Journal of Imaging Science and Technology® 51(4): 380–385, 2007.

Error Spreading Control in Image Steganographic Embedding Schemes Using Unequal Error Protection

Ching-Nung Yang, Guo-Jau Chen and Tse-Shih Chen
CSIE Dept., National Dong Hwa University, #1, Da Husueh Rd., Sec. 2, Hualien, Taiwan
E-mail: cnyang@mail.ndhu.edu.tw

Rastislav Lukac
The Edward S. Rogers Sr. Department of ECE, University of Toronto, 10 King's College Road, Toronto, Ontario, M5S 3G4 Canada

Abstract. *A steganographic scheme proposed by van Dijk and Willems can alter a relatively small amount of bits to hide the secret compared to other schemes while reducing the distortion and improving the resistance against the steganalysis. However, one bit error in the embedding scheme by van Dijk and Willems may result in multibit error when extracting the hidden data. This problem is called as error spreading. It is observed that only some single-bit errors suffer from error spreading. In this paper, we propose a new steganographic solution which takes advantage of unequal error protection codes and allows for the different protection of different secret bits. Thus the proposed solution can effectively protect bits which could suffer from error spreading. In addition, it saves parity bits, thus greatly reducing the amount of bit alterations compared to the relevant previous schemes. Experimentation using various test images indicates that the proposed solution achieves the tradeoff between the performance and the protection of the embedded secret.* © *2007 Society for Imaging Science and Technology.*
[DOI: 10.2352/J.ImagingSci.Technol.(2007)51:4(380)]

INTRODUCTION

Steganography is a method of hiding or embedding the secret message into a cover media to ensure that an unintended party will not be aware of the existence of the embedded secret. Popular steganographic techniques for visual data protection embed the secret message such as a binary image by manipulating the least significant bit (LSB) plane of a cover image, thus producing the so-called stegoimage. In the simplest form, the data embedding can be realized by $c_0 \oplus s$ with c_0 denoting the least significant bit of a pixel from the cover image and s denoting the secret bit.

A more efficient steganographic method was proposed by van Dijk and Willems.[1] Their *coded LSB* method attempts to reduce the distortion when the noise (e.g., due to faulty communication channels) or active attacks by the third party (intentional modifications of some insignificant bits in a cover image to prevent the extraction of hidden secret by the authorized user) are introduced into the stegoimage. However, in some situations, one error bit produced during the stegoimage transmission phase or by active attacks often results in two or more errors when decoding (extracting) the embedded message. This phenomenon, known as an *error spreading* problem, affects the clearness of the extracted secret image. A straightforward application of error correction, i.e., using stegoencoding to hide the secret first and then adding the parity to provide the *error correcting* (EC) capability, will inevitably increase the required amount of bit alterations. Thus, the risk of being detected will increase. Zhang and Wang[2] proposed a new stegoencoding approach combining the coded LSB and EC capability simultaneously to address the error spreading problem. Their solution has the same error correcting capability for all protected bits, but according to our observation the error spreading does not affect all spatial locations of the secret image.

In this paper, we propose a more reasonable solution, *unequal error protection* (UEP) codes, to obtain the different protection ability for nonaffected and affected bits and to save parity bits. It will be shown that the proposed solution outperforms the previous relevant solutions in terms of the tradeoff between the performance and the protection of the embedded secret.

The rest of this paper is organized as follows. In the Coded LSB Scheme Section, the *coded LSB* scheme is described. In the EC Codes Based Error Correction: Zhang-Wang Scheme Section, Zhang-Wang stegoencoding based on EC codes is presented to show a solution for the error spreading problem. Our scheme based on UEP codes is proposed in the UEP Codes Based Error Correction the Proposed Scheme Section. Motivation and design characteristics are discussed in detail. In the Comparison and Experimental Results Section, the proposed method is tested using a variety of test images. The effect of UEP codes-based data embedding is evaluated and compared with the previous approaches. Finally, conclusions are drawn in the Conclusions Section.

CODED LSB SCHEME

In the plain LSB embedding scheme, the secret bits are hidden by simply replacing LSBs of the cover pixels. Due to the noiselike appearance of the LSB plane of natural images, embedding n bits implies, in average, the alteration of $n/2$ original LSBs. To reduce the number of altered LSBs and

Received Oct. 28, 2006; accepted for publication Mar. 22, 2007.
1062-3701/2007/51(4)/380/6/$20.00.

