A CONTENT-ADAPTIVE APPROACH FOR REDUCING EMBEDDING IMPACT IN STEGANOGRAPHY

Chao Wang, Xiaolong Li, Bin Yang, Xiaoqing Lu and Chengcheng Liu

Institute of Computer Science and Technology, Peking University, Beijing 100871, China

ABSTRACT

In this paper, a content-adaptive steganographic scheme is proposed. The novel scheme can be viewed as an improvement of the conventional LSB matching. In this scheme, we take advantage of embedding redundancy in LSB matching to select modification direction (i.e., increasing or decreasing the pixel value by 1), and the dependency of neighboring pixels is taken into consideration. More specifically, if the secret message bit does not match the LSB of the corresponding cover pixel value, the choice of modification direction is not random but a specific selection, in order to hold the correlation of neighboring pixels as far as possible. The resulting stego image looks more like a natural one, and smooth to some extent. Comparing with LSB matching and other state-of-the-art steganography, higher level security of the proposed scheme is experimentally verified. In addition, the proposed approach can be also applied in LSB-based steganography to enhance the security.

Index Terms— Information hiding, steganography, security

1. INTRODUCTION

Steganography is the art and science of covert communication, whose aim is to hide not only a secret message but also the presence of communication. For this purpose, steganographic algorithm embeds secret message into innocuous looking cover data (e.g., digital images) by slightly modifying the cover in such a way that the intended recipient can precisely extract the embedded message. Meanwhile, steganographic algorithm must be secure: the resulting stego data should be visually and statistically indistinguishable from its natural cover. As the counter-technology of steganography, steganalysis aims to expose the presence of covert communication. Typically, existing steganalyzers can be classified into two categories: targeted and blind. Targeted steganalyzers aim to identify the presence of secret message embedded by a specific algorithm, while blind steganalyzers are intended to detect a wide range of steganographic algorithms including previously unknown ones. However, the attackers usually do not know the algorithm used for hiding information. Therefore, blind steganalyzers are practically more valuable and desirable than targeted steganalyzers. Furthermore, blind steganalyzers are usually designed to measure embedding noise [1, 2]. The observed data that appears noisier is more likely to be assigned as a stego one. In this work, we consider digital images as cover data, and our focus is to design more secure steganography.

In [3], Cachin proposed to quantify the steganographic security by the relative entropy: $D(P_c \parallel P_s) = \sum P_c \log \frac{P_c}{P_s}$, where $P_c$ and $P_s$ are respectively the statistical distributions of the cover images and the stego images. In general, the relative entropy $D(P_c \parallel P_s)$ is nonnegative and equal to zero if and only if $P_c = P_s$. Thereafter, according to Cachin’s work, some steganographic algorithms are proposed to preserve a given set of statistics, for example, Sallee’s model-based steganography [4]. Nevertheless, other statistics can still be exploited for steganalysis of this type of steganography. For instance, Sallee’s Cauchy model-based JPEG steganography can be detected by using only the first order statistics [5].

Besides the statistics preserving steganography, there are two common ways to enhance steganographic security. One way is to reduce the embedding changes at a given embedding rate, i.e., to increase the embedding efficiency [6]. For example, the well-known least significant bit (LSB) matching [7] was improved by increasing its embedding efficiency, and better resistance to steganalysis of the improved scheme was illustrated [8, 9]. This type of steganography usually introduces a pixel-to-pixel uncorrelated noise to cover image, and it does not take into account the dependency of neighboring pixels. The fact leaves attackers a clue by considering the correlation between adjacent pixels [10]. Another way is to modify the cover data only in inconspicuous parts, e.g., the noisy regions of an image. This goal can be achieved, for example, using the wet paper codes (WPC) [11]. The WPC allows the sender to freely select embedding pixels (namely, “dry” pixels) while the rest pixels (namely, “wet” pixels) remain unchanged, and most importantly, the partition of dry and wet pixels might be unknown to the receiver. The WPC is an important technique in steganography [12], but the expensive computational cost limits its usage in practice.

