Improving the Embedding Efficiency of Wet Paper Codes by Paper Folding

Weiming Zhang and Xuexiu Zhu

***Abstract*—Wet paper codes are used to design steganographic schemes, in which the sender can embed messages into a cover with arbitrarily selected changeable bits that are not shared by the recipient. In this letter, we propose a novel approach to wet paper codes by folding the cover into several layers and applying basic wet paper coding methods with low computational complexity to each layer. This method uses the changes introduced in the first layer to embed messages into every layer and therefore achieves high embedding efficiency (average number of bits embedded by per change).**

***Index Terms*—Embedding efficiency, LT process, relative payload, steganogaphy, wet paper codes.**

I. Introduction

STEGANOGRAPHY is used to communicate secret messages under the cover of innocence objects, e.g., digital images. To resist detection, the sender should consider two basic problems when embedding messages into the cover.

1) How to avoid changing inconspicuous parts of the cover, e.g., the flat and less textured areas of a cover image.
2) How to decrease the number of embedding changes on the cover, or in other words, how to improve the embedding efficiency (the average number of bits embedded by per change).

The first problem in fact requires that the sender determines changeable pixels of the cover image to carry messages and forbids any changes on the remaining pixels. The changeable pixels are called selection channel which is not shared with the recipient. This data embedding problem in steganography is referred as to wet paper codes (WPCs) by Fridrich *et al.* [1], [2], who also proposed an efficient encoding method based on Luby transform (LT) process [3]. Wet paper codes have been used to various branches of information hiding, such as [4]–[7].

The second problem is related to covering codes [8], which is usually called matrix embedding in steganography. Many embedding methods based on covering codes have been reported to increase the embedding efficiency [9]–[12].

However, it is still hard to embed messages using WPCs with high embedding efficiency. BCH codes [10] and Reed–Solomon (RS) codes [13] were proposed to improve the embedding efficiency of WPCs. These structured codes however are inefficient for the steganographic applications, because the number of the changeable pixels varies significantly for different cover objects. That's why Fridrich *et al.* [1] proposed variable-rate random linear codes when they brought up wet paper codes. To increase the embedding efficiency of random linear codes with feasible computational complexity, only codes with small codimension can be used, which idea is utilized by the Meet-in-the-Middle method [14].

In this letter, we propose a novel wet paper coding method, which folds the cover into two layers with XOR operations and embeds messages into each layer by using WPC based on LT process [3] or Gaussian elimination [2]. Because one change on the cover can be used to embed messages into two layers at locations such that bits in both layers are changeable, the embedding efficiency is significantly increased. By recursively using this method, we get efficient wet paper codes for various relative payloads.

The rest of this letter is organized as follows. WPCs are briefly introduced in Section II. In Section III, we present the paper folding method for WPCs. The performance of the paper folding method is compared with that of the Meet-in-the-Middle method in Section IV. The letter is concluded in Section V.

II. Wet Paper Codes

We denote matrices and vectors by boldface fonts, and take gray-scale images as example to describe the proposed method. Assume the image consists of n pixels, and $\mathbf{x} = (x_1, \ldots, x_n)$ are the LSBs of pixels which are used as carriers for binary embedding. First, the sender determines a selection channel with k changeable bits x_j, $j \in J \subset \{1, 2, \ldots, n\}$, $|J| = k$, which is not shared with the recipient. The sender will embed a secret message $\mathbf{m} = (m_1, \ldots, m_m)$ into $\mathbf{x}$ by only modifying some bits in the selection channel J.

In the terms of wet paper codes, we say the bits in the selection channel are dry and other bits are wet; define the wet rate as $\gamma = (n-k)/n$, and dry rate as $1-\gamma$. If a wet paper code can embed m bits of messages into an n-length cover with wet rate γ by R_a changes on average, we say the code has relative payload $\alpha = m/(1-\gamma)n = m/k$, change rate $c = R_a/(1-\gamma)n = R_a/k$, and the code is called WPC (α, c). Define embedding efficiency as $e = m/R_a = \alpha/c$. It has been proved [3] that the embedding efficiency is bounded from above by

$$e(\alpha) \leq \frac{\alpha}{H_2^{-1}(\alpha)}, \quad 0 \leq \alpha \leq 1 \tag{1}$$

where $H_2^{-1}(y)$ is the inverse function of the binary-entropy function $H_2(x) = -x \log_2 x - (1-x) \log_2 (1-x)$.

