# Constructing Near-optimal Double-layered Syndrome-Trellis Codes for Spatial Steganography


Zengzhen Zhao
State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
zhaozengzhen@iie.ac.cn

Qingxiao Guan*
State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
guanqingxiao@iie.ac.cn

Xianfeng Zhao
State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
zhaoxianfeng@iie.ac.cn



## ABSTRACT

In this paper, we present a new kind of near-optimal double-layered syndrome-trellis codes (STCs) for spatial domain steganography. The STCs can hide longer message or improve the security with the same-length message comparing to the previous double-layered STCs. In our scheme, according to the theoretical deduction we can more precisely divide the secret payload into two parts which will be embedded in the first layer and the second layer of the cover respectively with binary STCs. When embed the message, we encourage to realize the double-layered embedding by $\pm 1$ modifications. But in order to further decrease the modifications and improve the time efficient, we allow few pixels to be modified by $\pm 2$. Experiment results demonstrate that while applying this double-layered STCs to the adaptive steganographic algorithms, the embedding modifications become more concentrative and the number decreases, consequently the security of steganography is improved.



## Keywords

Information hiding; steganography; additive distortion; security; syndrome-trellis codes


## 1. INTRODUCTION

In steganography, secret message is embedded in a cover which is delivered to receiver through public channel but arouses no suspicion to others. Digital image is the most popular media on the internet and the ideal cover for hiding message. Steganographic schemes for digital images can realize the embedding by changing the pixels or DCT coefficients. As a countermeasure, the steganalyzers can attack this by using statistical detectability of embedding changes, such as using Rich Models [6] with high-dimensional feature spaces. In order to resist this kind of steganalysis, adaptive stegonagraphic scheme by minimizing additive distortion in steganography is the state-of-the-art scheme. Actually, the additive distortion introduced by embedding can be regarded as the sum of costs of all changed pixels, and the costs values assigned to pixels evaluate the disturbance of modification. Typically, the costs values can be obtained by distortion calculation. Recent works of image steganography concentrate on this. And several distortion designed methods for image steganography have been proposed, such as HUGO [14], UNIWARD [8], and HILL [9]. What' more, minimizing additive distortion depends on the STCs.

In [3], [11], Filler *et al.* proposed the STCs framework for minimizing distortion. It includes binary STCs and multi-layered STCs. Among multi-layered STCs, the double-layered STCs are more feasible to enhance the security. Double-layered STCs utilize the least two significant bits of cover for embedding. Comparing to single-layered STCs (binary STCs), double-layered STCs still use $\pm 1$ modification but remarkably reduce the number of modifications. As a result fewer disturbances are brought to steganalysis feature. Also, the max payload rate for double-layered STCs is larger than single-layered STCs whose largest payload rate is 0.5.

The ideal way of constructing double-layered STCs is using 3-ary STCs encoding because when we have to convert a cover unit into any other value in 3-ary, it can be done by $+1$ or $-1$ modification. However, due to the high computational complexity and large storage space, 3-ary STCs are not very practical. Alternatively, double-layered STCs are implemented by applying binary STCs on two layers. Filler *et al.* proposed a framework for multi-layered STCs in [5] based on chain rules of decomposing multi-layered entropy. The double-layered STCs in this framework can realize the ternary embedding. Firstly the probability of ternary modification on each cover data unit is calculated by solving the equation between ternary entropy and message length. Then the costs values of two layers are computed from their modification probabilities, and STCs encoding is carried out twice on two layers respectively with their costs. For pixels need to be modified, $+1$ or $-1$ operation is applied on them to make their least two significant bits conform to the final embedding result of double-layered STCs. This method derives from optimal ternary embedding theory and has received excellent result. Since the double-layered STCs only allow $\pm 1$ modification and double-layered embedding performs individually, it requires the final embedding result can be achieved within $\pm 1$ modification. Consequently, flipping any bit in the second layer is accompanied by that in the first layer. If a pixel requires more than $\pm 1$ modification, this is a wet pixel [12], [17] and it will stop the embedding. In this case, Filler *et al.* suggest permutating the


Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.


*IH&MMSec 2016, June 20-23, 2016, Vigo, Spain .*
© 2016 ACM. ISBN 978-1-4503-4290-2/16/06…$15.00.
DOI: http://dx.doi.org/10.1145/2909827.2930802


* the corresponding author

order of cover pixels with different random seed and try STCs encoding again. Filler has also proposed pentary STCs in [5], which are implemented by applying binary STCs on two layers too. Since too many $\pm 2$ modifications are unable to effectively resist the steganalysis by high-dimensional feature spaces, pentary STCs are not popular in adaptive steganographic algorithms.