Table I. Eight cosets for the coded LSB scheme with $l=3$, $n=7$ using $(7,4)$ Hamming code with $G(x)=x^3+x^2+1$.

L_0	(0000000), (0001101), (0011010), (0010111), (0110100), (0111001), (0101110), (0100011), (1101000), (1100101), (1110010), (1111111), (1011100), (1010001), (1000110), (1001011)
L_1	(0000001), (0001100), (0011011), (0010110), (0110101), (0111000), (0101111), (0100010), (1101001), (1100100), (1110011), (1111110), (1011101), (1010000), (1000111), (1001010)
L_2	(0000010), (0001111), (0011000), (0010101), (0110110), (0111011), (0101100), (0100001), (1101010), (1100111), (1110000), (1111101), (1011110), (1010011), (1000100), (1001001)
L_3	(0000011), (0001110), (0011001), (0010100), (0110111), (0111010), (0101101), (0100000), (1101011), (1100110), (1110001), (1111100), (1011111), (1010010), (1000101), (1001000)
L_4	(0000100), (0001001), (0011110), (0010011), (0110000), (0111101), (0101010), (0100111), (1101100), (1100001), (1110110), (1111011), (1011000), (1010101), (1000010), (1001111)
L_5	(0000101), (0001000), (0011111), (0010010), (0110001), (0111100), (0101011), (0100110), (1101101), (1100000), (1110111), (1111010), (1011001), (1010100), (1000011), (1001110)
L_6	(0000110), (0001011), (0011100), (0010001), (0110010), (0111111), (0101000), (0100101), (1101110), (1100011), (1110100), (1111001), (1011010), (1010111), (1000000), (1001101)
L_7	(0000111), (0001010), (0011101), (0010000), (0110011), (0111110), (0101001), (0100100), (1101111), (1100010), (1110101), (1111000), (1011011), (1010110), (1000001), (1001100)

preserve the original features such as edges and fine details of the cover image, the coded LSB scheme of Ref. 1 divides the secret image into chips of l bits. Each l-bit chip is then embedded into LSBs of n pixels using (n,k) cyclic codes where n is the information code length and $l=n-k$.

Let $G_1(x)=\Sigma_{i=0}^{k} g_{1,i}x^i$ and $G_2(x)=\Sigma_{i=0}^{l} g_{2,i}x^i$ be two binary *polynomials* with degree k and l, respectively, so that $G_1(x)\cdot G_2(x)=x^n+1$. Using (n,k) cyclic codes with the generating function $G(x)=G_2(x)$, it is possible to construct 2^l code sets which consist of unique 2^k codewords. Thus, each code set can be used to describe one l-bit secret chip by choosing the nearest codeword to represent the embedded secret, as depicted in algorithm 1.

Algorithm 1

Inputs: secret message of l bits $s_{l-1}s_{l-2}\ldots s_0$, and cover image O with n LSBs $c_{n-1}c_{n-2}\ldots c_0$.

Output: stegoimage O' with n LSBs $c'_{n-1}c'_{n-2}\ldots,c'_0$.

Step 1: Choose one cyclic (n,k) code with the generating function $G(x)=g_lx^l+\cdots+g_1x+g_0$ and then select any k-tuples as the input to construct a code set (coset) of 2^k codewords. Choose one n-tuple codeword that does not appear in this coset, and then add an unused n-tuple to all the codewords in the coset to construct another coset.

Step 2: Repeat step 1, until all 2^n codewords are used. The process generates 2^l different cosets $L_0,L_1,\ldots,L_{2^l-1}$ that include 2^k codewords in each coset.

Step 3: Encrypt the secret bits $s_{l-1},s_{l-2},\ldots,s_0$ by choosing the coset L_i with $i=\Sigma_{i=0}^{l-1}s_i2^i$. Then, find the codeword $c'_{n-1}c'_{n-2}\ldots c'_0$ in the coset L_i such that the Hamming distance between $c'_{n-1}c'_{n-2}\ldots c'_0$ and $c_{n-1}c_{n-2}\ldots c_0$ is minimum.

Step 4: Deliver n LSBs $c'_{n-1}c'_{n-2}\ldots c'_0$ to the corresponding pixels in the embedded stegoimage O'.

