symbol and T is a lookup table. Once the AE process is over, the different parts of the $A_{i,j}^{1}$ ….. $A_{i,j}^{8}$ and $B_{i,j}^{1}$ …. $B_{i,j}^{8}$ cipher text's information are written in a .txt file e as per the shuffle order in Eq. 19. The header information h is created and written in e.



001 002 003 115 215 241 255 125 036 129 156 112 163 175

AB AC AD LF VF YB ZF MF DG MJ PG LC QD RF

Fig. 5: An example of pixel to character conversion

$$\sum_{i=0,j=0}^{m,n} A_{i,j}^{1} \bowtie \coprod T = e_1 \qquad (15)$$

$$\sum_{i=0,j=0}^{m,n} A_{i,j}^{8} \bowtie \coprod T = e_8 \qquad (16)$$

$$\sum_{i=0,j=0}^{m,n} B_{i,j}^{1} \bowtie \coprod T = e_9 \qquad (17)$$

$$\sum_{i=0,j=0}^{m,n} B_{i,j}^{8} \bowtie \coprod T = e_{16} \qquad (18)$$

$$(e_1 \oplus e_3 \oplus e_5 \oplus e_7 \oplus e_9 \oplus e_{11} \oplus e_{13} \oplus e_{15} \oplus e_2 \oplus e_4 \oplus e_6 \oplus e_8) \oplus (\oplus e_{10} \oplus e_{12} \oplus e_{14} \oplus e_{16}) + h = e \qquad (19)$$

2.3. Hill cipher encryption process for e

The Hill Cipher (HC) is a symmetric encryption technique, where the secret letters are encrypted into cipher letters. It is also called a polygraphic system. In this process, AE cipher text e is considered as a secret text S. A secret text S is encrypted as a cipher text C using an encryption key κ_e in Eq. 20. After the HC encryption process, the cipher text C is sent along with the encryption key κ_e to the other end/authenticated person for the decryption process.

$$C = [\kappa_e \times S] \; mod \; 26 \qquad (20)$$

As an example, the secret letters 'HATS' and encryption key κ_e 'DCDF' are considered. The secret letters are divided into different pairs of HA and TS. The paired characters are converted into numeric values using Table 1 (Anton, 2010). In result, HA is substituted as 7 0 and TS is converted into 19 18. The same process is done for the κ_e. In connection with same, the κ_e is converted into 3 2 3 5. The overall HC encryption process steps are given as follows:

- $S = S_1 \oplus S_2$
- $S_1 = \begin{bmatrix} 7 \\ 0 \end{bmatrix}$, $S_2 = \begin{bmatrix} 19 \\ 18 \end{bmatrix}$, $\kappa_e = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix}$
- As per the Eq. 20, the two secret pairs are encrypted in a separate manner.
- $C_1 = [\kappa_e \times S_1] \; mod \; 26$
- $C_1 = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} 7 \\ 0 \end{bmatrix} mod \; 26, C_1 = \begin{bmatrix} 21 \\ 14 \end{bmatrix} \; mod \; 26, C_1 = \begin{bmatrix} 21 \\ 14 \end{bmatrix}$
- $C_2 = [\kappa_e \times S_2] \; mod \; 26$
- $C_2 = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} 19 \\ 18 \end{bmatrix} mod \; 26, C_2 = \begin{bmatrix} 111 \\ 128 \end{bmatrix} \; mod \; 26,$
- $C_2 = \begin{bmatrix} 7 \\ 24 \end{bmatrix}$
- $C = C_1 \oplus C_2$
- $C = 21 \quad 14 \quad 7 \quad 24 = VOHY$

Therefore, the original text ‘HATS’ is converted as a cipher text of ‘VOHY’.

3. Proposed triple decryption to reconstruct a secret image

Once the cipher information C and encryption key κ_e are received from the sender, this will be considered as an input for the Triple Decryption (TD) process in the receiver’s side. It can be classified into three different processes of: HC decryption, Alpha-Decryption (AD) and Inverse Matrix (IM) decode. In the first stage, the C is decrypted by HC decryption and AD processes using a lookup table (Table 1). The decrypted data $\sum_{i=0,j=0}^{m,n} D_{i,j}$ is decoded in an IM decode process. In result, the original secret and cover images are obtained. The above mentioned different processes are stated in the following sections (Almutairi and Manimurugan, 2016).