In this paper, different from the aforementioned approaches, a content-adaptive steganography, which can be viewed as an improvement of LSB matching, is proposed. The novel scheme exploits the embedding redundancy in LSB matching to select modification direction (i.e., increasing or decreasing the pixel value by 1) and takes the dependency of neighboring pixels into consideration. More specifically, in order to hold as far as possible the correlation of neighboring pixels, the choice of modification direction is non random but a specific selection. The stego image thus obtained is smooth to some extent, which is proved experimentally hard to be perceived by steganalysis. The rest of this paper is organized as follows. We first present in detail the proposed scheme in Section 2. Then in Section 3, extensive experimental results are given. The final conclusion is drawn in last section.

2. THE PROPOSED SCHEME

LSB replacement and LSB matching are two widely-used steganographic algorithms, which have advantages of high payload, good visual imperceptibility and extreme ease of implementation. We know that, in LSB matching embedding, if the secret message bit does not match the LSB of the corresponding cover pixel value (for convenience, we call such a pixel “changeable pixel”), 1 is either added to or subtracted from the pixel value randomly. The random modification eliminates the embedding asymmetry in LSB replace-
ment. Thus LSB matching is considered and experimentally proved more secure than LSB replacement. However, it can still be detected by the recent proposed steganalyzers, for example, the calibration-based steganalyzers [7] or the blind steganalyzer based on wavelet absolute moment (WAM) [2]. In fact, the security of LSB matching can be enhanced by exploiting its embedding redundancy: we may select the modification direction specifically rather than randomly. Such methods will be given in this section.

Let us introduce some notations. Let \( I_c : D \rightarrow \{0, 1, \ldots, 255\} \) be a \( h \times w \) sized gray-scale cover image defined on \( D \), where \( D = \{(i, j) : 1 \leq i \leq h, 1 \leq j \leq w\} \subset \mathbb{N}^2 \) is the set of pixel locations; \( M \subseteq D \) be the set of all changeable pixels (note that this set is determined by the secret message to be embedded); \( S : D \rightarrow \{0, 1, -1\} \) be a possible modification to the cover image due to data embedding:

\[
S(x) = \begin{cases} 
0 & \text{if } x \not\in M, \\
1 & \text{if } x \in M; 
\end{cases}
\]

and \( S \) be the set of all possible modifications. We remark here that in LSB matching embedding, a modification \( S \) is first randomly chosen from \( S \), and then \((I_c + S)\) is taken as the stego image of \( I_c \). Different from LSB matching embedding, our idea is to make the stego image as smooth as possible (in other words, less noisy) by choosing the right modification \( S \in S \). We now start our presentation by introducing a simple data embedding method, and a more valuable approach will be given later.

First, we point out that in LSB matching embedding, the noise introduced by increasing or decreasing may be different for certain cover pixels. For instance, assume that \( x \) is a changeable pixel and most of its neighboring pixels have larger values than \( x \). Under this assumption, decreasing the value of \( x \) would make it more isolated and bring more noise than increasing its value. Thus, the stego image could be less noisy if we select the modification direction wisely rather than randomly. Based on this observation, a straightforward idea is that we may modify the changeable pixels so as to make them closer to the average value of their neighbors. In general, we may select modification direction by referencing the denoised version of the cover image. Now, we introduce a novel data embedding method based on the above discussion. Denote \( I_c^{\text{smooth}} \) as the denoised image of \( I_c \) by a certain denoising algorithm. For each changeable pixel \( x \in M \), we determine the stego pixel value \( I_c(x) \) as

\[
I_c(x) = \begin{cases} 
I_c(x) + 1 & \text{if } I_c(x) < I_c^{\text{smooth}}(x), \\
I_c(x) - 1 & \text{if } I_c(x) > I_c^{\text{smooth}}(x), \\
I_c(x) + 1 \text{ (randomly) } & \text{if } I_c(x) = I_c^{\text{smooth}}(x).
\end{cases}
\]

We call briefly this data embedding algorithm “CAS-D” (content-adaptive steganography via denoising). The data extraction procedure of CAS-D is just the same as LSB matching, and we only need to read successively the LSB of each embedding pixel and to link up the LSBs to rebuild the secret message.

Fig.1 presents the comparison of the receiver operating characteristic (ROC) curves between CAS-D and LSB matching for resisting the WAM steganalyzer [2]. Here, the denoised image is obtained by using an average filter of size \( 3 \times 3 \), a Gaussian low-pass filter of the same size and with a standard deviation \( \sigma = 0.5 \), and a Wiener filter of size \( 5 \times 5 \), respectively. From this figure, we see that CAS-D is harder to detect and thus more secure than LSB matching. This primary experimental result verifies our aforementioned discussion. In addition, the CAS-D with Wiener filter shows nearly the best resistance to steganalysis.

