STEGNOGRAPHY OF HIGH EMBEDDING EFFICIENCY BY USING AN EXTENDED MATRIX ENCODING ALGORITHM

Borkar Bharat Sampatrao[1], Patil Pritesh Kashinath[2]

[1]Assistant Professor, [2]Student, Department of IT, AVCOE, Sangamner, Maharashtra, India
borkar.bharat@gmail.com, mr.pkpatil1989@gmail.com

Abstract

F5 Steganography is way totally different from most of LSB replacement or matching steganographic schemes, as a result of matrix encryption is used to extend embedding potency while reducing the amount of necessary changes. By victimisation this theme, the hidden message inserted into carrier media observably is transferred via a safer imperceptible channel. The embedding domain is that the quantitative DCT coefficients of JPEG image, which makes the theme, be proof against visual attack and statistical attack from the steganalyst. Based on this effective theme, An extended matrix encoding algorithm is planned to improve the performance further in this paper. The embedding potency and embedding rate get accrued to large extent by changing the hash function in matrix encryption and changing the coding mode. Eventually, the experimental results demonstrate the extended algorithm is more advanced and efficient to the classic F5 Steganography.

Index Terms: *Steganography, LSB replacement, DCT coefficient, Hash function.*

--*--**

1. 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 withstanding 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].

2. BACKGROUND

To improve the encoding step by scattering the embedding locations over the complete DCT-domain in keeping with a pseudo-random number generator, the Outguess0.1 is developed [6]. The randomly distributed data cannot detected by the X^2 -test for JSteg successfully. The creator of Outguess0.1 releases a revised version soon. The new algorithm known as Outguess0.2 tries to make sure that the statistics properties of the cover image may be maintained after encoding [7]. Shortly, Fridrich et al. use the discontinuity of the border between 2 adjacent eight by eight blocks to accurately estimate the length of the hidden information embedded with Outguess0.2 [8]. As another to the Outguess0.2 algorithm, Westfeld invent the more secure algorithm called F3 algorithm[9]. This method compares the absolute value of the coefficients with the secret message. If the absolute value is same, then no modification is done. Otherwise, The corresponding DCT coefficient's absolute value is reduced by one. F3 steganography eliminates the POVs problem which exists in JSteg by using above embedding strategy. In F3 the distribution of histogram looks unnatural because in F3 there are more even coefficients are introduced into histrogram than odd coefficients. Hence, to deal with unstructured frequency distribution of histogram an improvement called F4 is developed. In F4, even-negative coefficients and odd-positive coefficients represent one and odd-negative coefficients and even-positive coefficients stands for zero. If the secret bit value is same as the symbolic coefficient, then no changes happen. Otherwise, absolute value of the corresponding DCT coefficient is reduced by one. With the assistance of this embedding strategy, the histogram of stego image looks statistically like a clean image. Additionally, there are still several steganographic algorithms proposed like model-based [10,11], Perturbed quantization [12], YASS [13] so on [14–18].

Nonetheless, Many steganographic algorithms consider the embedding capacity but but neglect the embedding efficiency and security. A number of them simply work oppositely. During this paper, by improving the matrix encoding of F5 stegosystem to increase the stegosystem we propose an extended algorithm, while the embedding rate is essentially improved likewise. 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, we convert the triple (d_{max}, n, k) to quad (d_{max}, n, k, L) to extend the embedding efficiency and embedding rate at same time. To indicate current number of layer, the symbol positions are deployed in every embedding process. This mechanism makes the receiver understand blind detection effectively. 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 proposed extended algorithm. In Section four, some simulation results and analysis are given, and final conclusion is provided in Section five.

3. EXISTING SYSTEM

3.1 Matrix Encoding Basics

The F5 scheme is developed with the help of F3 and F4 which are developed by JSteg, The F5 scheme is developed by Westfeld [9]. We insert the secret message by modifying corresponding LSB positions of quantized DCT coefficients of image, JPEG image as well. Instead of classic LSB replacement or matching methods, the matrix encoding in F5 are used to implement the insertion and detection of secrete message. To enhance the security of stegosystem, the additional 'permutative straddling' technique is used to scatter the secret message over the whole carrier media.