3.1. Hill Cipher decryption process for cipher text C

The received cipher text of C and encryption key κ_e are used for the decryption process. In this process, κ_e^{-1} and the determinant of κ_e are computed from the encryption key κ_e in Eqs. 21-23. The D denotes the determinant of κ_e. To find the decryption key κ_d, the computational value B is calculated from Eqs. 24 and 25. Using κ_d and C, the AE cipher text is retrieved from Eq. 26 (Fig. 6).

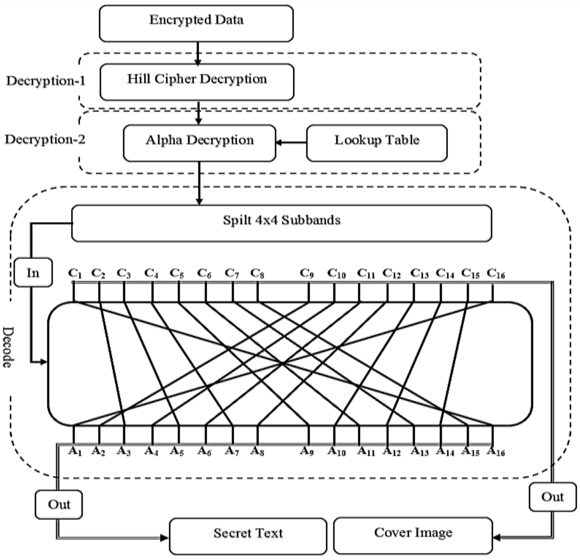


Fig. 6: The triple decryption process

$$\kappa_e = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{21}$$

$$\kappa_e^{-1} = \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \tag{22}$$

$$D = |k_e| = (ad - bc) \tag{23}$$

$$D \times B = 1\, mod\ 26 \tag{24}$$

$$k_d = B[k_e^{-1}]\, mod\ 26 \tag{25}$$

$$S = [k_d \times C]\, mod\ 26 \tag{26}$$

As an example, consider the HC encryption process (section II.C). The cipher text is ‘VOHY’ and the encryption key is $k_e = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix}$. As per Eqs. 21 and 22 the following steps are done.

- $k_e = \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix}, \kappa_e^{-1} = \begin{bmatrix} 5 & -3 \\ -2 & 3 \end{bmatrix}$
- Compute the D from Eq. 24
- $D = |k_e| = (3 \times 5 - 2 \times 3) = 9$
- Find the value of B from Eq. 24
- $9 \times B = 1\, mod\ 26\ , 9 \times 3 = 1\, mod\ 26, B = 3$
- Find the decryption key of k_d from Eq. 25
- $k_d = 3\begin{bmatrix} 5 & -3 \\ -2 & 3 \end{bmatrix} mod\ 26\ , k_d = 3\begin{bmatrix} 5 & 23 \\ 24 & 3 \end{bmatrix} mod\ 26$
- $k_d = \begin{bmatrix} 15 & 69 \\ 72 & 9 \end{bmatrix} mod\ 26, k_d = \begin{bmatrix} 15 & 17 \\ 20 & 9 \end{bmatrix}$
- Decrypt the cipher text $C =$ ‘VOHY’ using Eq. 26
- $C = VOHY =\ 21 \quad 14 \quad 7 \quad 24$
- $C = C_1 \oplus C_2$
- $S_1 = [k_d \times C_1]\, mod\ 26\ , S_1 = (\begin{bmatrix} 15 & 17 \\ 20 & 9 \end{bmatrix}\begin{bmatrix} 21 \\ 14 \end{bmatrix})\, mod\ 26$
- $S_1 = \begin{bmatrix} 553 \\ 546 \end{bmatrix} mod\ 26 = \begin{bmatrix} 7 \\ 0 \end{bmatrix}$
- $S_2 = [k_d \times C_1]\, mod\ 26\ , S_2 = (\begin{bmatrix} 15 & 17 \\ 20 & 9 \end{bmatrix}\begin{bmatrix} 7 \\ 24 \end{bmatrix})\, mod\ 26$
- $S_2 = \begin{bmatrix} 513 \\ 356 \end{bmatrix} mod\ 26 = \begin{bmatrix} 19 \\ 18 \end{bmatrix},$
- $S = S_1 \oplus S_2 = 7 \quad 0 \quad 19 \quad 18 = HATS$
- Finally, the decrypted cipher text is $‘HATS'$. This is also called secret text S

