Increasing Embedding Efficiency & Security of Extended Matrix Encoding Algorithm by Providing Compression & Encryption

Prof. Manoj A.Chaudhari[1]
Assistant Professor, Dept. of IT,
Amrutvahini College of Engineering,
Sangamner, Maharashtra, India
e-mail: manojchaudhari1988@yahoo.com

Mr. Pritesh K. Patil[2]
Student, Dept. of IT,
Amrutvahini College of Engineering,
Sangamner, Maharashtra, India
e-mail: mr.pkpatil1989@gmail.com

Abstract:- Extended Matrix Encoding Algorithm is totally different from most of the LSB replacement or matching steganographic schemes. With reducing the amount of necessary changes the extended matrix algorithm is used to increase embedding efficiency. By using this algorithm, the hidden message is inserted into carrier media and can be transferred via safer channel. In this algorithm the quantitative DCT coefficients of JPEG image which makes the data safe from visual attack. The embedding efficiency and embedding rate get increased to large extent by changing the hash function in matrix encryption and changing the coding mode. In this paper I am trying to show that we can increase embedding efficiency by compressing the secrete data also increase the security by encryption and provision of double password.

Keywords:- *Compression, DCT Coefficient, Encryption, Embedding efficiency and rate.*

*****

I. Introduction

Steganography is the art and science of writing the secret content inside cover media and transferring the stego media from the sender to intended recipient through a subliminal channel without arousing the suspicion of adversary. The presence of hidden info is meant to be undetectable. If the actual fact that communication is happening is revealed, the steganography is cracked not with standing whether or not or not the hidden info is exposed. Thus, compared with other connected techniques like watermarking, the property of covertness plays a crucial role within the stegosystem.

In order to create stegotext apparently innocent, the confidential message is typically embedded into the redundant components of cover media. For digital image, the least significant bit plane in spacial domain is one reasonably these components that appear as if completely random and noisy. The modification of LSB won't cause noticeable change of the looks of image. Several LSB based techniques of data hiding are proposed in recent years [2,3]. Derek Upham’sJSteg was most likely the primary in public accessible steganographic system for JPEG images [4]. This technique is actually a copy of the LSB substitution algorithm in spacial domain. The least-significant bit of DCT coefficients is consecutive replaced with the secret message. Since the replacement solely happens on 2 adjacent coefficients, it'll cause a statistically obvious POVs (pairs of values) problem which may be with success detected by X^2 -test proposed by Westfeld and P fitzmann [5].

II. EXISTING SYSTEM

A. *Extended Encoding Algorithm*

How to do the independent increase of k? In matrix encoding, the hash function maps n bits carrier data into a certain length of binary sequence that depends upon the bit length of index i. And the bit length of i is chosen as the same size as that of secret message w. Given the extension of the length of i is realizable, we will finally embed additional bits of secret message into one cell. Taking the cell (1,3,2) as associate example, 2-bit secret message will be embedded into 3-bit embedding cell by changing only 1-bit position of the cell. The length of i is two. If the length of i is extended to be 3-bit, we can take one additional bit of secret message to implement exclusive-or with the binary result of hash function. However, the problem shows up. The result of XOR operation, namely, the index y which will be used to indicate the position to be modified is out of the range. We may get $y = (101)_2 = 5$ in that case, However it’s not possible to find out the fifth bit position for a cell of 3-bit length. Essentially modifying just 1 bit or keeping unchanged in the carrier cell with 3-bit length can only express four kinds of secret code. These type of code are named as ‘00’, ‘01’, ‘10’, ‘11’ with the length of $log_2 4$. The secret code extended to 3 bits has 2^3 states in all. Thus, there is no way to embed all of eight kinds of code into 3-bit cell by using matrix encoding algorithm.

Actually, there is indeed a way to extend but require to select some extended codes elaborately. The extension appears to be conditional. Since 3-bit modifiable cell is only able to express four states, we still have a half opportunity to extend by selecting four extended codes to embed from eight codes[6]. During calculating the result of hash function, we can simply multiply the index i by 2 to extend 1 bit where we call it 1-layer extension. In this case, the codes ‘00’, ‘01’, ‘10’, ‘11’ are extended to ‘000’, ‘010’, ‘100’, ‘110’. In a similar way, we can multiply i by 2^2 to extend 2 bits called 2-layer extension. The rest may be deduced by analogy. L-Layer extension is performed by multiplying i by 2^L. Due to the closure property of XOR operation,

we can embed the secret message with more than 2-bit length into 3-bit cell, provided that the secret code is equal to any one of the specific extended codes. The mode of extension is illustrated in Fig. 1.

