LETTER

Stego-Encoding with Error Correction Capability*

Xinpeng ZHANG[†a)] ***and*** **Shuozhong WANG**[†], ***Nonmembers***

SUMMARY Although a proposed steganographic encoding scheme can reduce distortion caused by data hiding, it makes the system susceptible to active-warden attacks due to error spreading. Meanwhile, straightforward application of error correction encoding inevitably increases the required amount of bit alterations so that the risk of being detected will increase. To overcome the drawback in both cases, an integrated approach is introduced that combines the stego-encoding and error correction encoding to provide enhanced robustness against active attacks and channel noise while keeping good imperceptibility.

key words: *steganography, encoding, imperceptibility, error correction*

1. Introduction

As an important class of information hiding techniques, steganography is to send secret messages under the cover of a carrier signal [1]. In the past, various statistical techniques have emerged to detect the presence of hidden data [2]–[5]. Generally speaking, the more the data are embedded, the more vulnerable the system will be to most of the steganalytic attempts. In another type of attack, by overwriting some insignificant contents in the cover, an active warden can effectively prevent extraction of the hidden message [6]–[8]. A secure steganographic system must be capable of resisting both types of attack.

There are two approaches to improving steganographic security against statistical analysis. The first is to preserve, or compensate for the induced change in order to restore, the statistical property of the carrier medium [9]–[11]. The other is to reduce the amount of alterations needed for embedding the required number of bits. For example, a proposed scheme [12] can conceal as many as $\log_2(mn+1)$ bits of data in a binary image block sized $m \times n$ by changing, at most, two bits in the block. Matrix encoding [13] uses less than one change of LSB (least significant bit) to embed l bits into $2^l - 1$ pixels. In this way, the introduced distortion is significantly lowered compared to a plain LSB technique in which secret bits are used to simply replace the LSB plane. Meanwhile, some effective encoding methods derived from the cyclic coding have been described [14], and the matrix encoding can be viewed as a special case of these encoding methods. For the convenience of discussion, the techniques described in [14] are referred to as stego-encoding in the present work. We will show that, by using stego-encoding, each bit of error introduced by channel noise or active attack will cause more errors in message extraction.

To combat channel noise or active attacks, a popular approach is to introduce error-correction codes. Error correction techniques, however, inevitably increase the required amount of bit alterations so that the probability of the secret data being detected will increase. Therefore a steganographic technique capable of both correcting bit-errors and keeping low distortion is desirable.

In this paper, a novel encoding approach is proposed, in which the stego-encoding is presented in a suitable way and combined with an error-correction scheme when a substantial amount of the host samples are available. Using this technique, security of the hidden data with respect to both statistical analysis and active attack is effectively enhanced.

In Sect. 2, an alternative presentation of stego-encoding method is provided, and the problem of error spreading is described. The proposed combination encoding is described with performance analyzed in Sect. 3. Section 4 discusses an application of combination encoding, and Sect. 5 concludes the paper.

2. Stego-Encoding

In steganography, some designated space within the cover signal is generally used to accommodate the additional message to be conveyed in secrecy. For example, the entire LSB plane of a cover image may be replaced with the secret data, and insertion of each bit causes a change of 1/2 least-significant-bit in average. In the stego-encoding, the host data space available for accommodating secret message are divided into many blocks, each of which possesses n bits and is used to carry a message chip containing l secret bits ($l < n$). This means the available space in the host signal is greater than that required by the secret bits. This way, the stego-encoding consumes more cover data, among which only a small fraction is actually modified, for a better cover-bit-alteration efficiency, or equivalently, an enhanced degree of security against steganalysis. In other words, the number of required cover-bit-alteration is reduced.

We call a permutation of n bits in a host data block as a code. There are 2^n different codes and 2^l different types of secret chips. One can use several different codes to represent

Manuscript received May 18, 2005.
Manuscript revised August 10, 2005.
Final manuscript received September 5, 2005.

[†]The authors are with the School of Communication and Information Engineering, Shanghai University, China.