3.2. Alpha-decryption (AD) process for S

The received decrypted data S is segregated into a header and cipher characters. The S is also called cipher text e. After this segregation, the cipher characters are divided into 16 parts of $e_1, e_2, e_3 \ldots e_{16}$ in Eq. 27. The $e_1, e_2, e_3 \ldots e_{16}$ parts are decrypted in a separate manner using a lookup table in Eqs. 28-31. During the decryption process, the first two cipher characters are considered as an input and its corresponding values are found from a lookup table (Table 1). The same process is continued up to the end cipher character in e_1, e_2, e_3 ... e_{16}. Once the process is over, all decrypted data are combined together in reshuffled process (Eqs. 32-35). In result, the decoded image $\sum_{i=0,j=0}^{m,n} D_{i,j}$ can be obtained.

$$e = h + (e_1 \oplus e_3 \oplus e_5 \oplus e_7 \oplus e_9 \oplus e_{11} \oplus e_{13} \oplus e_{15})$$
$$\oplus(\oplus e_2 \oplus e_4 \oplus e_6 \oplus e_8 \oplus e_{10} \oplus e_{12} \oplus e_{14} \oplus e_{16}) \tag{27}$$

$$e_1 \bowtie \coprod T = \sum_{i=0,j=0}^{m,n} A_{i,j}^1 \tag{28}$$

$$e_8 \bowtie \coprod T = \sum_{i=0,j=0}^{m,n} A_{i,j}^8 \tag{29}$$

$$e_9 \bowtie \coprod T = \sum_{i=0,j=0}^{m,n} B_{i,j}^1 \tag{30}$$

$$e_{16} \bowtie \coprod T = \sum_{i=0,j=0}^{m,n} B_{i,j}^8 \tag{31}$$

The segregated parts of e_1, e_2, e_3 ... e_{16} generated values are combined and validated with header information in Eqs. 32-35. If the validation is true, the decrypted values of $\sum_{i=0,j=0}^{m,n} D_{i,j}$ is considered for the IM decode process. If not, the receiver has to send a request to the sender to resend the data once again. The overall decryption process steps are given below (Almutairi and Manimurugan, 2016):

- Read input file
- loop
- Read the first 2 characters
- find the corresponding value for the character from the table
- create image array
- Write the value in image array
- do until last two characters
- End loop
- If header information = generated array information
- The generated image is correct
- Else
- Generated image is incorrect
- Request sender to resend the file once again
- Do step 1
- End if

$$\sum_{i=0,j=0}^{m,n} A_{i,j}^{1} \oplus \sum_{i=0,j=0}^{m,n} A_{i,j}^{2} \dots \sum_{i=0,j=0}^{m,n} A_{i,j}^{8} = \sum_{i=0,j=0}^{m,n} A_{i,j} \tag{32}$$

$$\sum_{i=0,j=0}^{m,n} B_{i,j}^{1} \oplus \sum_{i=0,j=0}^{m,n} B_{i,j}^{2} \dots \sum_{i=0,j=0}^{m,n} B_{i,j}^{8} = \sum_{i=0,j=0}^{m,n} B_{i,j} \tag{33}$$

$$\sum_{i=0,j=0}^{m,n} A_{i,j}^{1} \dots \sum_{i=0,j=0}^{m,n} A_{i,j}^{8} \oplus \sum_{i=0,j=0}^{m,n} B_{i,j}^{1} \dots = \sum_{i=0,j=0}^{m,n} B_{i,j}^{8} \sum_{i=0,j=0}^{m,n} D_{i,j} \tag{34}$$

$$\sum_{i=0,j=0}^{m,n} A_{i,j} \oplus \sum_{i=0,j=0}^{m,n} B_{i,j} = \sum_{i=0,j=0}^{m,n} D_{i,j} \tag{35}$$

The reverse process of Fig. 5 is the AD process. The paired characters are considered to reconstruct the value. For example, consider the first two cipher characters in Fig. 5. Instead of the first character, the value of two digits will be substituted using Table 1. Similarly, instead of the last character a single digit value will be substituted. Therefore, every pair of characters will generate three digit values. These values vary from 000 to 255.