For extended algorithm, the coding mode implemented on the embedding cell is redefined by a quad (d_{max}, n, k, L) , wherethe new parameter L denotes the maximum of extension layer. Firstly, take out $(k + L)$-bit secret code $w = w_1w_2 \dots w_k \dots w_{k+L}$ from the whole secret message sequence to test if the secret code matches a specific extended code in the L-th layer. Thematching method is to test whether $mod(w, 2^L) = 0$ is true. If the remainder equals to zero, the extension layer of currentcell l_{crt} is L and a $(k + L)$-bit secret data will be able to be embedded successfully. If not, then continue to test if the prior $(k + L - 1)$-bit secret code $w = w_1w_2 \dots w_k \dots w_{k+L-1}$ matches a specific extended code in the $(L-1)$ -th layer by testing the resultof $mod(w, 2^{L-1})$. If $mod(w, 2^{L-1}) = 0$ is true, then the current extension layer is $l_{crt} = L - 1$ and the secret code $w = w1w2 \dots w_k \dots w_{k+L-1}$ will be embedded into this cell. But if not, continue to do this kind of test until we find out a matching code in a certain layer or there is no matching code in all extension layers. In latter case, the extended algorithm rolls back to the standard matrix encoding. The final embeddable secret code is in the form of $w = w_1w_2 \dots w_k \dots w_{k+l_{crt}}$. If no extension takes place, the layer of current cell is $l_{crt} = 0$.

In extended algorithm, the hash function is updated as Formula (1):

$$f: f(a) = \bigoplus_{i=1}^{n} a_i.(i.2^{l_{crt}}) \qquad (1)$$

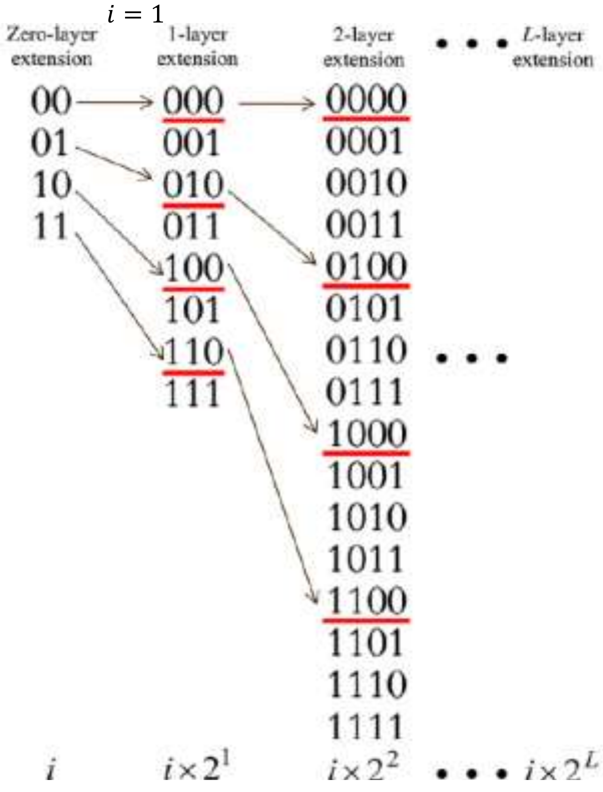


Fig -1: The chart of the extension mode.

Subsequently, implement XOR operation on the result of hash function and secret message $w = w1w2 \dots w_k \dots w_{k+l_{crt}}$ to obtain a decimal number y. At the moment, the range of index y has already been extended. We must shrink it to n by making$y$ divided by a coefficient $2^{l_{crt}}$.

$$y = \frac{w \oplus f(a)}{2^{l_{crt}}}, \qquad (2)$$

where the result of $w \oplus f(a)$ is expressed as a decimal number. Eventually, we obtain a stego cell a' by negating the y-th position in carrier cell a.

$$a' = \begin{cases} a & if\ y=0 \\ a_1a_1 \dots \bar{a}_y \dots a_n & otherwise \end{cases} \qquad (3)$$

From the above statement, it is implied that the introduction of extension mechanism raises a new problem to the receiver in detection process. The coding quad (d_{max}, n, k, L) can be confirmed and shared by the sender and the receiver before the start of the communication. Since the current layer l_{crt} is relative to the content of secret message, the receiver cannot predict this parameter definitely. Accordingly, the sender has to transfer l_{crt} to the receiver in the embedding process. We decide to append a symbol $s = s_1s_2 \dots s_m$ to the stego cell $a' = a_1a_2 \dots \bar{a}_y \dots a_n$ to mark the layer l_{crt}. Because the value of l_{crt} is fallen into the closed interval of [0,L], the length of symbol m can be calculated by Formula (4). We use the binary number of l_{crt} to assign the symbol s.