*This work was supported by the Natural Science Foundation of China (No. 60372090 and 60502039) and the Key Project of Shanghai Municipality for Basic Research (No. 04JC14037).

a) E-mail: xzhang@staff.shu.edu.cn

DOI: 10.1093/ietfec/e88–a.12.3663

one type of secret chip. The host bits should be changed to the nearest code corresponding to the secret chip. The more the codes corresponding to each secret chip are scattered, the less is the distortion caused by the steganographic embedding. In an error correction coding, error-free codes are dispersed. So, a general algorithm of stego-encoding can be derived as follows, based on the cyclic coding.

Assume that

$$x^n + 1 = F(x) \cdot G(x) \tag{1}$$

where

$$F(x) = f_k \cdot x^k + f_{k-1} \cdot x^{k-1} + \cdots + f_0$$
$$f_k = 1, \quad f_{k-1}, f_{k-2}, \cdots, f_0 \in \{0,\ 1\} \tag{2}$$
$$G(x) = g_l \cdot x^l + g_{l-1} \cdot x^{l-1} + \cdots + g_0$$
$$g_l = 1, \quad g_{l-1}, g_{l-2}, \cdots, g_0 \in \{0,\ 1\} \tag{3}$$

In these expressions, the plus sign represents a modular arithmetic operation, and $n = k + l$. A binary polynomial $c_{n-1} \cdot x^{n-1} + c_{n-2} \cdot x^{n-2} + \ldots + c_0$ can also be represented in the form of a binary code $(c_{n-1}c_{n-2}\ldots c_0)$. The codes corresponding to the polynomials with a factor $G(x)$ are error-free in the (n, k) cyclic encoding. We denote a set consisting of all error-free codes as L, and the average Hamming distance between each code and its nearest code in L as d_{mean}.

A secret chip $(s_{l-1}s_{l-2}\ldots s_0)$ is equivalent to the polynomial $S(x) = s_{l-1} \cdot x^{l-1} + s_{l-2} \cdot x^{l-2} + \ldots + s_0$. The 2^l different cosets, each containing 2^k codes, can be produced by adding different polynomials $S(x)$ to the coset leader L. These cosets have a number of important properties: (i) Different cosets do not share any identical code; (ii) Any n-bit code is an element belonging to one of the cosets; and (iii) Given any coset, the average Hamming distance between each code and its nearest code in the coset is still d_{mean}.

We now use different cosets to represent different secret chips, and each bit in the available cover data space is adjusted to the nearest code in the corresponding coset for information embedding. Two parameters are used to indicate the performance of a stego-encoding (n, k, l), namely, (i) the embedding rate that is the number of embedded bits per host pixel or sample, associated with the payload:

$$R_S = l/n \tag{4}$$

and (ii) the embedding efficiency, namely, the number of embedded bit per change of the host bit, associated with the imperceptibility:

$$E_S = l/d_{mean} \tag{5}$$

The parameter d_{mean} is the mathematical expectation of the LSB alterations when l stego-bits are embedded into n cover data.

Consider (7, 3, 4) as an example, which serves to illustrate the principle of stego-encoding. Because $x^7+1 = (x^3 + x^2 + 1)(x^4 + x^3 + x^2 + 1)$, the coset leader L includes 0000000, 0011101, 0100111, 0111010, 1001110, 1010011, 1101001, and 1110100. Different cosets are listed in Table 1. The average number of LSB alterations is 1.50 when embedding 4 secret bits into 7 cover data, i.e., R_S=57.1% and E_S=2.67. Clearly, in the simple replacement approach, R_S=100% and E_S=2. In essence, the stego-encoding uses more cover data, while it actually modified fewer bits, than the plain bit-replacement embedding for the same amount of secret data. In other words, the price paid for the reduced distortion is to *preserve*, or *consume*, more data in the cover media.

Table 1 List of codes in different cosets of the stego-encoding (7, 3, 4).