Matrix encoding, as a lower rate method improves the embedding efficiency to a great extent than the classic LSB modification methods. It is introduced by Crandall [19]. The embedding efficiency depends on, how many bits of secret message can be loaded by one change taken place in carrier media. Reference from the LSB method, assume that the secret message and the values at the positions to be changed subject to the 0–1 uniform distribution. The positions which are to be embedded modified with a probability of 0.5 on an average. Means we have an embedding efficiency of 2 bits per change. If we want to fully embed the whole cover image, 50% of the carrier data will be changed. And it is so usefull in a high possibility of statistical stegnalysis. By reducing the density of changes in the cover image we reduce the possibility of detection, this is the most obvious and well known way as ever. Matrix encoding decreases the necessary number of changes (i.e the change density), as well as it increases the embedding efficiency. The description of the encoding process is given below:

1) Implement the JPEG lossy compression on the carrier image and obtain the quantized DCT coefficients for embedding [20].
2) The LSB plane of quantized DCT coefficients is partitioned into many embedding cells which are in the form of vector $a = a_1 a_2 \ldots a_n$ 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, namely, the length n. k is the bit length of secret message $w = w_1 w_2 \ldots w_k$ to be embedded into one cell. Also, an embedding cell with n positions will be changed in not more than dmax 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 denotes 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 selected as the same size as secret message w. Subsequently, implement XOR operation on the value of hash function and secret message w to obtain a decimal number y.

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

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$y = 0$.

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

Where, $\bar{a}_y$ is the negation ofa_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. The detailed procedure of matrix encoding is given in Fig. 1.

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):

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

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

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

Using the change density and embedding rate, we can calculate the embedding efficiency E which indicates the average bit number of embedded secret message per change:

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

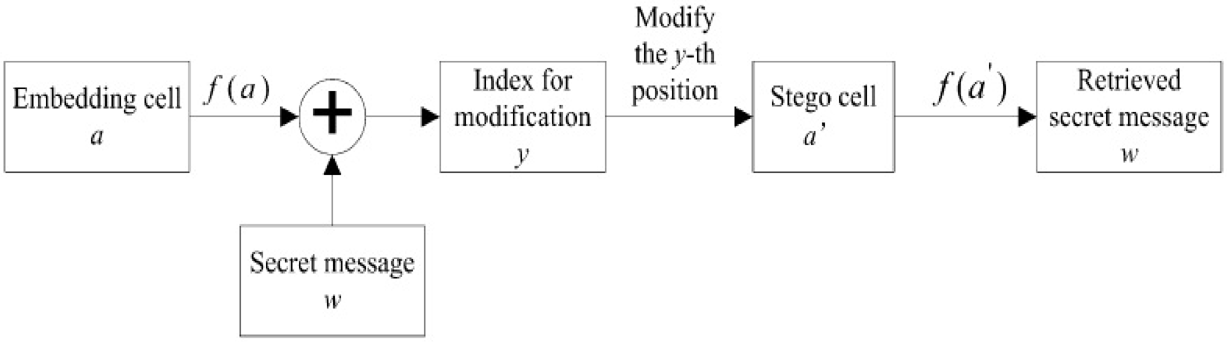


Fig -1: The block diagram of matrix encoding.

The theoretic values of the change density are given in Table 1, the embedding rate and embedding efficiency in according to various of (n,k) pairs. 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 possible states in all. It is inevitable that the binary carrier cell has the length of $2^k - 1$ to show all states of k bits secret message by means of modifying only 1 bit or keeping unchanged. That is to say, the length of embedding cell increases with the increase of the length of embedded secret message in manner of exponential growth. Meanwhile, the modification of carrier cell maintains 1-bit-change. Thus, the embedding rate will getting less and less with increase of efficiency. The value of higher efficiency is the lower rate. This regulation not only reveals how matrix encoding works, but also naturally inspires us that we are able to improve both the embedding efficiency and the embedding rate at the same time if k increases in the context of fixing on n and the number of change.

4. PROPOSED SYSTEM

4.1 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-bitembedding 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. 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 iby 2^L. Due to the closure property of XOR operation,

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

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. 2.

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 $w = w_1 w_2 \ldots w_k \ldots w_{k+L}$ 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 $\mathrm{mod}(w, 2^L) = 0$ 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 $w = w_1 w_2 \ldots w_k \ldots w_{k+L-1}$ matches a specific extended code in the $(L - 1)$-th layer by testing the resultof $\mathrm{mod}(w, 2^{L-1})$. If $\mathrm{mod}(w, 2^{L-1}) = 0$ is true, then the current extension layer is $l_{crt} = L - 1$ and the secret code $w = w1w2 \ldots w_k \ldots w_{k+L-1}$ 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 $w = w_1 w_2 \ldots w_k \ldots w_{k+l_{crt}}$. 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):

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

Subsequently, implement XOR operation on the result of hash function and secret message $w = w1w2 \ldots w_k \ldots w_{k+l_{crt}}$ 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 makingy divided by a coefficient $2^{l_{crt}}$.