$$m = [log_2(L + 1)] \qquad (4)$$

Thus, the new stego cell c with the length of $(n + m)$ is composed of two parts, namely, data part and symbol part (i.e. the cell is reformed as $c = a's = a_1a_2 \dots \bar{a}_y \dots a_n s_1 s_2 \dots s_m$). In extraction phase, the receiver firstly take out the symbol part of the stego cell c and calculate the layerl_{crt}.

$$l_{crt} = dec(s_1s_2 \dots s_m) \qquad (5)$$

Eventually, the extended secret data $w = w_1w_2 \dots w_k \dots w_{k+l_{crt}}$ is retrieved by putting the data part of the stego cell c into the updated hash function $f: f(a) = \bigoplus_{i=1}^{n} a_i.(i.2^{l_{crt}})$

$$w = f(a') \qquad (6)$$

The detailed procedure of the extended matrix encoding is shown in Fig. 2.

To be more clear, we take the coding mode of (1,7,3,2) as an example to show how the extended algorithm works. Assume that the carrier data is $a = 1101010$, the secret data taken from the whole secret sequence is $w = 11001$.

First of all, the sender tests if $mod(dec(11001), 2^2) = 0$ is true. Due to $mod(dec(11001), 2^2) = 1$, the sender continue to testthe shorter secret data $w = 1100$. Since $mod(dec(1100), 2^1) = 0$ is true, it is confirmed that the secret data $w = 1100$ can be embedded into the carrier data and the current layer l_{crt}= 1.

Secondly, calculate the length of symbol $m = [log_2 3] = 2$ and assigns the symbol $s = 01$.

Thirdly, calculate the hash function with the carrier data $a = 1101010$ as shown as follows:

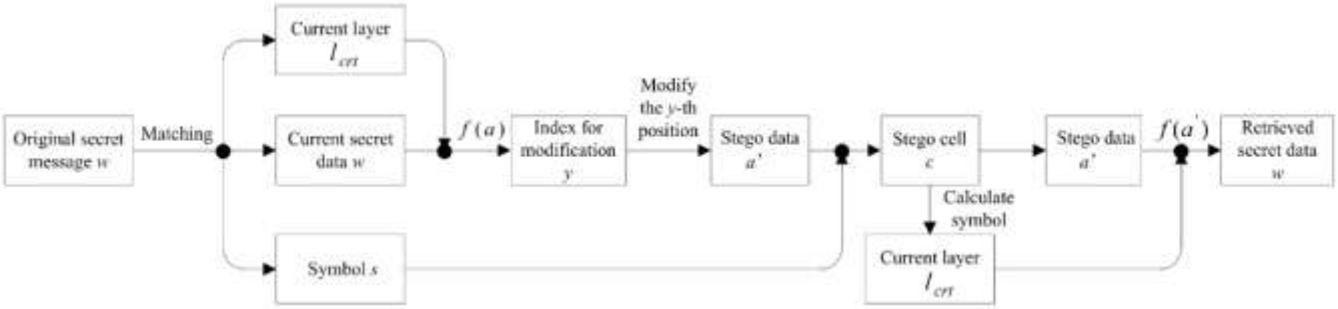


Fig -2: The flowchart of the extended matrix encoding.

$$\begin{array}{ll} & 0011 \\ & 0100 \\ & 1000 < \cdots a_i.(i.2^1) \\ f(a){:}\oplus & \underline{1100} \\ & 0011 \end{array} \qquad (7)$$

Finally, calculate index $y = (w \oplus f(a))/2 = ((1100)_2 \oplus (0010)_2)/2 = 7$ and flip the seventh bit position of carrier data $a = 1101010$ to generate a stego data $a' = 1101011$. Up to now, a stego cell $c = 110101101$ is obtained.