Coset 00	0000000	0011101	0111010	0100111
	1110100	1101001	1001110	1010011
Coset 01	0000001*	0011100	0111011	0100110
	1110101	1101000	1001111	1010010
Coset 02	0000010*	0011111	0111000	0100101
	1110110	1101011	1001100	1010001
Coset 03	0000011	0011110	0111001	0100100
	1110111	1101010	1001101	1010000
Coset 04	0000100*	0011001	0111110	0100011
	1110000	1101101	1001010	1010111
Coset 05	0000101	0011000	0111111	0100010
	1110001	1101100	1001011	1010110
Coset 06	0000110	0011011	0111100	0100001
	1110010	1101111	1001000	1010101
Coset 06	0000110	0011011	0111100	0100001
	1110010	1101111	1001000	1010101
Coset 07	0000111	0011010	0111101	0100000*
	1110011	1101110	1001001	1010100
Coset 08	0001000*	0010101	0110010	0101111
	1111100	1100001	1000110	1011011
Coset 09	0001001	0010100	0110011	0101110
	1111101	1100000	1000111	1011010
Coset 10	0001010	0010111	0110000	0101101
	1111110	1100011	1000100	1011001
Coset 11	0001011	0010110	0110001	0101100
	1111111	1100010	1000101	1011000
Coset 12	0001100	0010001	0110110	0101011
	1111000	1100101	1000010	1011111
Coset 13	0001101	0010000*	0110111	0101010
	1111001	1100100	1000011	1011110
Coset 14	0001110	0010011	0110100	0101001
	1111010	1100111	1000000*	1011101
Coset 15	0001111	0010010	0110101	0101000
	1111011	1100110	1000001	1011100

The drawback of stego-encoding is that each bit of error caused by channel noise or by an active warden attack may arouse multi-bit errors on the extraction side because the codes belonging to different cosets are inter-mixed together. For example, let a stego-code 0000000 carry a secret chip 0000. If one bit in stego-code is changed during transmission, the erroneous stego-codes as marked by asterisks in Table 1 fall into different cosets. Therefore, on the decoding side, it may be interpreted as 0001, 0010, 0100, 0111, 1000, 1101, or 1110. This means that the secret message is almost completely destroyed. This problem will be dealt with by integrating an error-correction coding into the stego-encoding framework as described in the next section.

3. Combination Encoding

As stated in the previous section, although the stego-encoding can improve imperceptibility by reducing the

amount of cover-bit-alteration, it increases vulnerability to active-warden attacks or channel interferences due to error spreading. Since the stego-encoding is derived from cyclic encoding, we can combine the two to keep a low level of embedding-induced distortion on the one hand, and correct bit errors introduced in the channel on the other.

Assume that

$$x^n + 1 = H_1(x) \cdot H_2(x) \tag{6}$$

where $H_1(x)$ can further be factored into $F(x)$ and $G(x)$:

$$x^n + 1 = F(x) \cdot G(x) \cdot H_2(x) \tag{7}$$

Here, the highest powers in $H_1(x)$, $H_2(x)$, $F(x)$, and $G(x)$ are m, $n-m$, k, and l, respectively, where $m = k + l$. A group including all codes corresponding to the polynomials having a factor $H_2(x)$ is named T. So, the order of T is 2^m. We also denote a set consisting 2^k codes or polynomials with a factor $G(x)\cdot H_2(x)$ as L_T and the average Hamming distance between each n-bit code and its nearest code in L_T as D_{mean}.

As a subset of T, L_T, can be treated as a coset leader of T to produce 2^l cosets. Let the polynomial $S(x) = s_{l-1} \cdot x^{l-1} + s_{l-2} \cdot x^{l-2} + \ldots + s_0$ represent a secret chip $(s_{l-1} s_{l-2} \ldots s_0)$. The polynomial $H_2(x)\cdot S(x)$ belongs to T. A coset can be constructed by adding $H_2(x)\cdot S(x)$ to the coset leader L_T. There are 2^l cosets since the total number of different $S(x)$ is 2^l. These cosets have the same properties as given in Sect. 2: (i) No identical code is shared between different cosets; (ii) Any code in T is an element in one of the cosets; and (iii) In any coset, the average Hamming distance between each n-bit code and its nearest code within the coset is D_{mean}.