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

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.

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

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 $s = s_1 s_2 \ldots s_m$ to the stego cell $a' = a_1 a_2 \ldots \bar{a}_y \ldots a_n$ 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.

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

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. thecell is reformed as $c = a's = a_1 a_2 \ldots \bar{a}_y \ldots a_n s_1 s_2 \ldots s_m$). In extraction phase, the receiver firstly take out the symbol part of thestego cell c and calculate the layerl_{crt}.

$$l_{crt} = \mathrm{dec}(s_1 s_2 \ldots s_m) \qquad (11)$$

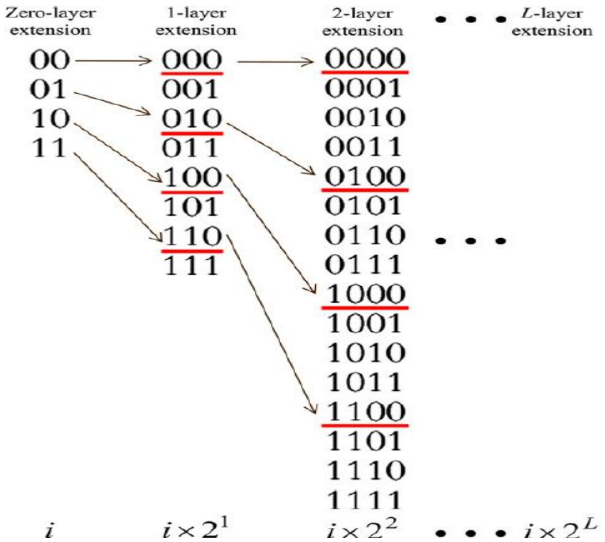


Fig -2: The chart of the extension mode.

Eventually, the extended secret data $w = w_1 w_2 \ldots w_k \ldots w_{k+l_{crt}}$ is retrieved by putting the data part of the stego cell c intothe updated hash function $f: f(a) = \bigoplus_{i=1}^{n} a_i.(i.2^{l_{crt}})$

$$w = f(a') \qquad (12)$$

The detailed procedure of the extended matrix encoding is shown in Fig. 3.

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 $\mathrm{mod}(\mathrm{dec}(11001), 2^2) = 0$ is true. Due to $\mathrm{mod}(\mathrm{dec}(11001), 2^2) = 1$, the sender continue to testthe shorter secret data $w = 1100$. Since $\mathrm{mod}(\mathrm{dec}(1100), 2^1) = 0$ 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:

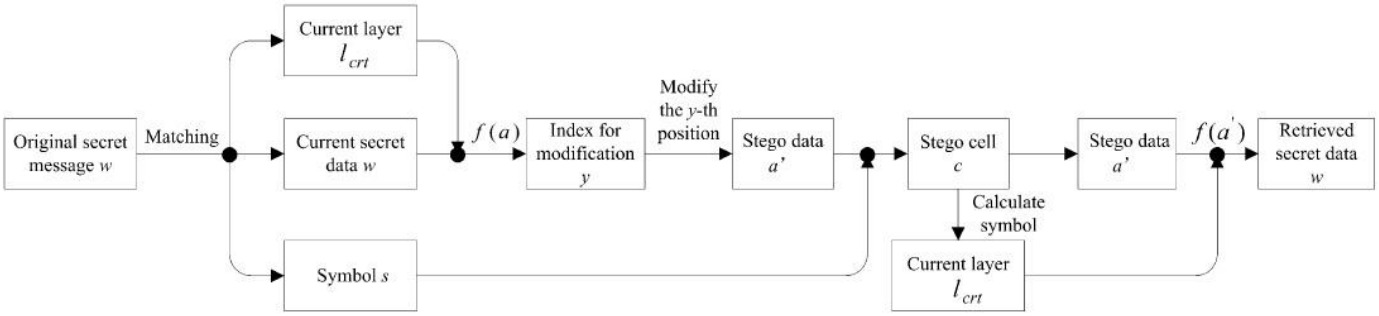


Fig -3: The flowchart of the extended matrix encoding.

$$\begin{array}{ll} & 0011 \\ & 0100 \\ & 1000 < \cdots a_i.(i.2^1) \\ f(a):\oplus & \underline{1100} \\ & 0011 \end{array} \tag{13}$$

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 $l_{crt} = \mathrm{dec}(01) = 1$. Subsequently, calculate hash function with the stego data $a' = 1101011$ to retrieve the secret dataw $= 1100$ as follows:

$$\begin{array}{ll} & 0010 \\ & 0100 \\ & 1000 \\ & 1100 \\ f(a'):\oplus & \underline{1110} \\ & 1100 \end{array} \tag{14}$$

5. ARCHITECTURE

5.1 Simulation and Analysis

In order to check the performance of the proposed algorithm, we tend to use 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. 4(a and b). The corresponding stego image is shown in Fig. 4(c).