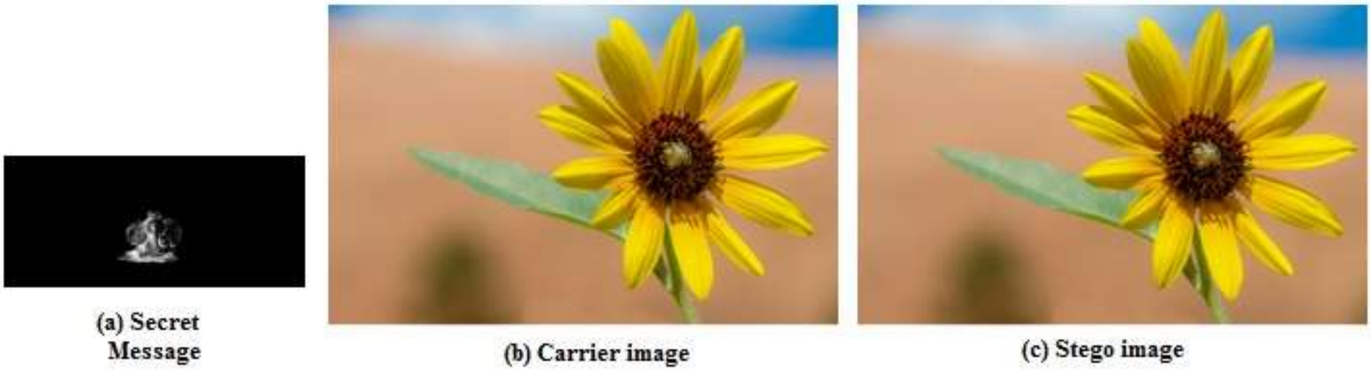
In detection process, the receiver firstly takes the symbol $s = 01$ from the stego cell and calculates the current layer $l_{crt} = dec(01) = 1$. Subsequently, calculate hash function with the stego data $a' = 1101011$ to retrieve the secret data$w = 1100$ as follows:

$$\begin{array}{ll} & 0010 \\ & 0100 \\ & 1000 \\ & 1100 \\ f(a'){:}\oplus & \underline{1110} \\ & 1100 \end{array} \qquad (8)$$

III. PROPOSED SYSTEM

From some experimental results, it can be seen that the embedding efficiency of extended algorithm is not always higher than F5. When the k to n ratio becomes small, the embedding efficiency gets decreased due to the symbol bits. The reason of this phenomenon is mainly that the changes for setting symbols do not load any secret message and a random-type secret message cannot always be extended as well. For the extended algorithm, this problem seems inevitable. While using binary images which are full of consecutive black pixels '00000000. . .' like the logos, the number of layers we can use for embedding the secrete data.

In the proposed system we can increase the embedding efficiency by using minimum layers. Fig. 4 gives the small description about the proposed system. To become algorithm more effective we are adding two more steps, firstly we will input our secrete data, it will compress the data, due to the compression, the layer required to embed the data get reduced. If we encrypt the compressed data, then at the receiver end the stegnalyst cannot understand the data after successfully fetching of data from the carrier media. At the receiver end the reverse process will be done like decryption of the data and uncompress the compressed data.