Manuscript received April 11, 2009; revised May 22, 2009. First published June 10, 2009; current version published July 09, 2009. This work was supported by the Natural Science Foundation of China (60803155) and the High-Tech Research and Development Program of China (2007AA01Z477). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. James E. Fowler.

The authors are with Department of Information Research, Zhengzhou Information Science and Technology Institute, Zhengzhou 450002, China (e-mail: xiuzi0305@163.com; zwmshu@gmail.com).

Digital Object Identifier 10.1109/LSP.2009.2024807

Wet paper codes can be constructed by random linear codes [2], in which, to embed $\mathbf{m}$ into $\mathbf{x}$, the sender changes some bits in the selection channel and modifies the cover-object $\mathbf{x}$ to the stego-object $\mathbf{y} = (y_1, \ldots, y_n)$, which satisfies

$$\mathbf{D}\mathbf{y}^T = \mathbf{m}^T \tag{2}$$

where $\mathbf{D}$ is an $m \times n$ binary pseudo-random matrix. $\mathbf{D}$ usually is generated with a stego-key shared by the sender and the recipient. Therefore the recipient can extract $\mathbf{m}$ from $\mathbf{y}$ by only calculating $\mathbf{D}\mathbf{y}^T$ without any knowledge about the selection channel J.

It is pointed out [2] that we can communicate about k bits of messages by above manner when the number of the changeable bits $k(k < n)$ is large enough. The encoding procedure can be executed by solving (2) with Gaussian elimination [2]. Fridrich *et al.* [3] proposed an fast method to solve (2) based on LT process, which can communicate a long message, such as $k = 10^5$, by one time. The WPC by using Gaussian elimination or LT process can embed messages with relative payload about 1 and change rate 1/2, which we call basic WPC $(1, 1/2)$.

To minimize the change rate or maximize the embedding efficiency, we should find an optimal solution to (2), i.e., a solution with the least modifications. When the message length is small, we can search for such an optimal solution with feasible complexity. For example, the Meet-in-the-Middle method [14] decreases the change rate by embedding a short message into each small block of the cover. Next, we will present a more efficient method by folding the wet paper cover.

III. Paper Folding Method

A. 1-Folding Construction

Now assume that there is a 2^N -length wet paper cover with wet rate γ.

$$x_1, \ldots, x_{2^N}. \tag{3}$$

We will embed messages into (3) with a WPC (α, c) and the basic WPC $(1, 1/2)$ by two steps.

First, fold the wet paper (3) into two layers as follows

$$\begin{matrix} x_1, & x_2, & \ldots, & x_{2^{N-1}} \\ x_{2^{N-1}+1}, & x_{2^{N-1}+2}, & \ldots, & x_{2^N}. \end{matrix} \tag{4}$$

Compress the double-layered paper (4) into single layer with exclusive-or operation such that

$$y_i = x_i \oplus x_{2^{N-1}+i}, \quad 1 \le i \le 2^{N-1}. \tag{5}$$

Obviously, as long as one bit in $(x_i, x_{2^{N-1}+i})$ is changeable, the y_i can be changed, i.e., y_i is dry. Therefore we define that y_i is wet if and only if both of x_i and $x_{2^{N-1}+i}$ are wet. Because the probability of an x_i been wet is γ, the wet rate of cover (5) is γ^2 and its dry rate is $1-\gamma^2$. Applying the WPC (α, c) to cover (5), we can embed $2^{N-1}(1-\gamma^2)\alpha$ bits of messages with on average $2^{N-1}(1-\gamma^2)c$ changes.