Different cosets are used to represent different secret chips, and each LSB of the cover data is adjusted to the nearest code in the corresponding coset for information embedding. We name this method (n, m, k, l)-combination encoding, where (n, m, k, l) are the factorization indices. Because all the error-free codes belong to T, the error-correction capability of the (n, m, k, l)-combination encoding is equivalent to that of (n, m)-cyclic encoding. As in the stego-encoding, two parameters are defined: the embedding rate

$$R_C = l/n \tag{8}$$

and the embedding efficiency,

$$E_C = l/D_{mean} \tag{9}$$

When the (n, m)-cyclic encoding is used in a traditional manner, an m-bit secret chip is encoded as a code with length n for error correction capability. The embedding rate and the embedding efficiency are, respectively,

$$R_E = m/n \tag{10}$$

and

$$E_E = m/D_T \tag{11}$$

where D_T is the average Hamming distance between each n-bit code and a given code in T. From (8) and (10), it is clear that

$$R_C < R_E < 1 \tag{12}$$

In a word, simple introduction of the cyclic encoding sacrifices the embedding rate for error-correction capability by increasing cover-bit-alteration, or equivalently, decreasing embedding efficiency, whereas the combination encoding uses more host data to improve embedding efficiency, or to lower the embedding-induced distortion.

For example, a polynomial $x^{15}+1$ can be factored into the product of $x^{11}+x^8+x^7+x^5+x^3+x^2+x+1$ and x^4+x+1. If the (15, 11)-cyclic encoding is used, one-bit of error per 15 bits can be corrected with an embedding rate R_E=0.73 and embedding efficiency E_E=1.47. In other words, to embed 100 secret bits, 137 host bits and an average of 68 alterations are needed. Because $x^{11} + x^8 + x^7 + x^5 + x^3 + x^2 + x + 1 = (x^4 + x^3 + x^2 + x + 1)(x^7 + x^6 + x^4 + 1)$, the (15, 11, 4, 7)-combination encoding can be implemented. In this case, 7 secret bits can be embedded into 15 host bits, in which one-bit of error per 15 bits can still be corrected with a decreased embedding rate R_C=0.47 and an increased embedding efficiency E_C=1.65. For embedding 100 secret bits, 215 host bits and an average of 61 alterations are needed. Because the polynomial $x^{11} + x^8 + x^7 + x^5 + x^3 + x^2 + x + 1$ can also be factored into $(x^2 + x + 1)\,(x^9 + x^8 + x^5 + x^4 + x^3 + 1)$, using the (15, 11, 2, 9)-combination encoding, the embedding rate and the embedding efficiency are R_C=0.60 and E_C=1.72 respectively. In this case, one-bit of error per 15 bits can be corrected, and 167 host bits and 58 alterations in average are needed for embedding 100 secret bits. It is observed that performance of the (15, 11, 2, 9)-combination encoding is better than that of the (15, 11, 4, 7)-combination encoding in terms of embedding efficiency and consumed host data.

To give another example, consider the polynomial $x^{21}+1=(x^6 + x^5 + x^4 + x^2 + 1)(x^6 + x^4 + x + 1)(x^9 + x^8 + x^7 + x^5 + x^4 + x + 1)$. Using the (21, 12)-cyclic encoding with 2-bit-per-21-bit error correction capability, the embedding rate is 0.57 and the embedding efficiency 0.93. If the (21, 12, 6, 6)-combination encoding is used, the embedding rate is reduced to 0.29, while the embedding efficiency increased

Table 2 Comparison between the cyclic encoding and the combination encoding. A polynomial $x^{15} + 1$ is used. In each case, one-bit of error in every 15 bits can be corrected.

Encoding methods	Cyclic	Combination	Combination
Factorization indices	(15, 11)	(15, 11, 4, 7)	(15, 11, 2, 9)
Embedding rate R	0.73	0.47	0.60
Embedding efficiency E	1.47	1.65	1.72
Host bits needed for embedding 100 bits	137	215	167
Alterations for embedding 100 bits	68	61	58

Table 3 Comparison between the cyclic encoding and the combination encoding using the polynomial $x^{21}+1$. In both cases, 2 bits of error in every 21 bits can be corrected.

