Derived Extended Matrix Encoding Algorithm for Better Embedding Efficiency

Prof. Arindam Das Gupta
Professor, Dept. of IT, AVCOE,
Sangamner, Maharashtra, India.
e-mail: arindam.dasgupta.family@gmail.com

Mr. Pritesh Kashinath Patil
Student of M.E. (IT), AVCOE,
Sangamner, Maharashtra, India.
e-mail: mr.pkpatil1989@gmail.com

***Abstract*- an extended matrix encoding algorithm is different from F5 algorithm. It increase the embedding efficiency, by using this algorithm the secrete message can be embed in carrier data. The quantized DCT coefficients of JPEG image are used as the embedding domain. By using this type of embedding domain, the secrete data will be secure from visual attack and statistical attack. In extended matrix encoding algorithm, the embedding efficiency and the embedding rate get increase to large extent by changing the hash function. In derived extended matrix encoding algorithm first, we are compressing the secrete data for minimization of necessary changes then encrypt it to provide more security before calculating the hash function.**

Keywords- DCT Coefficient, Embedding efficiency and rate.

I. INTRODUCTION

Many stegnographic algorithms consider the embedding capacity but ignore the embedding efficiency and security. By improving the matrix encoding of F5 stegosystem to increase the stegosystem L. Fan propose an extended algorithm, in which the embedding rate is essentially improved. Some changes in hash function in matrix encoding are performed to cover additional secret bits into definite range of cover bits. By exploiting the n-layer extension, at same time we can extend the embedding efficiency and embedding rate by converting the triple (d_{max}, n, k) to quad (d_{max}, n, k, L). To indicate current number of layer, the symbol positions are consider in every embedding process. This process makes the receiver understand blind detection effectively. In this paper we are compressing the secrete data to increase the efficiency of the algorithm. The rest of this paper is organized as follows. In Section two, the quality matrix encoding is represented. In the next section we represent the details of the extended algorithm proposed derived extended algorithm. In Section four, some simulation results and analysis are given, and final conclusion is provided in Section five.

II. BACKGROUND

A.Matrix encoding

The F5 scheme is developed with the help of F3 and F4 which are developed by JSteg, The F5 scheme is developed by Westfeld [1]. We insert the secret message by modifying corresponding LSB positions of quantized DCT coefficients of image[5][6]. Instead of classic LSB replacement or matching methods, the matrix encoding in F5 is used to implement the insertion and detection of secrete message.

Matrix encoding decreases the necessary number of changes. The description of the encoding process is given below:

1) Implement the JPEG lossy compression on the image which will be used as carrier image and obtain the quantized DCT coefficients for embedding [2].
2) The LSB plane of quantized DCT coefficients is partitioned into many embedding cells which are in the form of vector $a = a_1 a_2 \dots a_n$ with the length of n.

The coding which implemented on each embedding cell is denoted by an ordered triple (d_{max}, n, k). where, n is the number of modifiable bit positions in an embedding cell. k is the bit length of secret message $w = w_1 w_2 \dots w_k$ to be embedded into one cell. Also, an embedding cell with n positions will be changed in not more than d_{max} positions to embed k bits secret message. F5 implements matrix encoding only for $d_{max} = 1$. A hash function f is defined as Formula (1) to map n bits cover data a into k bits binary string.

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

where a_i is the i-th position of cell $a.i$ is the corresponding index number of the bit position and is in binary form during the operation. The bit length of binary i is same size as secret message w. Implement XOR operation on the value of hash function and secret message w to obtain a decimal number y.

$$y = w \oplus f(a) \quad (2)$$

By logically flipping the y-th bit position of cell a, an embedded stego data a' is generated. In special case, the carrier cell a' will be left intact when $y = 0$.

$$a' = \begin{cases} a & if\ y = 0 \\ a_1 a_1 \dots \bar{a}_y \dots a_n & otherwise \end{cases} \quad (3)$$

where, $\bar{a}_y$ is the negation of a_y.

In extraction phase, the receiver would retrieve the secret message w by directly putting the stego cell a'into the same hash function f.