(a) Secret Message

(b) Carrier image

(c) Stego image

Fig -3: The performance of the proposed system

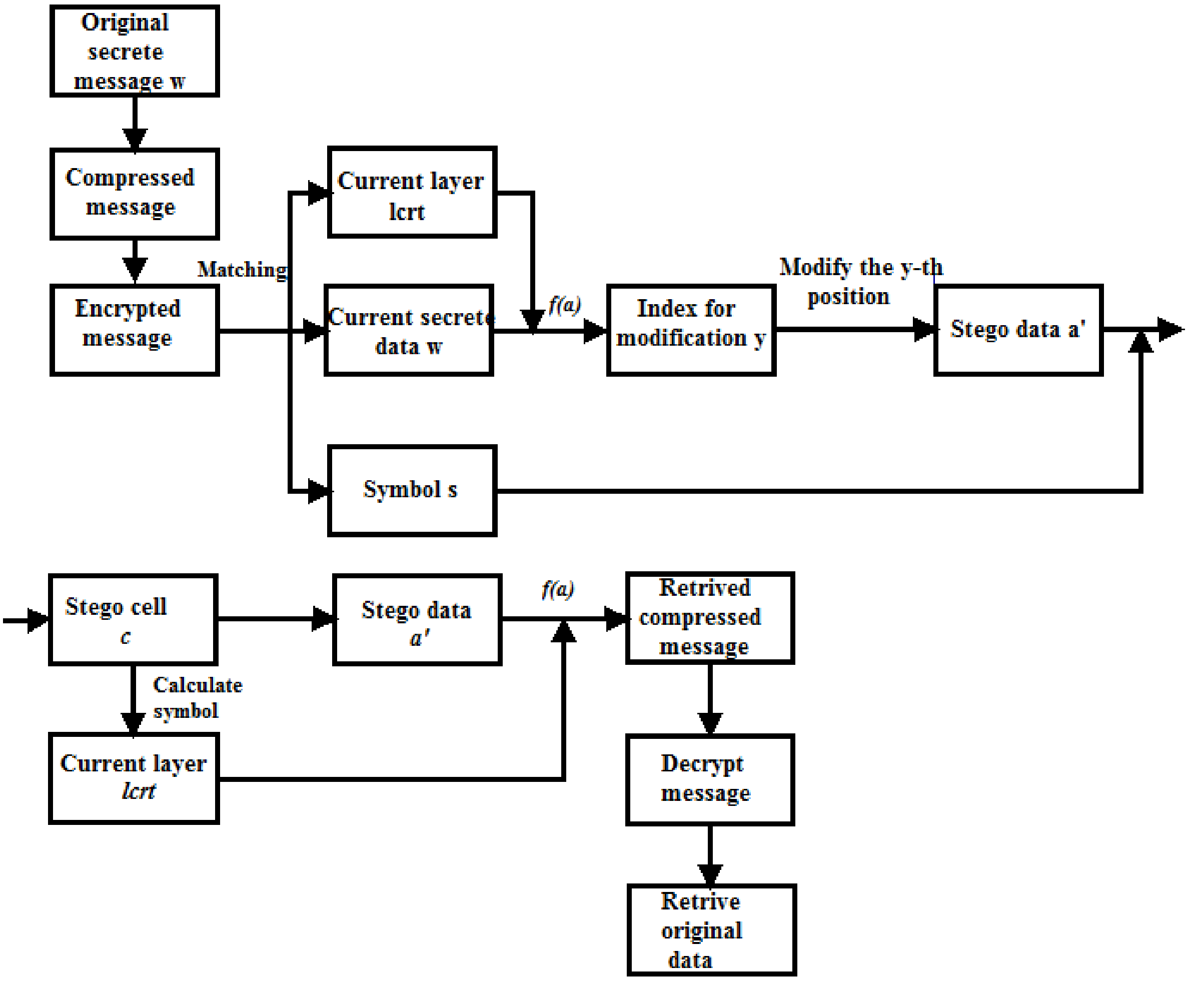


Fig -4: The architecture of the proposed system.

In order to test the performance of the proposed algorithm, we employ a meaningful image as the secret message and hide it into the carrier image flower shown in Fig. 3(a and b). The corresponding stego image is shown in Fig. 3(c).

The PSNR of original carrier image and stego image is same. The proposed system does not affect the PSNR value of the image.

From the figure we can see that the embedding efficiency of the proposed algorithm is higher than that of extended matrix encoding. Before compression as we embed the image in the cover image, in the result the number of modifiable bits and number of embedded bits are 38 and 64 respectively, the total file length is 2532 bytes. After the compression number of modifiable bits and number of embedded bits are 34 and 39 respectively and the total file length is 1787 bytes.

IV. CONCLUSION

This algorithm is mainly proposed to embed the secrete data by using minimum layer. Using this new algorithm we are providing more security to embedding data. In many cases the embedding rate will be 100%. In new algorithm the embedding efficiency will be increase as compare to the extended matrix encoding algorithm.

The secret message made of binary image has more opportunities to be extended.

REFERENCES

[1] Simmons GJ. The prisoners' problem and the subliminal channel. In: Advances in cryptology: proceedings of crypto 83, New York; 1984. p. 51–67.

[2] Van Schyndel RG, Tirkel A, Osborne CF. A digital watermark. In: Proc. of int. conf. on image processing, Austin; November 1994. p. 86–9.

[3] Franz E, Jerichow A, Moller S, Pfitzmann A, Stierand I. Computer based steganography: how it works and why therefore any restrictions on cryptography are nonsense, at best. In: Proc. of the 1st international workshop on information hiding, Cambridge; May 1996. p. 7–21.

[4] Zhang Tao, Ping Xijian. A fast and effective steganalytic technique against JSteg-like algorithms. In: Proc. 8th ACM symp. on applied computing, Florida; March 2003. p. 307–11.

[5] Westfeld A, Pfitzmann A. Attacks on steganographic systems. Lect Notes Comput Sci 2000;1768:61–75.

[6] Li Fan, Tiegang Gao, Qunting Yang. An extended matrix encoding algorithm for steganography of high embedding efficiency. Computers and Electrical Engineering 37 (2011) 973–981.