As shown in Ref. 2, the efficiency of the steganographic schemes can be demonstrated using the so-called *embedding rate* (*ER*) which is defined as the number of embedded bits per pixel, i.e., $ER=l/n$. Another suitable criterion is the so-called *embedding efficiency* (*EE*) which is calculated as the number of altered bits per pixel, i.e., $EE=l/l_{\text{alt}}$ where l_{alt} denotes the average LSB alteration when l secret bits are embedded into n LSBs. The value l_{alt} can be calculated from all codewords in the cosets by a computer program. The *ER* parameter is suitable for discussing the embedded capacity whereas the *EE* parameter is used to evaluate the distortion in the cover image. It is obvious that for the plain LSB embedding scheme $ER=l/n=n/n=100\%$ and $EE=1/l_{\text{alt}}=n/n/2=200\%$. For the comparison purposes, the values corresponding to algorithm 1 (coded LSB scheme with $l=3$ and $n=7$) are provided below.

Let us assume algorithm 1 with $n=7$ and $l=3$, i.e., the objective is to embed three secret bits into seven LSBs. The above setting implies that $k=4$, resulting in $x^7+1=(x^4+x^3+x^2+1)(x^3+x^2+1)$ and $G(x)=x^3+x^2+1$. After the first two steps in *algorithm* 1, we construct eight cosets with sixteen codewords in each coset (see Table I). Suppose that 101 denotes the secret and 0001110 denotes the original set of LSBs. We use the coset L_5 to find the codeword 1001110 that has the minimum Hamming distance equal to one from 0001110 while altering only one LSB to embed three secret bits. The embedding rate and embedding efficiency are $ER=3/7=42.9\%$ and $EE=3/0.875=343\%$, respectively. Note that $l_{\text{alt}}=0.875$ for Table I. It is easy to see that the plain LSB embedding scheme $(ER=100\%, EE=200\%)$ has larger embedded capacity than the coded LSB scheme. On the other hand, the coded LSB scheme modifies fewer bits, thus reducing the distortion in the cover image.

However, the coded LSB scheme suffers from the error spreading problem. The occurrence of one error bit in the encoded LSBs may cause more than one bit error during the secret message extraction process. Considering the above example, we can use 0000000 to carry the secret 000. Suppose that there is one error bit, for example, 0010000. Then, according to Table I, the extracted secret is 111, i.e., there are three error bits. However, employing the error pattern 0000001 results in the extracted secret 001 which corresponds to only one error bit (no error spreading). This sug-

Table II. Eight cosets for the EC-based coded LSB scheme (Zhang-Wang Scheme) with $l=3$, $N_E=11$ and 1-error correcting capability using $(7,4)$ Hamming code and $(11,7)$ shorten Hamming code.

L_0	(00000000000), (10010001101), (10110011010), (00100010111), (11110110100), (01100111001) (01000101110), (11010100011), (01111101000), (11101100101), (11001110010), (01011111111) (10001011100), (00011010001), (00111000110), (10101001011)
L_1	(11100000001), (01110001100), (01010011011), (11000010110), (00010110101), (10000111000) (10100101111), (00110100010), (10011101001), (00001100100), (00101110011), (10111111110) (01101011101), (11111010000), (11011000111), (01001001010)
L_2	(01010000010), (11000001111), (11100011000), (01110010101), (10100110110), (00110111011) (00010101100), (10000100001), (00101101010), (10111100111), (10011110000), (00001111101) (11011011110), (01001010011), (01101000100), (11111001001)
L_3	(10110000011), (00100001110), (00000011001), (10010010100), (01000110111), (11010111010) (11110101101), (01100100000), (11001101011), (01011100110), (01111110001), (11101111100) (00111011111), (10101010010), (10001000101), (00011001000)
L_4	(10100000100), (00110001001), (00010011110), (10000010011), (01010110000), (11000111101) (11100101010), (01110100111), (11011101100), (01001100001), (01101110110), (11111111011) (00101011000), (10111010101), (10011000010), (00001001111)
L_5	(01000000101), (11010001000), (11110011111), (01100010010), (10110110001), (00100111100) (00000101011), (10010100110), (00111101101), (10101100000), (10001110111), (00011111010) (11001011001), (01011010100), (01111000011), (11101001110)
L_6	(11110000110), (01100001011), (01000011100), (11010010001), (00000110010), (10010111111) (10110101000), (00100100101), (10001101110), (00011100011), (00111110100), (10101111001) (01111011010), (11101010111), (11001000000), (01011001101)
L_7	(00010000111), (10000001010), (10100011101), (00110010000), (11100110011), (01110111110) (01010101001), (11000100100), (01101101111), (11111100010), (11011110101), (01001111000) (10011011011), (00001010110), (00101000001), (10111001100)

gests that if the error falls in the right three positions, i.e., 0000100, 0000010, and 0000001, then there is still only one error in the extracted secret and the damage is not expanded. However, for the other four error patterns 1000000, 0100000, 0010000, and 0001000, the decoded secret is 110, 011, 111, and 101, respectively, indicating that the extracted secret suffers from more than one error bit. Since such errors affect the quality of the extracted secret message, the scheme should be improved by using the error correcting mechanism, such as one described below based on EC codes.