As we move on to introduce a more valuable approach. Note that during data embedding procedure of CAS-D, several changeable pixels might be very close to each other in location. In this situation, the modification of these pixels may affect the average pixel value of this area severely. But the CAS-D does not take this situation into account. Its choice of modification direction only depends on the original cover. Take the following special case for example: the value of a changeable pixel \( x \in M \), \( I_c(x) \), is larger than the average value of its neighbors, but most of its neighbors are going to be increased which will make the average value larger than \( I_c(x) \), then increasing \( I_c(x) \) should be a better choice which is not chosen by CAS-D. This is an extreme example but could explain the problem. We then propose another embedding algorithm named CAS-NE (content-adaptive steganography via noise estimation), in which the dependency of neighboring pixels is emphasized. In this scheme, we choose the modification as the one that can minimize the noise contained in stego image:

\[
S^* = \arg\min_{S \in S} F(I_c + S),
\]

where \( F(I) \) is a function that evaluates the "noisy level" of an image \( I \). The larger value \( F \) has, the noisier the image \( I \) is. Clearly, there are many such noise estimation functions, and different functions may result in steganographic algorithm with different levels of security. In this paper, we limit our discussion and take the function \( F \) as

\[
F(I) = \sum_{x \in D} \sum_{x' \in N_x} \lambda_{x,x'} (I(x) - I(x'))^2,
\]

where \( D \) is the set of all pixels of image \( I \), \( N_x \) is the set of neighboring pixels of \( x \), and \( \lambda_{x,x'} \) are pre-defined weighted parameters. In this case, the term \( F(I_c + S) \) can be rewritten as

\[
F(I_c + S) = \sum_{x \in D, x' \in N_x} \lambda_{x,x'} \left( (I_c(x) - I_c(x'))^2 + (S(x') - S(x))^2 + 2(I_c(x) - I_c(x'))(S(x) - S(x')) - 2S(x)S(x') \right).
\]

Note that for any \( x \in D \), \( I_c(x) \) and \( S(x) \) are constants. Thus we can get

\[
F(I_c + S) = 2G(I_c, S) + C,
\]

where \( C \) is a constant and \( G(I_c, S) \) is the following function

\[
\sum_{x \in D, x' \in N_x} \lambda_{x,x'} \left( (I_c(x) - I_c(x'))(S(x) - S(x')) - S(x)S(x') \right).
\]

Moreover, we can further express \( G(I_c, S) \) as

\[
\sum_{x \in D} S(x) \left( \sum_{x' \in N_x} \lambda_{x,x'} (I_c(x) - I_c(x')) - \sum_{x' \in N_x} \lambda_{x,x'} S(x') \right) - \sum_{x \in D, x' \in N_x} \lambda_{x,x'} S(x')(I_c(x) - I_c(x')).
\]
If \( N_x \) satisfies the following reasonable condition: \( x' \in N_x \Leftrightarrow x \in N_{x'} \), and the weighted parameters \( \lambda_{x,x'} \) are symmetric: \( \lambda_{x,x'} = \lambda_{x',x} \). Then, we can prove

\[
\sum_{x \in D} \sum_{x' \in N_x} \lambda_{x,x'} S(x')(I_c(x) - I_c(x')) = -\sum_{x \in D} S(x) \sum_{x' \in N_x} \lambda_{x,x'} (I_c(x) - I_c(x')).
\]

It yields that

\[
G(I_c, S) = \sum_{x \in D} S(x) \left( 2\Delta(x) - \sum_{x' \in N_x} \lambda_{x,x'} S(x') \right),
\]

where \( \Delta(x) = \sum_{x' \in N_x} \lambda_{x,x'} (I_c(x) - I_c(x')) \) is a constant.

In summary, reviewing Eq.(1) and (2), our goal is to find a modification \( S^* \in S \) such that:

\[
S^* = \arg \min_{S \in S} G(I_c, S),
\]

where \( G \) is a function defined in Eq.(3).