For an embedding cell, the change density *D* refers to the proportion of altered bit position. Neglecting shrinkage, we can calculate the change density depending on Formula (4):

$$D = \frac{\frac{n}{n+1}.1+\frac{1}{n+1}.0}{n} = \frac{1}{n+1} \quad (4)$$

We can also get another important performance measure of steganographic algorithm called the embedding rate R:

$$R = \frac{k}{n} \quad (5)$$

We can define the embedding efficiency E which indicates the average bit number of embedded secret message per change:

$$E = \frac{R}{D} = \frac{n+1}{n}.k \quad (6)$$

Table 1 shows The theoretic values of the change density are given in Table 1. The values are calculated by using Formulas (4)–(6).

In F5 algorithm, the value of n and k must satisfy the equation $n = 2^k - 1$. Because k-bit binary sequence has 2^k potential states in all. The binary carrier cell has the length of $2^k - 1$ to express all states of k bits secret message by means of modifying only 1 bit or keeping unchanged.

Table 1:The performance of matrix encoding.

(n, k)	D(%)	R(%)	E(bit)
(1,1)	50	100	2
(3,2)	25	66.67	2.67
(7,3)	12.5	42.86	3.43
(15,4)	6.25	26.67	4.27
(31,5)	3.13	16.13	5.16
(63,6)	1.56	9.52	6.09
(127,7)	0.78	5.51	7.06
(255,8)	0.39	3.14	8.03
(511,9)	0.2	1.76	9.02

III. EXISTING SYSTEM

A. *Extended Algorithm*

How to increase k? In matrix encoding, the hash function maps n bits carrier data into a certain length of binary sequence which depends upon the bit length of index i. And i is chosen as the same size of secret message w. The extension of the length of i is realizable, we can finally embed more bits of secret message into one cell. Taking the cell (1,3,2) as an example, 2-bit secret message will be embedded into 3-bit embedding cell by altering only 1-bit position of the cell. The length of i is 2-bit. If the length of i is extended to be 3-bit, we can take one more 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, y will be used to indicate the position to be changed is out of the range. We may get $y = (101)_2 = 5$ in that case, but it is impossible to find out the fifth bit position for a cell of 3-bit length. Essentially modifying only 1 bit or keeping unchanged in the carrier cell with 3-bit length can only express four kinds of secret code. These kinds 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 [3].

Actually, there is indeed a way to extend but needed to select some extended codes elaborately. The extension seems to be conditional. Since 3-bit modifiable cell is only able to express four states, w still have a half opportunity to extend by selecting four extended codes to embed from eight codes. During evaluating 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,

The secret message with more than 2-bit length can be embedded 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), where the new parameter L denotes the maximum of extension layer. Firstly, take out $(k + L)$ -bit secret code $w = w_1 w_2 \dots w_k \dots w_{k+L}$ 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 $w = w_1 w_2 \dots w_k \dots w_{k+L-1}$ matches a specific extended code in the $(L - 1)$-th layer by testing the result of $mod(w, 2^{L-1})$. 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 $w = w_1 w_2 \dots w_k \dots w_{k+l_{crt}}$. If no extension takes place, the layer of current cell is $l_{crt} = 0$.

In extended algorithm, the hash function is updated as Formula (7):

$$f: f(a) = \bigoplus_{i=1}^{n} a_i . (i . 2^{l_{crt}}) \quad (7)$$

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 $2^{l_{crt}}$.

$$y = \frac{w \oplus f(a)}{2^{l_{crt}}}, \quad (8)$$

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.

$$a' = \begin{cases} a & if\ y=0 \\ a_1 a_1 \dots \bar{a}_y \dots a_n & otherwise \end{cases} \quad (9)$$

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 $s = s_1 s_2 \dots s_m$ to the stego cell $a' = a_1 a_2 \dots \bar{a}_y \dots a_n$ 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 (10). We use the binary number of l_{crt} to assign the symbol s.

$$m = [log_2(L + 1)] \quad (10)$$

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 $c = a's = a_1 a_2 \dots \bar{a}_y \dots a_n s_1 s_2 \dots s_m$).