EC CODES BASED ERROR CORRECTION: ZHANG-WANG SCHEME

Following the previous approach, a coded LSB scheme with $ER=l/n$ is constructed using the generating function $G(x)=G_2(x)$. Then, 2^k n-tuple vectors in each coset are encoded into an (N_E,n) cyclic EC code with N_E denoting the code length of EC codes. Although in this improved coded LSB scheme the embedding rate is reduced to $ER=l/N_E$, the scheme now has the error correcting capability of (N_E,n) cyclic EC codes.

Let us consider the previous example with one-error correcting capability and parameters $n=7$, $l=3$ and $N_E=11$. A $(7,4)$ Hamming code is used to embed three secret bits and a $(11,7)$ shorten Hamming code is used to achieve one error correcting capability. For example, embedding the secret 000 into the 7-tuple 0001101 first and then appending the parity 1001 forms the codeword 10010001101 which is listed as the second codeword in the coset L_0 (see Table II listing all eight generated cosets). If one error occurs in the sixth position, resulting in the codeword 10010101101, then the minimum Hamming distance is associated with the codeword 10010001101 from L_0 and the extracted secret is 000. Because of the error correcting capability of the $(11,7)$ shorten Hamming code, the error is always corrected no matter where the error bit occurs, thus overcoming the error spreading problem. On the other hand, the approach is less efficient than the conventional method, as it reduces the embedding rate from 42.9% to $3/11=27.3\%$ and also decreases the embedding efficiency from 343% to $3/2.625=114\%$ (note that $l_{\text{alt}}=2.625$ for Table II). Therefore, the different correction mechanism is needed. Since only the error in the first k bits of an n-tuple will produce additional errors in the secret extraction phase, it should be sufficient to ensure the validity of the first k bits instead of all n bits.

UEP CODES BASED ERROR CORRECTION: THE PROPOSED SCHEME

UEP codes, a category of EC codes, allow different protection for different bit locations.[3,4] In practice, some information bits are protected against a greater number of errors than other, less significant, information bits. Basically, a UEP code can be denoted as $[n,k,(d_1,d_2,\ldots,d_k)]$. By employing UEP codes to protect the message, the occurrence of no more than $\lfloor(d_i-1)/2\rfloor$ errors in the transmitted codeword does not affect the correctness of the ith bit in the decoded message.

It was noted that the first k bits of vectors need an enhanced protection to prevent the error spreading. Therefore, we propose to apply UEP codes to assure the correctness of these k bits and reduce the number of redundant parity bits. The main difference between the UEP-based scheme and the EC-based scheme relates to the use of

Table III. Eight cosets for the UEP-based coded LSB scheme (the proposed scheme) with $l=3$, $N_U=11$ using $[10,7,(3333222)]$ UEP code.

L_0	(0000000000)*(1100001101)(1000011010)(0100010111)(1000110100)(0100111001) (0000101110)(1100100011)(0001101000)(1101100101)(1001110010)(0101111111) (1001011100)(0101010001)(0001000110)(1101001011)
L_1	(0010000001)*(1110001100)(1010011011)(0110010110)(1010110101)(0110111000) (0010101111)(1110100010)(0011101001)(1111100100)(1011110011)(0111111110) (1011011101)(0111010000)(0011000111)(1111001010)
L_2	(0100000010)(1000001111)(1100011000)(0000010101)(1100110110)(0000111011) (0100101100)(1000100001)(0101101010)(1001100111)(1101110000)(0001111101) (1101011110)(0001010011)(0101000100)(1001001001)
L_3	(0110000011)(1010001110)(1110011001)(0010010100)(1110110111)(0010111010) (0110101101)(1010100000)(0111101011)(1011100110)(1111110001)(0011111100) (1111011111)(0011010010)(0111000101)(1011001000)
L_4	(1000000100)*(0100001001)(0000011110)(1100010011)(0000110000)(1100111101) (1000101010)(0100100111)(1001101100)(0101100001)(0001110110)(1101111011) (0001011000)(1101010101)(1001000010)(0101001111)
L_5	(1010000101)(0110001000)(0010011111)(1110010010)(0010110001)(1110111100) (1010101011)(0110100110)(1011101101)(0111100000)(0011110111)(1111111010) (0011011001)(1111010100)(1011000011)(0111001110)
L_6	(1100000110)(0000001011)(0100011100)(1000010001)(0100110010)(1000111111) (1100101000)(0000100101)(1101101110)(0001100011)(0101110100)(1001111001) (0101011010)(1001010111)(1101000000)(0001001101)
L_7	(1110000111)(0010001010)(0110011101)(1010010000)(0110110011)(1010111110) (1110101001)(0010100100)(1111101111)(0011100010)(0111110101)(1011111000) (0111011011)(1011010110)(1111000001)(0011001100)