We remark that Eq.(4) can be viewed as a quadratic programming. Nevertheless, the usual way such as ellipsoid method [13] can not resolve this problem efficiently because of the large problem dimension (note that this dimension is just the total number of changeable pixels). We now propose the following computationally efficient iterative method to determine a sequence \( S_1, S_2, \ldots \), such that \( G(I_c, S_i+1) \leq G(I_c, S_i) \), and we take \( S_i \) as the final result for a certain \( k \). We will see later that this compromising approach will give fairly good solution. First, all pixels of \( I_c \) are divided into several disjoint subsets: \( D = \bigcup_{i=1}^{m} D_i \), in such a way that for any \( x \in D_i \), we have \( N_x \cap D_i = \emptyset \). In other words, a pixel and its neighbors can not be contained in the same subset. For example, when \( N_x \) is defined as the 8-neighborhood for all \( x = (x_1, x_2) \in D \):

\[
N_x = \{(x_1 + 1, x_2), (x_1 - 1, x_2), (x_1, x_2 + 1), (x_1, x_2 - 1)\}.
\]

We can divide the pixels into 2 disjoint subsets: \( D = D_1 \cup D_2 \), with \( D_1 = \{(x_1, x_2) \in D : x_1 + x_2 \equiv 0 \mod 2\} \). Another example, when \( N_x \) is defined as the 8-neighborhood for all \( x = (x_1, x_2) \in D \):

\[
N_x = \{(x_1 + i, x_2 + j) : i, j \in \{0, 1, -1\}, (i, j) \neq (0, 0)\}.
\]

We can divide the pixels into 4 disjoint subsets: \( D = D_{00} \cup D_{01} \cup D_{10} \cup D_{11} \), with \( D_{ij} = \{(x_1, x_2) \in D : x_1 + x_2 \equiv i \mod 2, x_2 \equiv j \mod 2\} \). Then, for any \( i \in \{1, ..., m\} \), we can get

\[
G(I_c, S) = \sum_{x \in D_i} 2S(x) \left( \Delta(x) - \sum_{x' \in N_x} \lambda_{x,x'} S(x') \right)
+ \sum_{x \in D-D_i} S(x) \left( 2\Delta(x) - \sum_{x' \in N_x, x' \notin D_i} \lambda_{x,x'} S(x') \right).
\]

In the above expression of \( G(I_c, S) \), we see that for any \( x \in D_i \), \( S(x) \) is only appeared in the first term. More specifically, in order to minimize \( G(I_c, S) \), if the value of \( S(x) \) is fixed for any \( x \in D - D_i \), we may simply take \( S(x) \) as the opposite sign of \( (\Delta(x) - \sum_{x' \in N_x} \lambda_{x,x'} S(x')) \) for any changeable \( x \in D_i \). In this way, for a modification \( S \in S \), we propose to adjust an modified function \( f(S) \in S \) as follows. Let \( S^0 = S \). For \( i = 1, 2, ..., m \), we then successively define \( S^i \) by

- for any \( x \notin M : S^i(x) = 0 \),
- for any \( x \in M \cap (D - D_i) : S^i(x) = S^{i-1}(x) \),
- for any \( x \in M \cap D_i : S^i(x) = \begin{cases} 1 & \text{if } \Delta(x) < \sum_{x' \in N_x} \lambda_{x,x'} S^{i-1}(x') \\ -1 & \text{if } \Delta(x) > \sum_{x' \in N_x} \lambda_{x,x'} S^{i-1}(x') \\ \pm 1 & \text{randomly} \end{cases} \text{ if } \Delta(x) = \sum_{x' \in N_x} \lambda_{x,x'} S^{i-1}(x') \).

Whereafter, we take \( f(S) = S^m \). Consequently, we have

\[
G(I_c, f(S)) \leq G(I_c, S).
\]

Now, we summarize the data embedding procedure of CAS-NE.

First, randomly select a modification \( S_0 \in S \). Then we take successively \( S_1 = f(S_0), S_2 = f(S_1), \ldots \). The iteration will stop if \( G(I_c, S_k) \geq G(I_c, S_{k-1}) \), and then we take \( (I_c + S_k) \) as the final stego image. In our test, the above procedure usually stops in less than 10 iterations, and the data embedding procedure can be finished in seconds (for \( 700 \times 500 \) gray-scale images, the average embedding time is approximately two seconds when taking 8-neighborhood as \( N_x \)). The corresponding extraction procedure of CAS-NE is just the same as CAS-D and LSB matching.