In this paper we adopt a new scheme for double-layered STCs, which is same to the scheme in [10]. This scheme is rather simple and understandable. We also divide double-layered STCs into two parts for two layers respectively. In the first layer, binary STCs are applied with costs. In the second layer, for those pixels need to be flipped in the first layer, we set their costs to be zero (zero cost pixels) and apply binary STCs on the second layers with revised costs. Finally, pixels are modified by $\pm 1$ according to double-layered encoding results. This scheme implies that the embedding in the second layer is an extra product. Since we can control the flipping of a bit in the second layer by $\pm 1$ operation when the bit in the first layer has to be flipped, the modifications in the second layer cost no more distortion.

Due to the properties of STCs encoding, the embedding in second layer will make the modifications occur in zero cost pixels as possible as it could, which means that it encourages the overlapping of modified pixels for two layers. In ideal condition, all the modified pixels in the second layer should be included in the set of modified pixels in the first layer, which means all modifications can be done by $\pm 1$ operation in this case. Hence, in our method there is a key problem that for a message of fixed length, how many bits should be distributed to two layers. The success of embedding in the second layer depends on the number of zero cost pixels. If we embedding too many bits in the first layer, it will produces more zero cost pixels for second layer. However, it may cause a waste of embedding resource and make excessive modifications. Otherwise, too few bits embedded in the first layer may leave insufficient zero cost pixels for second layer and cause too many wet pixels. There is an optimal strategy to distribute bits to two layers. We analyze this problem by an embedding model presented in this paper.

The second issue of this paper is that how to assign the cost for the second layer. If we rigorously stick to $\pm 1$ modification, there are only two kinds of pixel for the second layer: dry pixels (zero cost pixels) and wet pixels. However, it is possible that embedding in second layer may fail because of the wet pixels. Although we can use similar strategy of trying different seed to permute cover data as Filler *et al.* proposed in [3], we also investigate another way to deal with it because the trials of STCs encoding is time cost, especially for images of larger size. It is the simple way that we can allow few wet pixels to be modified by $\pm 2$ operation. This is simple and straightforward in our method. The problem equals to assigning proper cost value for wet pixels of second layer. We also analyze this case in this paper both theoretically and experimentally. We can see that the number of modifications on wet pixels conforms to the normal distribution whose mean and variance depend on the cost of second layer. If we want to minimize its mean value, we are inevitably to enlarge its variance. And experiment shows that $\pm 2$ modifications on few wet pixels do not degrade its security, and it is better to assign cost for wet pixels according to *complexity-first rule* [13].

In this paper we name our proposed STCs as NDSTCs (the Near-optimal Double-layered STCs). Experimental results demonstrate that our new double-layered STCs can directly be combined with the adaptive steganographic algorithms in spatial domain and make the performance approach the optimal ternary embedding simulator.

**Table 1: The flipping situation after doing $\pm 1$ operations on the least three bits.**

| $x_{i,j}$ | Flipping LSB | | Flipping LSB and 2LSB | |
|---|---|---|---|---|
| | operation | result | operation | result |
| **000** | +1 | 001 | -1 | 111[1] |
| **001** | -1 | 000 | +1 | 010 |
| **010** | +1 | 011 | -1 | 001 |
| **011** | -1 | 010 | +1 | 100 |
| **100** | +1 | 101 | -1 | 011 |
| **101** | -1 | 100 | +1 | 110 |
| **110** | +1 | 111 | -1 | 101 |
| **111** | -1 | 110 | +1 | 000[1] |

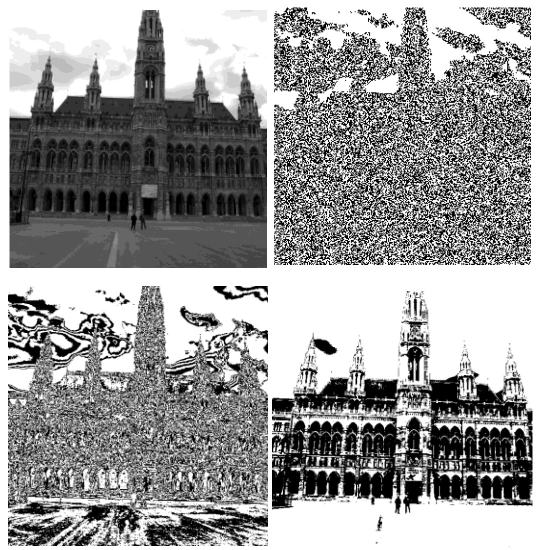

**Figure 1: The proportion each bit plane accounting for BOSSbase image '1029.pgm'. Top-left: original image, Top-right: LSB plane, bottom-left: 4LSB plane, bottom-right: 7LSB plane.**

The rest of this paper is organized as follows. The preparation is described in Section 2. In Section 3, the proposed NDSTCs is introduced. Experimental results and comparison at the best parameters are illustrated in Section 4. We further analysis the result in Section 5. Conclusion and future research directions are listed in Section 6. Appendix is showed in Section 7.