In extraction phase, the receiver firstly take out the symbol part of the stego cell c and calculate the layer l_{crt}.

$$l_{crt} = dec(s_1 s_2 \dots s_m) \quad (11)$$

Eventually, the extended secret data $w = w_1 w_2 \dots w_k \dots w_{k+l_{crt}}$ is retrieved by putting the data part of the stego cell c intothe updated hash function

$$f: f(a) = \bigoplus_{i=1}^{n} a_i . (i . 2^{l_{crt}})$$
$$w = f(a') \quad (12)$$

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:

$$f(a): \begin{array}{l} 0011 \\ 0100 \\ 1000 < \cdots a_i.(i.2^1) \\ \oplus\ 1100 \\ \hline 0011 \end{array} \qquad (13)$$

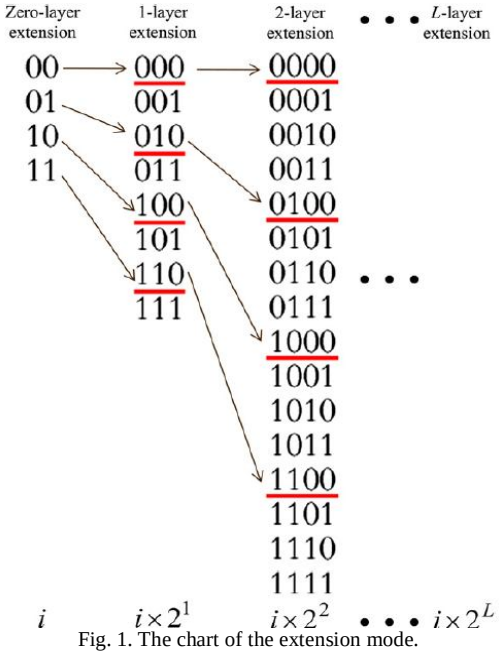

Fig. 1. The chart of the extension mode.

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:

$$f(a'): \begin{array}{l} 0010 \\ 0100 \\ 1000 \\ 1100 \\ \oplus\ 1110 \\ \hline 1100 \end{array} \qquad (14)$$

B. Simulation and Analysis

In order to test the performance of the proposed algorithm, we employ a meaningful binary logo image with 64 by 64pixels as the secret message and hide it into the carrier image Lena with 256 by 256 pixels 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 70.33. It is indicated that the extended method has a strong covertness

From the figure we can see that the embedding efficiency of the extended algorithm is higher than that of matrix encoding when the ratio of k to n is larger. However, when the ratio is less than 3/7, the efficiency of extended algorithm with $L = 7$ starts to be lower than that of standard algorithm. When the ratio is less than 4/15 and 5/31, the efficiency of extended algorithm with $L = 15$ and 31 starts to be lower than that of standard algorithm, respectively. Although the extended algorithm increases the bit number of embedded secret message, the modification for setting symbol positions affect the embedding efficiency because these changes do not load any secret message. Actually, the bit number of average changes for setting m-bit symbol is $m/2$ bits.

For every embedding cell, there are average $m/2$ bits modification used to set symbol rather than loading secret message. With the decrease of the ratio of k to n, the length of carrier data n becomes large which leads to the significant reduction of the number of embedding cells generated by partitioning the LSB plane of quantized DCT coefficients. When

the ratio of k to n is large, more secret data can be embedded and a large number of changes exist inside the cover data. The influence of symbol modification is relatively little. Although we can improve the embedding efficiency by increasing the maximal extension layer L, the increase of L will result in the increase of length of symbol which has a negative impact on the efficiency. Thus, we should findout the optimal tradeoff to confirm the coding mode $(1, n, k, L)$ according to the size and type of secret message. Taking the coding mode $(1,3,2, L)$ for example, a new logo image will be embedded into the test image using the extended scheme. It can be seen that the values of embedding efficiency vary in according to the maximum of extension layer L when n and k are specific.

In this case, the maximum of embedding efficiency is obtained when $L = 15$.When L is less than 15, the secret bits are not extended adequately. Contrarily, a larger L leads to the increase of the length of symbol which has a negative impact on the embedding efficiency. Obviously, the coding mode (1,3,2,15) is the best choice in this instance.

In matrix encoding, the length of carrier cell n is greater than the length of secret message k. It denote that the embedding rate is smaller than 100% for ever. In extended algorithm it is possible that the length of secret message$(k + l_{crt})$ is larger than the length of carrier cell $(n + m)$ due to the extension. Theoretically, the embedding rate can exceed 100% and the maximum value can be close to R_{max}.

$$R_{max} = \frac{L_s}{n+[log_2(L_s-k+1)]}, \quad (15)$$

Where Ls denotes the length of the whole secret message.