$[N_U,n,(d_1,d_2,\ldots,d_k)$ UEP codes with N_U denoting the code length of UEP codes instead of (N_E,n) EC codes. Since the value of N_U is smaller than N_E, the proposed UEP-based coded LSB scheme will have the higher embedding rate while still providing the same protection of the secret to the error spreading.

As before, let us consider the scenario with one error correcting capability and parameters $n=7$, $l=3$, and $N_U=10$. Suppose that the $[10,7,(3333222)]$ UEP code is used to ensure the protection against errors. The corresponding eight cosets are listed in Table III. Assuming that the embedded secret and the encoded result are, for instance, 000 and 0000000000, respectively, one error can occur in the following cases:

- The presence of the error in the 7th bit (from right) implies the codeword 0001000000. As shown in Table III, there is only one codeword 0000000000 in the coset L_0 with the unit Hamming distance to 0001000000. In this case, the recovered secret is 000, i.e., no processing error.
- If the error affects the third bit (from right) beyond the error correcting capability of the considered UEP code, then 0000000000 in the coset L_0 and 1000000100 in the coset L_4 are the two codewords with the unit Hamming distance to the codeword 0000000100 under consideration. The decoding process can result in the extracted secret 000 or 100, respectively. Thus, even in the latter situation (i.e., 100), there is still only one error, suggesting no error spreading.
- Finally, the alteration of the 8th bit (from right) due to the error implies 0010000000 which will be decoded as 0000000000 in the coset L_0 or 0010000001 in the coset L_1. This suggests that the secret can be extracted as 000 or 001, respectively. Similar to the previous case, even when 001 is used as the extracted secret, the proposed method still overcomes the error spreading problem.

It is evident that when no more than one error falls in the first four bits of the original 7-bit vector, the use of UEP codes ensures that the first four bits will be correctly decoded and the error will be corrected, as $\lfloor(3-1)/2\rfloor=1$, i.e., the Hamming distance between the first four bits of two codewords is three providing one error correcting capability for the first four bits. However, no error spreading is also observed in the situations when one error occurs in other places. This is due to the fact that a single error in other places will result, in the worst case, in a single error in the decoded secret bits. The achieved embedding rate and embedding efficiency are $ER=l/N_U=30\%$ and $EE=l/l_{\text{alt}}=133\%$. Note that $l_{\text{alt}}=2.25$ for Table III.

By employing the familiar representation used in UEP codes, the (11,7) shorten Hamming code can be represented as $[11,7,(3333333)]$. Therefore, if we protect the first four bits only, then we can save one redundant checking bit by using the $[11,7,(3333222)]$ UEP code. Note that the error-correcting capability of $[N_U,n,(d_1,d_2,\ldots,d_k)]$ UEP codes is not better compared to $[N_E,n,d]$ EC codes. However, UEP codes have better embedding rate and embedding efficiency and also overcome the error spreading problem.

COMPARISON AND EXPERIMENTAL RESULTS

Different analytical tools, such as the sample pair analysis[5] and image quality metrics,[6] are used to analyze the steganographic solutions. To resist the various attacks on the stegoimage while still providing the required performance, an ideal steganographic scheme should be constructed by considering the relation between the cover image and the

Table IV. Comparison of the conventional, EC-based, and UEP-based coded LSB schemes for $n=2$ to 12.