Second, take the first layer in (4) and construct a wet paper cover with new definition on the dry/wet bits.

$$x_1, \quad x_2, \quad \ldots, \quad x_{2^{N-1}}. \tag{6}$$

In (6), x_i is defined as a dry bit if and only if both of x_i and $x_{2^{N-1}+i}$ are dry and y_i needs to be changed in the first step. In fact, for the embedding process of the first step, if y_i needs to be changed and both of x_i and $x_{2^{N-1}+i}$ are dry, we can flip any one in $(x_i, x_{2^{N-1}+i})$ to change y_i. By the choice between x_i and $x_{2^{N-1}+i}$, we get a change-free bit x_i, i.e., if x_i equals the needed value, we flip $x_{2^{N-1}+i}$; otherwise we flip x_i. That implies we can embed one more bit into x_i by the change needed by y_i. The probability is $(1-\gamma)^2$ for both of x_i and $x_{2^{N-1}+i}$ being dry, and the corresponding y_i needs to be changed with probability c in the first step, so the dry rate in (6) is $(1-\gamma)^2 c$. Using basic WPC $(1, 1/2)$, we can embed on average $2^{N-1}(1-\gamma)^2 c$ bits of messages into (6).

For above two steps, the modifications are made in following manner. In the first step, when a y_i needs to be changed, if only one bit in $(x_i, x_{2^{N-1}+i})$ are dry, we just flip this dry bit to change y_i; if both of x_i and $x_{2^{N-1}+i}$ are dry, we label this position and the change will be done in the second step. All these labelled positions in the first step just correspond to the dry positions in cover (6). In the second embedding step, for any dry position i in cover (6), if x_i need to be changed, we flip x_i; otherwise we flip $x_{2^{N-1}+i}$, which simultaneously finishes the change required by the first step. Therefore the number of changed bits is determined in the first step. The second step embeds extra messages with some changes introduced by the first step.

Combining the two steps, we in total embed $2^{N-1}(1-\gamma^2)\alpha + 2^{N-1}(1-\gamma)^2 c$ bits into the cover (3) with on average $2^{N-1}(1-\gamma^2)c$ changes. Because the number of dry bits in (3) is $2^N(1-\gamma)$, we have relative payload

$$\frac{2^{N-1}(1-\gamma^2)\alpha + 2^{N-1}(1-\gamma)^2 c}{2^N(1-\gamma)} = \frac{(1+\gamma)\alpha + (1-\gamma)c}{2} \tag{7}$$

and change rate

$$\frac{2^{N-1}(1-\gamma^2)c}{2^N(1-\gamma)} = \frac{(1+\gamma)c}{2}. \tag{8}$$

We call above method 1-folding construction, on which we have the following theorem.

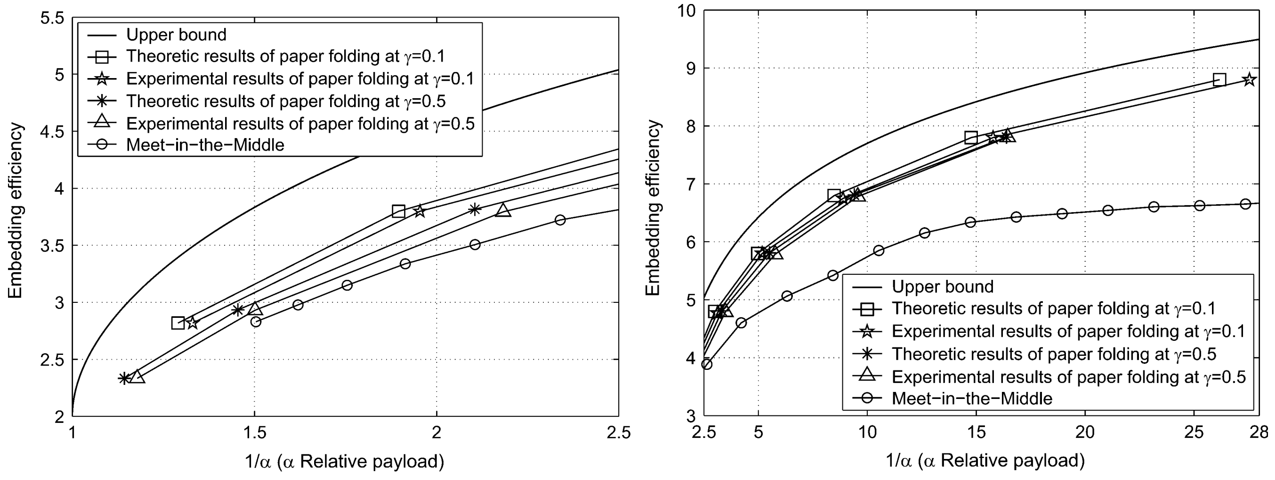
Theorem 1: For a cover (3) with wet rate γ, when using a WPC (α, c) to the folded cover (5), the 1-folding construction has change rate $C_1(c, \gamma)$ and relative payload $A_1(\alpha, c, \gamma)$ such that