3.3. Inverse-Matrix (IM) decode process for retrieve the original secret information

The decrypted image $\sum_{i=0,j=0}^{m,n} D_{i,j}$ is considered as an input for the IM decode process. The $\sum_{i=0,j=0}^{m,n} D_{i,j}$ is divided into 4X4 equal subbands of $\sum_{i=0,j=0}^{m,n} D_{i,j}^{1}$, $\sum_{i=0,j=0}^{m,n} D_{i,j}^{2} \dots \sum_{i=0,j=0}^{m,n} D_{i,j}^{16}$ in Eq. 36.

$$\sum_{i=0,j=0}^{m,n} D_{i,j} = \sum_{i=0,j=0}^{m,n} D_{i,j}^{1} \oplus \sum_{i=0,j=0}^{m,n} D_{i,j}^{2} \oplus \dots \oplus \sum_{i=0,j=0}^{m,n} D_{i,j}^{16} \tag{36}$$

All divided subband pixels are converted into 8-bit binary value and the least significant bits $\Subset lsp$ are separated from every 8-bit values in Eqs. 37-41. This process is continued up to end of the pixel in every subband. In result, the cover

$(\sum_{i=0,j=0}^{m,n} C_{i,j}^{1} \dots\dots \sum_{i=0,j=0}^{m,n} C_{i,j}^{16})$

and secret

$(\sum_{i=0,j=0}^{m,n} A_{i,j}^{1} \dots\dots \sum_{i=0,j=0}^{m,n} A_{i,j}^{16})$
images are obtained from this decode process.

$$\Subset lsp \left\| \sum_{i=0,j=0}^{m,n} D_{i,j}^{1} \right\| = \sum_{i=0,j=0}^{m,n} C_{i,j}^{1} \oplus \sum_{i=0,j=0}^{m,n} A_{i,j}^{16} \tag{37}$$

$$\Subset lsp \left\| \sum_{i=0,j=0}^{m,n} D_{i,j}^{2} \right\| = \sum_{i=0,j=0}^{m,n} C_{i,j}^{2} \oplus \sum_{i=0,j=0}^{m,n} A_{i,j}^{3} \tag{38}$$

$$\Subset lsp \left\| \sum_{i=0,j=0}^{m,n} D_{i,j}^{3} \right\| = \sum_{i=0,j=0}^{m,n} C_{i,j}^{3} \oplus \sum_{i=0,j=0}^{m,n} A_{i,j}^{5} \tag{39}$$

$$\Subset lsp \left\| \sum_{i=0,j=0}^{m,n} D_{i,j}^{16} \right\| = \sum_{i=0,j=0}^{m,n} C_{i,j}^{16} \oplus \sum_{i=0,j=0}^{m,n} A_{i,j}^{1} \tag{40}$$

The $\sum_{i=0,j=0}^{m,n} C_{i,j}^{1} \dots \sum_{i=0,j=0}^{m,n} C_{i,j}^{16}$ are combined together as an image $\sum_{i=0,j=0}^{m,n} C_{i,j}$ (cover image). Similarly, $\sum_{i=0,j=0}^{m,n} A_{i,j}^{1} \dots \sum_{i=0,j=0}^{m,n} A_{i,j}^{16}$ are combined together as an image $\sum_{i=0,j=0}^{m,n} A_{i,j}$ (secret image) in Eqs. 42 and 43. This $\sum_{i=0,j=0}^{m,n} A_{i,j}$ secret text image is validated by standard parameters in experimentation section IV.

$$\sum_{i=0,j=0}^{m,n} D_{i,j} = \Subset lsp \left\| \sum_{i=0,j=0}^{m,n} D_{i,j}^{1} \right\| \bowtie \Subset lsp \left\| \sum_{i=0,j=0}^{m,n} D_{i,j}^{2} \right\| \dots \bowtie \Subset lsp \left\| \sum_{i=0,j=0}^{m,n} D_{i,j}^{16} \right\| \tag{41}$$

$$\sum_{i=0,j=0}^{m,n} C_{i,j} = \sum_{i=0,j=0}^{m,n} C_{i,j}^{1} + \sum_{i=0,j=0}^{m,n} C_{i,j}^{2} \dots + \sum_{i=0,j=0}^{m,n} C_{i,j}^{16} \tag{42}$$

$$\sum_{i=0,j=0}^{m,n} A_{i,j} = \sum_{i=0,j=0}^{m,n} A_{i,j}^{1} + \sum_{i=0,j=0}^{m,n} A_{i,j}^{2} \dots + \sum_{i=0,j=0}^{m,n} A_{i,j}^{16} \tag{43}$$

4. Experimental results and discussions

The experimental results of the proposed method are presented and discussed in this section. The program was written in MATLAB and run on a personal computer. There are 1,000 images taken for the demonstration. However, in this documentation, we have declared four, standard grey scale covers images and a secret image in Fig. 7. All images are in .bmp format. We have considered a secret text image size of 256X256 of greyscale, and the cover image size is 512X512 greyscale in Fig. 7. Fig. 9 shows an overall flow diagram of the proposed TE and TD processes.