Before closing this section, some primary experimental results are presented. Fig.2 shows the comparison of ROC curves between CAS-NE and LSB matching for resisting the WAM steganalyzer [2]. Here, the set \( N_x \) is taken as 4-neighborhood or 8-neighborhood, and \( \lambda_{x,x'} = \frac{1}{d(x,x')} \), where \( d(\cdot) \) is the Euclidean distance. It can be observed that CAS-NE is also harder to detect than LSB matching, especially for large neighborhood. More experimental results including the comparison between CAS-D and CAS-NE will be reported in the next section.

### 3. EXPERIMENTAL RESULTS

The experiments are conducted as follows. Firstly, the test image sets include: (1) Image Set 1: 3000 images are downloaded from the USDA NRCS Photo Gallery.\(^1\) For testing, we resampled each of them to the 1/3 of the original size (the size of the resulting images are about \( 700 \times 500 \)) and converted each image to grayscale. These images are scanned from a variety of films and paper sources, which appear noisy to some extent. (2) Image Set 2: This set contains 5000 images in good quality. These images are collected from several types of digital cameras and then resampled to make all the images in the size from 400 \( \times \) 400 to 800 \( \times \) 800 and converted into grayscale. (3) Image Set 3: To test the sensitivity depending imagery [14], the above two image sets are collected together to get totally 8000 images. Secondly, several steganographic algorithms are used to embed random generated secret message into the above images, and the embedding rate is fixed at 1 for the following methods (1)-(4). The algorithms include: (1) CAS-D with 5 \( \times \) 5 sized Wiener filter, (2) CAS-NE with 8-neighborhood as neighboring pixel set, (3) LSB matching, (4) G-LSB-M with index 6 [9], (5) Improved PVD (pixel-value-based) embedding [15]. Note that here the G-LSB-M is a generalization of LSB matching by increasing its embedding efficiency from 2 (for LSB matching) to 1.03021 = 3.31 at embedding rate 1. The improved PVD embedding is an extension of the original PVD based steganography [16] which is also content-based steganography. For the improved PVD embedding, we first

\(^1\)http://photogallery.ncrs.usda.gov
divide the range $[0, 255]$ into two intervals: $R_1 = [0, 31]$ (lower-level range) and $R_2 = [32, 255]$ (higher-level range). One bit is then embedded into each pixel from lower-level range and two bits are embedded into each pixel from higher-level. In other words, for improved PVD embedding, we use 1-2 division with dividing line $D_{12} = 32$, and the resulting average embedding rate is respectively 1.1 (for Image Set 1) and 1.06 (for Image Set 2). Thirdly, the blind steganalyzer, WAM, is used in our experiments. Then, we use SVM to train and test. In each experiment, we use 25% of the cover images and 25% of the stego images for training, to detect the remaining 75% cover and stego images. The procedure is repeated 10 times for cross-validation and the ROC curves are vertically averaged to obtain the mean performance of the schemes. The resulting ROC curves are shown in Fig.3. From this figure, we can see that: (1) the proposed schemes CAS-D and CAS-NE are more secure than LSB matching, G-LSB-M and improved PVD embedding, (2) CAS-NE is the best one for resisting steganalysis and it can not be detected by WAM. In particular, CAS-NE is better than G-LSB-M. This phenomenon can be explained by the fact that embedding efficiency reflects only the amount of modifications, which does not consider the structures in natural images.

4. CONCLUSION

In this paper, different from the previous approaches for enhancing steganographic security, a content-adaptive steganographic scheme was proposed. In this scheme, by exploiting the embedding redundancy in LSB matching, we tried to select the most smooth one as the final stego image among all candidate stego images. The resulting stego image looks more like a natural one and hard to be perceived by steganalysis. Consequently, the conventional LSB matching was improved by enhancing its security. Following the approach proposed in this paper, there are many subsequent work to do, e.g., the more meaningful choice of the object function $F(I)$, finding the solution to the optimization problem given by Eq.(1) or Eq.(4) in acceptable time, etc. Finally, it should be noted that the proposed approach can be applied to any LSB-based steganographic algorithm (for instance, the well-known matrix embedding [6]). Since both increasing and decreasing the pixel value by 1 are valid when modifying pixels in these algorithms, the proposed method can be used to wisely select the increment or decrement for each changeable pixel and to reduce embedding impact.

5. REFERENCES