$$C_1(c, \gamma) = \frac{(1+\gamma)c}{2} \tag{9}$$

$$A_1(\alpha, c, \gamma) = \frac{(1+\gamma)\alpha + (1-\gamma)c}{2}. \tag{10}$$

B. n-Folding Construction

We can apply the paper folding method recursively, i.e., use the 1-folding construction to the folded cover (5) and get a 2-folding construction on the original cover (3). Note that the wet rate of cover (5) is γ^2, so by Theorem 1, the 1-folding construction can achieve change rate $C_1(c, \gamma^2)$ and relative payload $A_1(\alpha, c, \gamma^2)$ on cover (5). That means we replace the WPC (α, c) by WPC $\big(A_1(\alpha, c, \gamma^2), C_1(c, \gamma^2)\big)$ in Theorem 1, and get



(a)

(b)

Fig. 1. Comparison on the embedding efficiency of the paper folding method and the Meet-in-the-Middle method for various relative payloads. (a) is for 1, 2-folding at wet rate $\gamma = 0.1$ and 1, 2, 3-folding at $\gamma = 0.5$; (b) is for 3, 4, 5, 6, 7-folding at $\gamma = 0.1$ and 4, 5, 6, 7-folding at $\gamma = 0.5$.

change rate $C_2(c,\gamma)$ and relative payload $A_2(\alpha,c,\gamma)$ on cover (3) such that

$$C_2(c,\gamma) = C_1\left(C_1(c,\gamma^2),\gamma\right) \tag{11}$$
$$A_2(\alpha,c,\gamma) = A_1\left(A_1(\alpha,c,\gamma^2),C_1(c,\gamma^2),\gamma\right). \tag{12}$$

Moreover, using the 2-folding construction to the folded cover (5), we get the 3-folding construction on cover (3), which has change rate $C_3(c,\gamma)$ and relative payload $A_3(\alpha,c,\gamma)$ such that

$$C_3(c,\gamma) = C_1\left(C_2(c,\gamma^2),\gamma\right) \tag{13}$$
$$A_3(\alpha,c,\gamma) = A_1\left(A_2(\alpha,c,\gamma^2),C_2(c,\gamma^2),\gamma\right). \tag{14}$$

Generally, the n-folding ($n \geq 2$) construction has change rate $C_n(c,\gamma)$ and relative payload $A_n(\alpha,c,\gamma)$ such that

$$C_n(c,\gamma) = C_1\left(C_{n-1}(c,\gamma^2),\gamma\right) \tag{15}$$
$$A_n(\alpha,c,\gamma) = A_1\left(A_{n-1}(\alpha,c,\gamma^2),C_{n-1}(c,\gamma^2),\gamma\right). \tag{16}$$

For n-folding construction, the embedding process consists of $n+1$ steps. The first step uses WPC (α, c), and all other steps use basic WPC(1, 1/2). Of course, we can also use basic WPC (1, 1/2) in the first step.

IV. Implementation Issues and Performance Comparisons

The paper folding method is compared with the Meet-in-the-Middle method [14]. To embed messages with feasible computational complexity, we use random codes with maximal codimension $p = 20$ for the Meet-in-the-Middle method, which means at most 20 bits of messages can be embedded at once. Therefore we have to divide the cover into small blocks and embed a short message into each block. When the message length m is small, to make (2) have a solution, m usually has to be shorter than the number of changeable bits k, which leads to a capacity loss. In addition, the encoder needs to communicate the length of the message embedded into each block, which will expedt part of the payload. Thus, there is a limit on the the maximal relative payload for the Meet-in-the-Middle Method, which is about 0.7.