The reason for choosing the greyscale image is that, during the encode time, if the last two bit values are changed, there is little colour difference in the greyscale image. This is one of the main reasons/advantages in preventing the data from human visual attack. In the RM encode process, the cover image size is 512x512 and the total number of pixels is 262,144. On the other hand, a secret image size is 256x256 and the total number of the pixels is 65,536. The cover and secret images are converted into 8-bit binary values before the encode process. In connection with that, the cover image has 2,097,152 bits and 524,288 bits are in the secret image.

During the encode time, the cover image in the last two bit values of every pixel are replaced by the secret image bits shown in Fig. 10. This replacement process is made in reverse order in Fig. 8. The main reason for reverse replacement is to improve the algorithm complexity. Therefore, it is very difficult for an intruder or third parties to hack the original information.

Due to the above mentioned reasons, this process is called reverse a matrix encoding process (Manimurugan et al., 2014a).

The encoded image subbands are shuffled (4x4 basis) within the image itself in Fig. 9. This shuffled image is encrypted by the AE process. In Alpha-Encryption the shuffled image pixels are replaced by the characters, and this is purely based on the substitution process. After this AE process, the encrypted data are once again encrypted by HC encryption. In result, encrypted data and an encryption key can be generated.

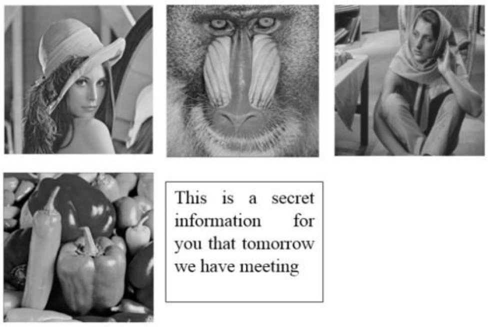

This is a secret information for you that tomorrow we have meeting

Fig. 7: Sample covers images and secret image

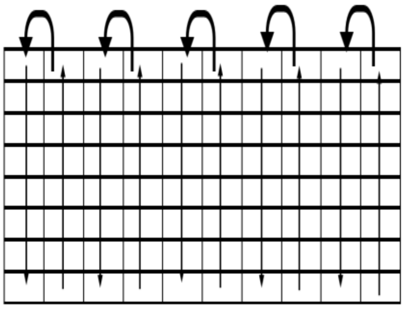


Fig. 8: The RM encode pixels order

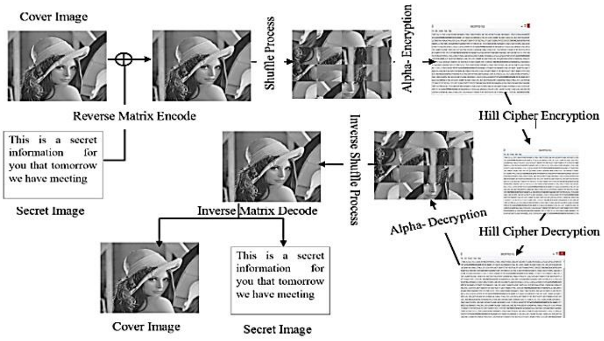


Fig. 9: Flow diagram of Triple encryption and decryption process

During the decryption process, the encrypted data of HC and the encryption key are considered for input. The cipher data of AE is retrieved from this HC decryption process. In addition, the same retrieved data is converted into a shuffled image with the support of a lookup table in the AD process. The shuffled image is reshuffled to retrieve the original encoded image in an inverse shuffled process. The encoded image is decoded by an IM decode process to retrieve the original secret and cover images in Fig. 9 (Almutairi and Manimurugan, 2016).