Conventional scheme		Zhang-Wang EC-based scheme		Proposed UEP-based scheme	
(n,k,l)	$ER=l/n$	$[N_E,n,d]$	$ER=l/N_E$	$[N_U,n,(d_1,d_2,\ldots,d_k)]$	$ER=l/N_U$
(2,1,1)	50.0%	[5,2,3]	20.0%	[4,2,(32)]	25.0%
(4,1,3)	75.0%	[7,4,3]	42.9%	[6,4,(3222)]	50.0%
(5,4,1)	20.0%	[9,5,4]	11.1%	[8,5,(33332)]	12.5%
(6,4,2)	33.3%	[10,6,4]	20.0%	[9,6,(333322)]	22.2%
(7,4,3)	42.9%	[11,7,4]	27.3%	[10,7,(3333222)]	30.0%
(8,4,4)	50%	[12,8,4]	33.3%	[11,8,(33332222)]	36.4%
(9,4,5)	55.6%	[13,9,4]	38.5%	[12,9,(333322222)]	41.7%
(10,4,6)	60%	[14,10,4]	42.9%	[13,10,(3333222222)]	46.2%
(11,4,7)	63.6%	[15,11,4]	46.7%	[14,11,(33332222222)]	50.0%
(12,4,8)	66.7%	[17,12,5]	47.1%	[15,12,(333322222222)]	53.3%

stegoimage (or the relations between the stegopixel and the original pixel) and simultaneously the scheme should allow to achieve the high embedding rates. Therefore, the schemes under consideration are evaluated here in terms of the embedding rate, embedding efficiency, and the peak-signal-to-noise (PSNR) ratio calculated using the original cover image and its stegoversion.

Table IV shows the embedding rates and codes used in the conventional scheme, EC-based scheme and UEP-based scheme for $n=2$ to 12. As it can be seen from the listed results, all UEP-based schemes have the shorter code length than the EC-based schemes. Both these schemes have the ability to correct the first k bits when no larger than one error occurs, and avoid the error spreading problem. Table V shows the detail comparison for these three schemes with $(n,k,l)=(7,4,3)$. Code-based schemes address the error spreading problem at the cost of their smaller *ER* and *EE*. The UEP-based scheme, with $ER=30.0\%$ and $EE=133\%$, needs 3334 pixels in a cover image to embed 1000 $(=3334\times 30\%)$ secret bits while altering 750 LSBs $(=3334\times 2.25/12)$ within these embedded pixels; and, as can be seen, it outperforms the EC-based scheme.

In order to compare the distortion caused by the schemes under consideration, the well-known 259×259 test gray-scale images "Baboon", "Barb", "Boat", "Elaine", "Mena", and "Peppers" have been used as the cover images. The secret NDHU (National Dong Hwa University) "logo" and "text" gray-scale images to be embedded are shown in Figure 1. Note that due to the different embedding rates for different schemes, the secret images of 59×59, 47×47, and 49×49 pixels for the conventional, EC-based, and UEP-based schemes, respectively, have been used to ensure fair comparisons. The achieved PSNR values are listed in Table VI. The results indicate that the considered schemes produce high-quality stegoimages and the highest PSNR was achieved by the conventional scheme due to the higher *EE* (343% versus 114% in the EC-based scheme and 133% in the UEP-based scheme). This suggests that adding the error correcting capability does not distort the stegoimage seriously and that employing the error correcting codes (EC or UEP) in the coded LSB embedding scheme constitutes a reasonable and practical solution to overcome the error spreading problem.

Figure 1. Secret images with size 59×59 (left), 47×47 (middle), and 49×49 pixels (right).

Table V. *ER* and *EE* for the conventional, EC-based, and UEP-based coded LSB schemes with $(n,k,l)=(7,4,3)$.

Coded LSB scheme	*ER*	*EE*	Embedding of 1000 bits	
			Number of pixels needed	Altered LSBs
Conventional	42.9%	342%	2334	292
EC-based	27.3%	114%	3667	877
UEP-based	30.0%	133%	3334	750

Table VI. PSNR(dB) between the cover image and its stegoimage for the conventional, EC-based, and UEP-based schemes.