For paper folding method, we hope to apply WPC $(1, 1/2)$ to all $n+1$ steps in above n-folding construction, which can be realized by Gaussian elimination. However, because of the high computational complexity of Gaussian elimination, we also have to divide the cover and the message into disjoint blocks, for example, embedding less than 1000 bits into each block. Although, the payload loss of communicating the message length for Gaussian elimination is very small, the implementation will be complex when the message is long. The alternate method is LT process [3] which can embed a long message by one time with low computational complexity. Wet paper coding based on LT process also imposes a payload loss, but this loss decreases as the message length increases. To trade payload for the simplicity of implementation, we propose the following embedding manner in practice. For the $n+1$ steps in the n-folding construction, if the number of changeable bits is larger than 10 000 in one step, we use the LT process; otherwise we use Gaussian elimination by dividing the message and the cover into several blocks.

We embedded messages into an image with 2^{20} pixels by the paper folding and Meet-in-the-Middle method for wet rate $\gamma = 0.1$ and 0.5, respectively. The performance are shown in Fig. 1, where the abscissa denotes α^{-1} (α is relative payload), and the vertical axis denotes embedding efficiency $e = \alpha/c$. Fig. 1(a) is for the range of relative payload $[1/2.5, 1]$, and Fig. 1(b) is for $[1/28, 1/2.5]$. The experimental results by above proposed embedding manner and the theoretic results derived in Section III are both depicted in Fig. 1 for n-folding construction $(n = 1, 2, \ldots, 7)$.

Fig. 1 shows that the embedding efficiency of the paper folding method decreases as the wet rate increases, while the Meet-in-the-Middle method is irrelative with the wet rate. However, note that large wet rate γ means few positions which can carry messages, so in this case we usually need large relative payloads for steganographic applications; and the paper folding method can generate more WPCs with larger relative payloads

TABLE I
COMPARISON ON THE EMBEDDING EFFICIENCY OF THE MEET-IN-THE-MIDDLE METHOD AND THE PAPER FOLDING METHOD FOR SOME RELATIVE PAYLOADS. THE VALUES WERE OBTAINED FOR A COVER IMAGE WITH 2^{20} PIXELS AT WET RATE $\gamma = 0.25$

Relative payload	0.813	0.578	0.373	0.228	0.135	0.078
Meet-in-the-Middle	–	3.46	4.23	5.01	5.59	6.59
Paper Folding	2.60	3.48	4.47	5.47	6.47	7.47

TABLE II
COMPARISON ON THE EMBEDDING TIME (IN SECONDS) OF THE MEET-IN-THE-MIDDLE METHOD AND THE PAPER FOLDING METHOD FOR SOME RELATIVE PAYLOADS. THE VALUES WERE OBTAINED FOR A COVER IMAGE WITH 2^{20} PIXELS AT WET RATE $\gamma = 0.25$

Relative payload	0.813	0.578	0.373	0.228	0.135	0.078
Meet-in-the-Middle	–	116	76	46	26	17
Paper Folding	80	61	40	30	15	6

for larger wet rates. As shown in Fig. 1(a), for wet rate $\gamma = 0.1$, only the 1-folding or 2-folding construction reaches relative payload larger than 0.4, and the largest relative payload is 0.75, while for $\gamma = 0.5$, the relative payload of 1, 2 or 3-folding construction is larger than 0.4, and the largest relative payload is 0.85. Fig. 1 also shows that, for n-folding construction, the relative payload decreases as n increases, and the embedding efficiency of the paper folding method is higher than that of the Meet-in-the-Middle method, especially for small relative payload.

We also compare the embedding speed of the paper folding method and the Meet-in-the-Middle method. We embed messages into a cover image with 2^{20} pixels for wet rate $\gamma = 0.25$ by using n-folding constructions $(n = 1, 2, 3, 4, 5, 6)$ and by using the Meet-in-the-Middle method based on random codes with codimension $p = 20$ at corresponding relative payloads, respectively. The experiments are implemented on a PC equipped with a 2.5 GHz Pentium Dual-Core CPU. The comparisons on embedding efficiency at various relative payloads are list in Table I, and the corresponding embedding speeds are compared in Table II. In both of tables, "–" denotes that the Meet-in-the-Middle method can not reach relative payload 0.813. For other relative payloads, it is shown that the paper folding method can achieve higher embedding efficiency with faster embedding speed.