Encoding methods	Cyclic	Combination
Factorization indices	(21, 12)	(21, 12, 6, 6)
Embedding rate R	0.57	0.29
Embedding efficiency E	0.93	1.15
Host bits needed for embedding 100 bits	175	344
Alterations for embedding 100 bits	108	87

to 1.15.

The above results are summarized in Tables 2 and 3. It is seen that the combination encoding provides higher embedding efficiency with less alterations.

4. Application of Combination Encoding

In fact, the combination encoding technique is independent of the selection of available cover data space as well as the method of cover data modification. In other words, the combination encoding can be used in conjunction with various data embedding approaches to gain both error correction capability and low distortion at a same time. This section presents two examples showing the advantage of using the combination encoding in steganography in different types of image.

First, consider data embedding in a binary image sized 156×300 as shown in Fig. 1. One can label a pixel *changeable* if there are both black and white pixels in its 8 neighbors. Obviously, flipping a *changeable* pixel is more tolerable than flipping a pixel in a uniform area. Pseudo-randomly permute and assign all pixels into 7800 groups, each containing 6 pixels. The way of permutation and assignment is derived from a secret key. In this case, most groups include at least one changeable pixel. Furthermore, pseudo-randomly select 2100 groups and count the number of white pixels in each group, forming a vector of 2100 elements that are the parities of the 2100 counts. The vector is used to carry 600 secret bits with the (21, 12, 6, 6)-combination encoding method. Should the parity of a pixel group be altered for data hiding, one changeable pixel in the group is flipped. Figure 2 presents the image containing a secret message, with only 521 original pixels flipped. If the cyclic encoding with a same capability of error correction were used, 648 flips would have been required. On the extraction side, the hidden information can be recovered reliably using the same permutation and assignment derived from the secret key. Because of the 2-bit-per-21-bit error correction capability of the (21, 12, 6, 6)-combination encoding, 743 pixels in the entire image, caused by channel noise or active attack, can be corrected in data extraction. Such a strong attack is

Fig. 1 Original image.

Fig. 2 Stego-image.

Fig. 3 A stego-image with 600 pixels in error.

in fact meaningless since distortion at this level will destroy the image. Figure 3 shows a stego-image with 600 scattered pixels in error.

As another example, the combination encoding is applied in conjunction with the quantization technique for embedding secret data into a gray scale image. After pseudo-randomly permuting all host pixels, an orthogonal transform is carried out to produce a series of coefficients. Select a part of the coefficients and calculate their quantized parities

$$b_i = \mathrm{mod}[\mathrm{round}(C_i/\Delta), 2] \tag{13}$$

where C_i is the selected coefficient, Δ a system parameter, and the operation round$(\cdot)$ returns the nearest integer of the argument. As in the previous application, the way of pixel permutation and coefficient selection is derived from a secret key. The obtained b_is form a host data space for carrying the secret message, to which the combination encoding is applied. If the value of b_i is to be changed in the data embedding, an amount of Δ is, at random, added to or subtracted from the corresponding C_i. On the extraction side,

after calculating stego-coefficients in the transform domain according to the secret key, the hidden bits can be recovered through quantization and decoding.

In this scheme, a larger Δ leads to a better robustness. The number of selected coefficients may be used to control the level of distortion caused by data hiding. Here the application of combination encoding contributes to both error correction capability and reduced distortion. For example, use the test image Lena sized 512×512, and select 3000 coefficients to carry 1800 secret bits. With the (15, 11, 2, 9) method and $\Delta = 32$, the distortion caused by data hiding is invisible, measured by PSNR = 42 dB. In this case, additive white Guassian noise with PSNR = 35 dB, which is clearly visible, can be effectively resisted.