The PSNR of original carrier image and stego image is 70.33. It is indicated that the extended method has a strong covertness. Fig. 5 illustrates the difference between the embedding efficiency of standard matrix encoding and that of extended algorithm with three different maximal extension layer (respectively $L = 7,15,31$).

From the figure it can be seen that the embedding efficiency of the extended algorithm is greater 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 kto n, the length of carrier data n becomes large which leads to the significant reduction ofthe number of embedding cells generated by partitioning the LSB plane of quantized DCT coefficients. In other words, the proportion of altered bits in carrier data is very small. In this case, the influence of symbol modification emerges and the cost of changing without loading effective information cannot be neglected. Oppositely, when the ratio of k to n is large, moresecret data can be embedded and a large number of changes exist inside the cover data. The influence of symbol modificationis 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 illustrated in Fig. 6(a) will be embedded into the test image Lena using the extended scheme. It can be seen from Fig. 6(b) that the values of embedding efficiency vary in according to the maximumof 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.

(a) secret message

(b) carrier image

(c) stego image

Fig -4: The performance of the extended algorithm

In matrix encoding, the length of secret message k is always smaller than the length of carrier cell n . It means the embedding rate is smaller than 100% for ever. However, it is possible for the extended algorithm 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} .

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

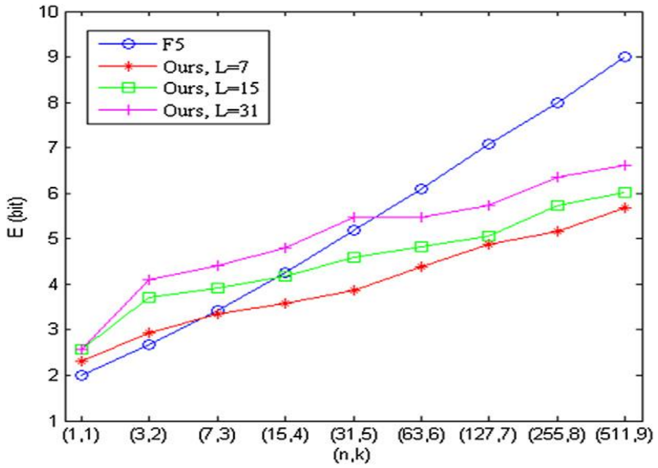
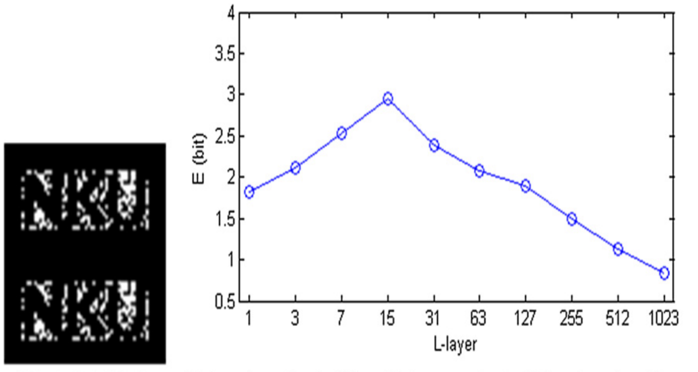


Fig -5: The comparison of the embedding efficiency.



(a) new secret logo (b) the values of embedding efficiency varying in different number of layer

Fig -6: The relationship between L and E when n and k are specific.

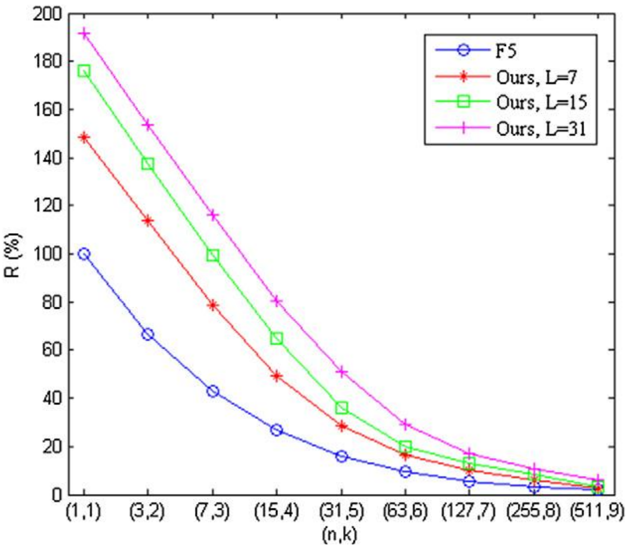


Fig -7: The comparison of the embedding rate.