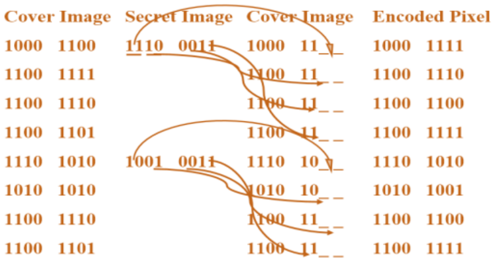


Fig.10: Secret image bit substitutions process

To strengthen the algorithm, the encoded image subbands are shuffled after the encode process. These shuffled image pixel values are converted as characters in the AE process. To achieve the integrity, a separate header file is generated and it is enclosed with the cipher text file during the AE process.

These converted characters are once again encrypted by HC encryption. Due to these processes, three encryption processes are performed. The main advantage of TE is that, in every stage the pixel positions are interchanged. Therefore, it is very difficult for intruders/third parties to retrieve the original data/hack the original data (Manimurugan et al., 2014b). Table 2 shows that triple encryption is superior to other conventional methods.

$$MSE = \frac{1}{m\,n} \sum_{i=o}^{m-1} \sum_{j=0}^{n-1} [O(i,j) - R(i,j)]^2 \qquad (37)$$

$$PSNR = 10\, log_{10} (\frac{MAX^2}{MSE}) \qquad (38)$$

In addition, to minimize the execution time, the algorithm is designed based on the substitution process. The experimentation was done in a systematic manner and the proposed algorithm provided excellent results. The Human Visual Attack (HVA) was also made after the encode process. Similarly, a Pixel Attack (PA) was done against the algorithm. In result, the proposed algorithm performed well against the both attacks.

The quality of the secret image is identified by the CC (correlation coefficient). The encoded image signal ratio is measured by PSNR (peak signal to noise ratio). PSNR (Peak signal to noise ratio) is one of the most common parameters used to measure the quality of reconstructed images in all areas. Although, a higher PSNR generally indicates that the reconstructed image is of a higher quality, in some cases it may not be. PSNR is most easily defined via the mean squared error (MSE). Based on monochrome m×n cover image O and encoded image R, we can define the MSE in Eq. 37 and PSNR (in dB) in Eq. 38.

In Table 2, the signal ratios of both the conventional and proposed RM encode methods are given. The existing steganography encode method is considered as a conventional method. This method was implemented for comparative purposes in MATLAB. In Table 2, the proposed RM encodes provided better results compared to the conventional method. This is illustrated in Figs. 11 and 12. The provided signal ratios are between 48 to 50 dB.

Table 2: Encoded Image Signal Ratio

Images	Conventional Encode (dB)	Proposed Encode (dB)
Lena	44.31	49.45
Baboon	42.74	48.99
Barbara	42.89	48.23
peppers	43.72	49.53

The proposed method (TE) and conventional methods are compared by different parameters of time, PSNR, correlation coefficient (CC), confidentiality, integrity, authentication, human visual attack, pixel attack and algorithm complexity, and are shown in Table 3. In this comparison, the cover image of lena.bmp and secret image secret.bmp are considered.

Table 3: Comparison of proposed and conventional method

Parameters	Conventional Encode	Proposed Encode (RM)	Triple Encryption (Proposed Encode + Encryptions)
Time (Sec)	68.00	56.20	75.28
PSNR (dB)	44.31	46.45	
Secret Image quality (CC)	0.862	0.998	0.992
Confidentiality	Medium	High	Too high
Intergrity	Medium	High	Too high
Authentication	Medium	High	Too high
HVA	High	Too High	Too High
PA	High	Too High	Too High
Complexity	Medium	High	Too High