In the combination encoding, all polynomials corresponding to error-free codes contain an identical factor $H_2(x)$. It might be argued that this regularity could provide a clue to steganalysis if the secret chips were serially encoded and inserted. As a matter of fact, however, this possible loophole has been removed in the above two examples as, by pseudo-random permutation and selection of the host data, the regularity existing in the stego-data-space is deeply concealed, therefore can no longer be used in steganalysis because the steganalyst does not have the secret key.

5. Conclusion

In summary, by preserving, or consuming, more host data, the stego-encoding as introduced in the literature reduces the number of cover-bit-alteration, hence providing improved security in the sense of evading steganalysis. The drawback is that slight noise can destroy the embedded message. On the other hand, when error correction encoding is introduced, distortion caused by data hiding is inevitably increased. The technique proposed in this paper integrates error-correction codes into the stego-encoding framework. It provides error-correction capability so that the embedded data have better chance to survive interferences of channel noise or active attacks while keeping low distortion caused by information hiding. The price paid is, of course, an additional increase of the host data needed for embedding the same quantity of secrete data. Table 4 summarizes the performance of the different encoding techniques, in which the number of '+/−' symbols represents the merits/drawbacks in each specification. The proposed combination encoding provides the best overall security both in terms of imperceptibility and robustness against channel noise or active attack.

Table 4 Performances comparison between steganographic techniques.

Steganographic techniques	Embedding rate (Payload)	Embedding efficiency (Imperceptibility)	Robustness
Plain embedding	+++++	+++	+
Stego-encoding	++++	++++	−
Cyclic encoding	++++	+	+++
Combination encoding	+++	+++	+++

References

[1] F.A.P. Petitcolas, R.J. Anderson, and M.G. Kuhn, "Information hiding—A survey," Proc. IEEE, vol.87, pp.1062–1078, 1999.

[2] H. Wang and S. Wang, "Cyber warfare—Steganography vs. steganalysis," Commun. ACM, vol.47, no.10, pp.76–82, 2004.

[3] J. Fridrich and M. Goljan, "Practical steganalysis of digital images—State of the art, security and watermarking of multimedia contents IV," Proc. SPIE, 4675, pp.1–13, 2002.

[4] A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems," 3rd International Workshop on Information Hiding, Lecture Notes in Computer Science, 1768, pp.61–76, 1999.

[5] S. Lyu and H. Farid, "Detecting hidden message using higher-order statistics and support vector machines," 5th International Workshop on Information Hiding, Lecture Notes in Computer Science, 2578, pp.340–354, 2002.

[6] G. Fisk, M. Fisk, C. Papadopoulos, and J. Neil, "Eliminating steganography in Internet traffic with active wardens," 5th International Workshop on Information Hiding, Lecture Notes in Computer Science, 2578, pp.18–35, 2002.

[7] J. Ettinger, "Steganalysis and game equilibria," 2nd International Workshop on Information Hiding, Lecture Notes in Computer Science, 1525, pp.319–328, Springer-Verlag, 1998.

[8] K. Zhang, S. Wang, and X. Zhang, "Detection and removal of hidden data in images embedded with quantization index modulation," Lecture Notes in Computer Science, 2776, pp.360–370, 2003.

[9] S. Wang, X. Zhang, and K. Zhang, "Steganographic technique capable of withstanding RQP analysis," J. Shanghai University, vol.6, pp.273–277, 2002.

[10] X. Zhang, S. Wang, and K. Zhang, "Steganography with least histogram abnormality," Lecture Notes in Computer Science, 2776, pp.395–406, 2003.

[11] X. Zhang and S. Wang, "Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security," Pattern Recognit. Lett., vol.25, pp.331–339, 2004.

[12] Y.-C. Tseng, Y.-Y. Chen, and H.-K. Pan, "A secure data hiding scheme for binary images," IEEE Trans. Commun., vol.50, no.8, pp.1227–1231, 2002.

[13] A. Westfeld, "F5—A steganographic algorithm," 4th International Workshop on Information Hiding, Lecture Notes in Computer Science, 2137, pp.289–302, 2001.

[14] M. Dijk and F. Willems, "Embedding information in grayscale images," Proc. 22nd Symp. Inform. Theory, pp.147–154, Benelux, The Netherlands, 2001.