Where Ls denotes the length of the whole secret message. Fig. 7 shows the comparison between the embedding rate of standard matrix encoding and that of extended algorithm with three different maximal extension layer (respectively L = 7,15,31).

CONCLUSION

Many steganographic algorithms provide a high capability for hidden messages however neglect the embedding efficiency. In fact, fewer modifications might guarantee the high covertness of the stegosystem. F5 steganography employs matrix encoding to get higher efficiency than classic LSB replacement or matching methods. However, with the increase of the efficiency, the embedding rate is decreased rapidly. In high efficiency case, a small amount of secret message can be embedded that leads to a waste of carrier data. The extended matrix encoding algorithm proposed in this paper uses n-layer extension to enhance the embedding efficiency and rate simultaneously. In the application of high ratio of kto n, the extended algorithm can obtain a better efficiency. Moreover, the extended algorithm makes the embedding rate exceed 100% theoretically and realizes a high capacity whilst performing a high efficiency.

From some experimental results, e.g. in Fig. 5, it can be seen that the embedding efficiency of improved scheme is not always higher than F5 all the time. When the ratio of k to n becomes small, the embedding efficiency gets decreased due to the introduction of 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 (actually with the possibility of 0.5). For the new method, this problem seems inevitable. In spite of this, our scheme is still important and meaningful because we also use binary images which are full of consecutive black pixels ‘00000000. . .’ like the logos in Figs. 4 and 6. This kind of secret message has more opportunities to be extended with a large number of extension layers. The experimental results demonstrate that the extended algorithm has a high covertness, a high embedding efficiency and a high embedding rate in the most cases. And the scheme is effective enough to be applied to the area of covert communication.

REFERENCES:

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

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

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

[04]. 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.

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

[06].Provos N. Defending against statistical steganalysis. In: Proc. of the 10th USENIX security symposium, Washington, DC; August 2001. p. 323–35.

[07].Provos N, Honeyman P. Hide and seek: an introduction to steganography. IEEE Secur Privacy 2003;1:32–44.

[08].Fridrich J, Goljan M, Hogea D. Attacking the OutGuess. In: Proc. of the 3rd information hiding workshop on multimedia and security, Juan-les-Pins; December 2002. p. 3–6.

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

[10].Sallee P. Model-based steganography. In: Proc. of the 2nd international workshop on digital watermarking, Seoul; October 2003. p. 154–67.

[11].Sallee P. Model-based methods for steganography and steganalysis. Int J Image Graphics 2005;5(1):167–90.

[12] Fridrich J, Goljan M, Soukal D. Perturbed quantization steganography with wet paper codes. In: Proc. of ACM multimedia workshop, Magdeburg; September 2004. p. 4–15.

[13] Solanki K, Sarkar A, Manjunath B. YASS: yet another steganographic scheme that resists blind steganalysis. Lect Notes ComputSci 2008;4567:16–31.

[14] Wang Xiangyang, Yang Yiping, Yang Hongying. Invariant image watermarking using multi-scale Harris detector and wavelet moments. ComputElectrEng 2010;36(1):31–44.

[15] Lu Wei, Lu Hongtao, Chung Fulai. Feature based robust watermarking using image normalization. ComputElectrEng 2010;36(1):2–18.

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

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

[18] Fan Li, GaoTiegang, Yang Qunting. A novel watermarking scheme for copyright protection based on adaptive joint image feature and visual secret sharing. Inf Control. 2011;7(7):3679–94.

[20] Crandall R. Some notes on steganography. Posted on steganography mailing list; 1998. http://os.inf.tu-dresden.de/~westfeld/crandall.pdf.

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

BIOGRAPHIES:

Borkar Bharat Sampatraoreceived his B.E and M.E degrees in Computer Science and Engineering from Pune University, He is a member of ISTE, ACM, IE. He published 3 National and 2 International papers. He presented 3 papers in national conference and 2 papers in international conference.His research interests includeImage processing.

Pritesh Kashinath Patil his B.E. in information Technology from North Maharashtra University, in 2012. He is currently doing his M.E. in Information Technology from University of Pune.