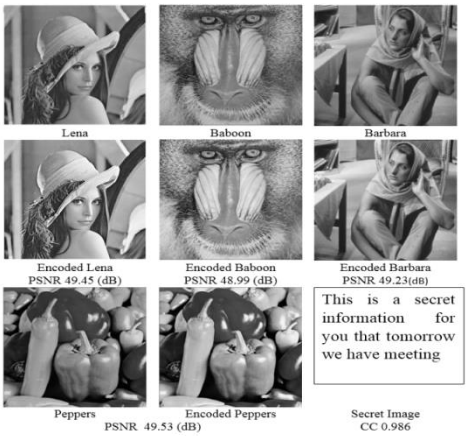


Fig. 11: The signal ratio of encoded and reconstructed secret images

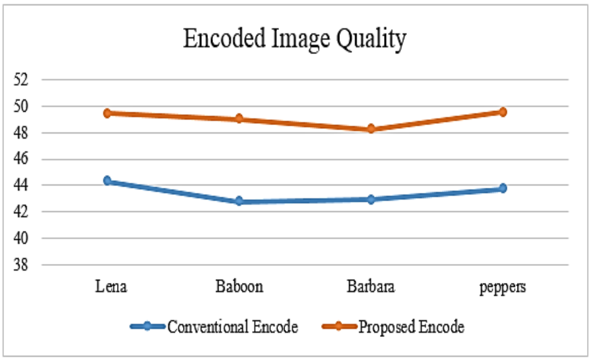


Fig. 12: PSNR comparison of conventional method with RM Encode

The conventional method has taken 68 seconds for the encode process, but the proposed RM encode has taken 56.2 seconds. In addition, the TE process has taken 75.28 seconds for the entire process of RM encode, AE and HC encryption processes. In terms of signal to noise ratio, the proposed RM encode method provided better PSNR results (46.45 dB) than the conventional method in Fig. 12. The PSNR is calculated between the encoded and original cover images, so it is not applicable in the TE process. In the correlation coefficient, the result values are from 0 to 1. When CC is near to 1, the reconstructed image is an exact replica of the original image.

Based on this point, the proposed RM encode has given the higher value of 0.998 compared to the conventional encode process. The TE also provided a better result of CC 0.992 than the conventional method.

Regarding confidentiality, both the conventional method and the proposed RM encode provided better performance. However, the TE provided higher confidentiality due to the triple processes of encode and encryptions in Fig. 13. In terms of authentication, both encode methods gave a good performance. But, the TE provided a better result than the other two encode processes due to the pixel shuffle and substitution. To ensure the integrity, the header files are sent along with the cipher text in the TE process. Therefore, compared to the other two encode methods, the proposed TE provided higher integrity. In experimentation, after the encode process the Human Visual Attack (HVA) is done. In this case, the proposed TE is superior to the other two encodes. This is because, after the encode process, the image is represented as characters in TE. In Pixel Attack (PA), due to the pixel exchange, the proposed TE algorithm is too strong compared to the other two. It also provided higher complexity, because in every stage the pixel positions are replaced and has converted as cipher characters. Due to these reasons, the proposed TE is more complex than the other two.

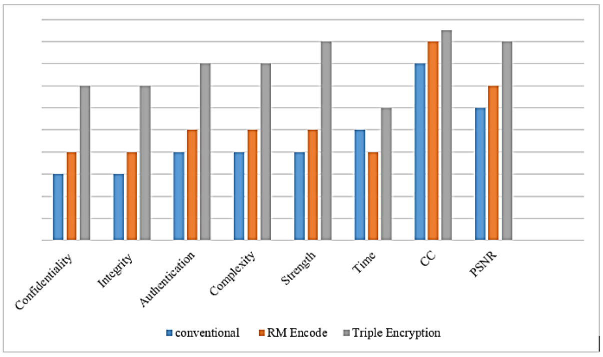


Fig. 13: Overall comparisons with conventional, RM Encode and Triple Encryption methods.

5. Conclusion

Many algorithms have been proposed to encrypt an image. However, in this paper, we have proposed an image hide technique with the suitable encryption of triple encryption. The main advantage of this method is based on the substitution and pixel shuffle process. To strengthen the proposed algorithm, the pixel substitution is made in reverse order in the RM encode process. To improve the algorithm complexity, at every stage the pixel positions are interchanged and are converted as cipher characters. For these reasons, the proposed TE takes a minimum execution time for the encryption and decryption processes. In addition, the PSNR and correlation coefficient (CC) results are also superior to the conventional method. We carried out different attack processes of PA and HVA in our research laboratory. The proposed TE algorithm performed well against both attacks. In the AE process, substitution of the characters and the procedure are entirely different from other substitution methods. To make this stronger, the encrypted data are once again encrypted by HC encryption. This is one of main reasons why third parties find it very difficult to hack the original secret information. However, conventional methods failed at this point. To ensure the integrity of the reconstructed data, the sender sends the encrypted data along with header information to the receiver. Therefore, the proposed TE achieved high CIA (confidentiality, Integrity, Authentication), less execution time of encryption and decryption, good PSNR, high complexity, good strength and 99% of the original secret information can be retrieved from our method.