Coded LSB scheme Cover image	Conventional	EC-based	UEP-based
Baboon	57.173	54.671	54.691
Barb	57.148	54.687	54.675
Boat	57.181	54.696	54.677
Elaine	57.143	54.675	54.681
Mena	57.151	54.683	54.710
Peppers	57.179	54.693	54.671

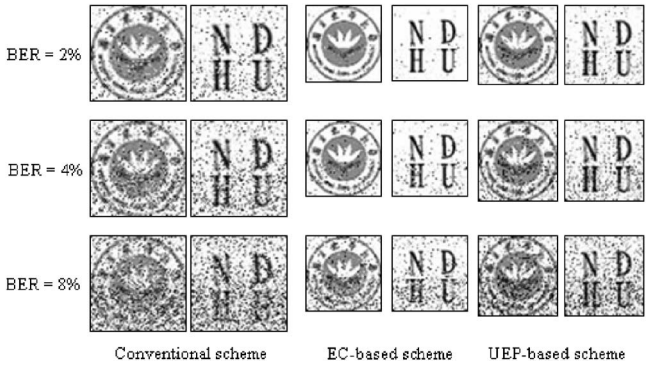

Figure 2. Recovered secret images with BER = 2%, 4%, and 8% for errors placed in random positions.

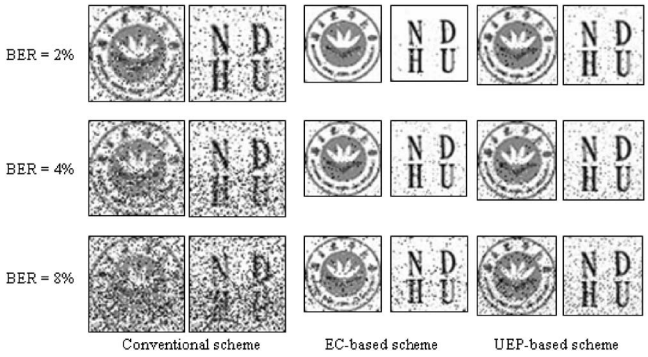

Figure 3. Recovered secret images with BER = 2%, 4%, and 8% for errors placed in the worst positions.

To further study the error spreading problem, the two types of error patterns, namely random errors and worst errors have been added into the LSBs of stegoimages. The first type (random errors) means that the errors are randomly distributed in n-bit vector. However, the second type (worst errors) means that the errors occur in the worst positions (the first k bits of the original n-bit vector) where will cause error spreading. Figures 2 and 3 show the corresponding results obtained by extracting the embedded secret images from the noise corrupted stegoimages. Visual inspection of the results reveals that in a noisy environment the schemes based on UEP and EC codes have comparable performance and clearly outperform the conventional coded LSB scheme. Moreover, since the proposed UEP-based scheme has higher ER than the EC-based scheme, it can be concluded that our solution provides a tradeoff between the data embedding performance and the protection of the embedded secret.

CONCLUSIONS

A refined steganographic solution was introduced. Using UEP codes, we overcame the error spreading problem in the coded LSB steganographic scheme originally proposed by van Dijk and Willems. Our solution has the same correction effect as the Zhang-Wang EC-based scheme while allowing for lower embedding rates. This suggests that the solution proposed in this paper embeds the same secret message with the higher efficiency and produces less distortion in the generated stegoimage. The proposed solution is suitable for applications, such as transmission of the private digital materials (e.g., documents or signature images) through public and wireless networks, where data hiding and protection against communication errors are required or recommended.

ACKNOWLEDGMENT

This work was supported in part by TWISC@NCKU, National Science Council under the Grants NSC 94-3114-P-006-001-Y.

REFERENCES

[1] M. van Dijk and F. Willems, "Embedding information in grayscale images", *Proc. 22nd Symp. Inform. Theory in the Benelux* (Elsevier, Netherlands, 2001), pp. 147–154.
[2] X. Zhang and S. Wang, "Stego-encoding with error correction capability", IEICE Trans. Fundamentals **E88-A**, 3663–3667 (2005).
[3] W. J. van Gils, "Two topics on linear unequal error protection codes: bounds on their length and cyclic code classes", IEEE Trans. Inf. Theory **29**, 866–876 (1983).
[4] M. C. Lin, C. C. Lin, and S. Lin, "Computer search for binary cyclic UEP codes of odd length up to 65", IEEE Trans. Inf. Theory **36**, 924–935 (1990).
[5] S. Dumitrescu, X. Wu, and Z. Wang, "Detection of LSB steganography via sample pair analysis", IEEE Trans. Signal Process. **51**, 1995–2007 (2003).
[6] I. Avcibas, N. Memon, and B. Sankur, "Steganalysis using image quality metrics", IEEE Trans. Image Process. **12**, 221–229 (2003).