In fact, the paper folding method decomposes the problem of searching for an optimal solution to (2) into several sub-problems of finding a solution that can be fast implemented, and hence, the computational complexity is decreased.

V. CONCLUSION

In this letter, we propose a fast algorithm to increase the embedding efficiency of wet paper codes. Although the proposed method achieves high embedding efficiency, it can reach only sparse values of relative payloads, as shown in Fig. 1. A simple solution to this problem is to combine existing WPCs. For example, assume there are two WPCs, W_1 and W_2, with relative payload 0.8 and 0.4, respectively, but we need relative payload 0.5. By using W_1 for 1/4 of the cover and W_2 for the remaining 3/4 of the cover, relative payload 0.5 can be attained. In fact, a more efficient method to reach dense relative payloads may be realized by binding n pages of wet paper covers where n can be any integer, which will be discussed in our another coming paper.

REFERENCES

[1] J. Fridrich, M. Goljan, and D. Soukal, "Perturbed quantization steganography using wet paper codes," in *Proc. ACM Multimedia and Security Workshop on Multimedia and Security*, 2004, pp. 4–15.

[2] J. Fridrich, M. Goljan, P. Lisonek, and D. Soukal, "Writing on wet paper," *IEEE Trans. Signal Process.*, vol. 53, pp. 3923–3935, 2005.

[3] J. Fridrich, M. Goljan, and D. Soukal, "Efficient wet paper codes," in *Proc. 7th Int. Workshop on Information Hiding, LNCS 3727*, 2005, pp. 204–218.

[4] M. Wu, J. Fridrich, M. Goljan, and H. Gou, "Handling uneven embedding capacity in binary images: A revisit," *Proc. SPIE, Electron. Imag., Security, Steganography, Watermarking of Multimedia Contents VII*, pp. 194–205, 2005.

[5] H. Gou and M. Wu, "Improving embedding payload in binary images with super-pixels," in *Proc. IEEE Int. Conf. Image Processing, ICIP 2007*, 2007, pp. 277–280.

[6] X. Quan and H. Zhang, "Data hiding in MPEG compressed audio using wet paper codes," in *Proc. 18th Int. Conf. Pattern Recognition, ICPR 2006*, 2006, pp. 727–730.

[7] J. Yu, X. Wang, J. Li, and X. Nan, "A fragile document watermarking technique based on wet paper code," in *Proc. Int. Conf. Intelligent Information Hiding and Multimedia Signal Processing, IIHMSP '2008*, 2008, pp. 25–28.

[8] F. Galand and G. Kabatiansky, "Information hiding by coverings," in *Proc. IEEE Information Theory Workshop 2004*, 2004, pp. 151–154.

[9] J. Bierbrauer and J. Fridrich, "Constructing good covering codes for applications in steganography," *LNCS Trans. Data Hiding and Multimedia Security*, vol. 4902, pp. 1–22, 2008.

[10] D. Schönfeld and A. Winkler, "Embedding with syndrome coding based on BCH codes," in *Proc. ACM 8th Workshop on Multimedia and Security*, 2006, pp. 214–223.

[11] C. Munuera, "Steganography and error-correcting codes," *Signal Process.*, vol. 87, pp. 1528–1533, 2007.

[12] W. Zhang, S. Wang, and X. Zhang, "Improving embedding efficiency of covering codes for applications in steganography," *IEEE Commun. Lett.*, vol. 11, pp. 680–682, Aug. 2007.

[13] C. Fontaine and F. Galand, "How can Reed-Solomon codes improve steganographic schemes?," *Proc. 9th Information Hiding, LNCS 4567*, pp. 130–144, 2007.

[14] J. Fridrich, M. Goljan, and D. Soukal, "Wet paper codes with improved embedding efficiency," *IEEE Trans. Inform. Forensics Secur.*, vol. 1, no. 1, pp. 102–110, 2006.