This TE is mainly designed for defence purposes. In future, the same algorithm will be used for telemedicine with some modifications.

Acknowledgement

The authors would like to thank the Saudi Electronics University and Tabuk University, Saudi Arabia for giving immense support to carry out this research work. The suggestions and comments of anonymous reviewers for this accomplishment, which have greatly helped to improve the quality of this paper, are acknowledged. Finally, special thanks to the editor for their support and valuable effort.

References

Almutairi S and Manimurugan S (2016). An efficient secret image transmission scheme using Dho-encryption technique. International Journal of Computer Science and Information Security (IJCSIS), 14(10): 446-460.

Anton H (2010). Elementary linear algebra application version. 10th Edition, John Wiley and Sons, New Jersey, USA.

Cao H and Kot AC (2013). On establishing edge adaptive grid for bilevel image data hiding. IEEE Transactions on Information Forensics and Security, 8(9): 1508-1518.

Haiping H, Shichao H, Jiutian C, and Ruchuan W (2013). Image hiding algorithm in discrete cosine transform domain based on grey prediction and grey relational analysis. China Communications, 10(7): 57-70.

Hu YC (2003). Grey-level image hiding scheme based on vector quantisation. Electronics Letters, 39(2): 202-203.

Ishimaru A, Zhang C, and Kuga Y (2014). Hard wall imaging of objects hidden by non-penetrating obstacles using modified time reversal technique. IEEE Transactions on Antennas and Propagation, 62(7): 3645-3651.

Kwon O (2014). Single image dehazing based on hidden Markov random field and expectation–maximisation. Electronics Letters, 50(20): 1442-1444.

Lee JD, Chiou YH, and Guo JM (2014). Information hiding based on block match coding for vector quantization-compressed images. IEEE Systems Journal, 8(3): 737-748.

Li M, Xiao D, Kulsoom A, and Zhang Y (2015). Improved reversible data hiding for encrypted images using full embedding strategy. Electronics Letters, 51(9): 690-691.

Ma K, Zhang W, Zhao X, Yu N, and Li F (2013). Reversible data hiding in encrypted images by reserving room before encryption. IEEE Transactions on Information Forensics and Security, 8(3): 553-562.

Manimurugan S, Narmatha C, and Porkumaran K (2014a). The new approach of visual cryptography scheme for protecting the grayscale medical images. Journal of Theoretical and Applied Information Technology, 69(3): 552-561.

Manimurugan S, Porkumaran K, and Narmatha C (2014b). The new block pixel sort algorithm for TVC-encrypted medical image. The Imaging Science Journal, 62(8): 403-414.

Nikolaidis A (2015). Reversible data hiding in JPEG images utilising zero quantised coefficients. IET Image Processing, 9(7): 560-568.

Qian Z and Zhang X (2016). Reversible data hiding in encrypted images with distributed source encoding. IEEE Transactions on Circuits and Systems for Video Technology, 26(4): 636-646.

Qin C, Chang CC, and Chiu YP (2014). A novel joint data-hiding and compression scheme based on SMVQ and image inpainting. IEEE Transactions on Image Processing, 23(3): 969-978.

Renza D, Ballesteros DM, and Ortiz HD (2016). Text Hiding in Images Based on QIM and OVSF. IEEE Latin America Transactions, 14(3): 1206-1212.

Wu DC and Tsai WH (2000). Spatial-domain image hiding using image differencing. IEE Proceedings-Vision, Image and Signal Processing, 147(1): 29-37.

Wu HT, Dugelay JL, and Shi YQ (2015). Reversible image data hiding with contrast enhancement. IEEE Signal Processing Letters, 22(1): 81-85.

Xiao D and Chen S (2014). Separable data hiding in encrypted image based on compressive sensing. Electronics Letters, 50(8): 598-600.

Zhang X and Zhang W (2014). Semantic image compression based on data hiding. IET Image Processing, 9(1): 54-61.

Zhang X, Long J, Wang Z, and Cheng H (2016). Lossless and reversible data hiding in encrypted images with public-key cryptography. IEEE Transactions on Circuits and Systems for Video Technology, 26(9): 1622-1631.

Zhou J, Sun W, Dong L, Liu X, Au OC and Tang YY (2016). Secure reversible image data hiding over encrypted domain via key modulation. IEEE Transactions on Circuits and Systems for Video Technology, 26(3): 441-452.