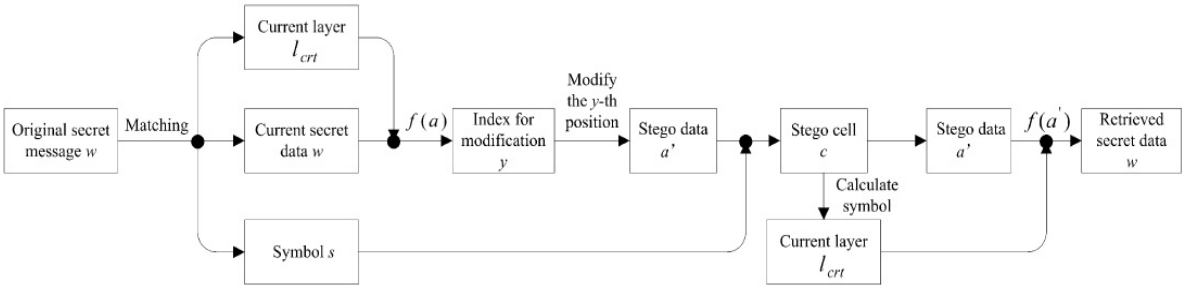

Fig. 2. The flowchart of the extended matrix encoding.

Fig. 3. The performance of the extended algorithm

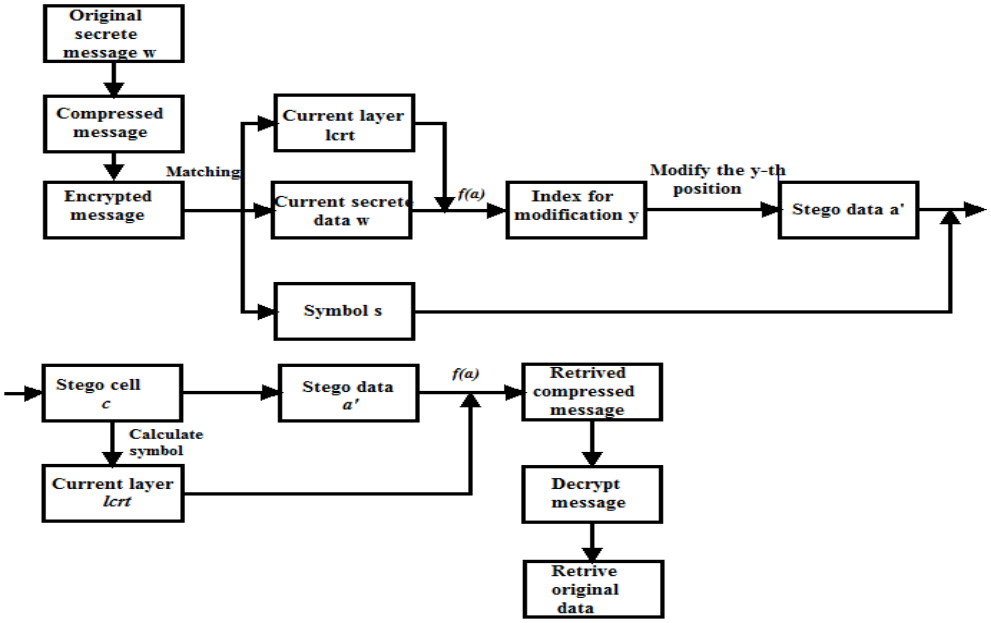

Fig. 4. Architecture of proposed system

IV. 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 of the data 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.

V. CONCLUSION

The derived matrix encoding 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 with a large number of extension layers.

REFERENCES

[1] Westfeld A. F5-a steganographic algorithm: high capacity despite better steganalysis. Lect Notes Comput Sci 2001;2137:289–302.

[2] Wallace GK. The JPEG still picture compression standard. IEEE Trans Consum Electron 1992;38(1):xviii–xxiv.

[3] Li Fan, Tiegang Gao, Qunting Yang. An extended matrix encoding algorithm for steganography of high embedding efficiency.

[4] Solanki K, Sarkar A, Manjunath B. YASS: yet another steganographic scheme that resists blind steganalysis.

[5] Lu Wei, Sun Wei, Lu Hongtao. Robust watermarking based on DWT and nonnegative matrix factorization. Comput Electr Eng 2009;35(1):183–8.

[6] Al-Otum HA, Al-Taba'a AO. Adaptive color image watermarking based on a modified improved pixel-wise masking technique. Comput Electr Eng 2009;35(5):673-95.