# 2. PREPARATION

## 2.1 Notation

Without loss of generality, let $X = (x_{i,j}) \in \{\mathcal{L}\}^{n_1 \times n_2}$ represent $n_1 \times n_2$ pixels cover image and $Y = (y_{i,j}) \in \{\mathcal{L}\}^{n_1 \times n_2}$ represent the stego image after embedding operation on cover X. $\mathcal{L}$ is the pixel dynamic range of image. For example, $\mathcal{L} = \{0, \cdots, 255\}$ for 8-bit grayscale image. So $x_{i,j}$ is the pixel in $(i, j)$ location. $y_{i,j}$ can be

1 Carry in or borrow from the former bit.

got by modifying $x_{i,j}$ by all most $\pm 1$ because of the characteristic of adaptive steganographic scheme. For more detailed representation to images, pixels can be converted to binary representation

$$x_{i,j} = \sum_{l=1}^{8} x_{i,j}^{l} \times 2^{l-1}, \quad (1)$$

where $x_{i,j}^{l} \epsilon \{0,1\}$ is the $l$ bit of pixel in $(i,j)$ location.

Based on formula (1), we can obtain 8 different bit planes for each image. Figure 1 shows that different bit plane contributes to image in different proportion, modifying the bit plane with high proportion means bringing about more impact to images. So reducing the disturbance to high bit plane is still our principle.

Let's consider the least three bits of image, the operation to them is obvious in Table 1. For most of the pixels we can achieve the least two bits flipping by modifying pixels by $\pm 1$, which means every pixel can carry more than just one bit. It can carry $\log_2 3$ bits of information [10]. However when the value of pixel is in the boundary of $\mathcal{L}$, modifying each pixel by $\pm 3$ to avoid pixel overstepping the boundary is needed but what we should avoid. Actually some images consisting of too many pixels of this kind are unsuited to be the carrier images for this reason.

## 2.2 Preliminary of Single-layered STCs

The adaptive steganographic scheme is to minimize the following additive distortion function.

$$D(X^l, Y^l) = \sum_{i,j} \rho_{i,j} \left(x_{i,j}^{l} \neq y_{i,j}^{l}\right), \quad (2)$$

where $\rho_{i,j}$ is the cost of flipping $x_{i,j}^{l}$, meanwhile the cost being sent to binary STCs, can be derived from the adaptive steganographic algorithm. Actually when we embed payload $m$ to the pixels with cost $\rho$, we have constructed a probability description via

$$\beta_{i,j} = \frac{e^{-\lambda \rho_{i,j}}}{1+e^{-\lambda \rho_{i,j}}}, \quad (3)$$

where $\lambda > 0$ is a parameter determined from embedding payload $m$ and cost $\rho$.

In theory, with the change probability $\beta$ we can get the embedding payload $m$ via

$$|\, m| = \sum_{i,j} H\left(\beta_{i,j}\right), \quad (4)$$

where $H(\beta_{i,j})$ is the entropy function of the change probability $\beta_{i,j}$.

$$H\left(\beta_{i,j}\right) = -\beta_{i,j} \log_2\left(\beta_{i,j}\right) - \left(1-\beta_{i,j}\right) \log_2\left(1-\beta_{i,j}\right). \quad (5)$$

With embedding payload $m$ and the cost $\rho$, $\lambda$ can be calculated using formula (3), (4), (5) after iteration operation.

We can interpret the probability description as a model building a connection between embedding payload $m$ and the cost $\rho$, which is the key point to execute the theoretical derivation.

# 3. PROPOSED WORK

In this section, we provide a general description of the proposed NDSTCs framework and give the theory explaining how to divide total payload to two parts respectively embedded in first layer and the second layer of the cover. To better understand our proposed work, we also show the practical implementation and explain some details that have deep connection with practical.

## 3.1 Theoretical Derivation

We assume the total payload $m$ are divided into two pieces $m_1$, $m_2$ with $m = m_1 \cup m_2$. Apply the probability description to first layer and for each pixel the probability of change its first bit can be expressed as

$$\beta_{i,j}^{1} = \frac{e^{-\lambda_1 \rho_{i,j}}}{1+e^{-\lambda_1 \rho_{i,j}}}. \quad (6)$$

It is clearer to define the change of a pixel in the first layer as a random variable $\mathrm{c}_{i,j}$, and it conforms to binary distribution:

$$\mathrm{c}_{i,j} \sim \begin{cases} 1 & P\left(\mathrm{c}_{i,j}\right) = \beta_{i,j}^{1} \\ 0 & P\left(\mathrm{c}_{i,j}\right) = 1-\beta_{i,j}^{1} \end{cases}. \quad (7)$$

$\beta_{i,j}^{1}$ depends on two quantities $\rho_{i,j}$ and $\lambda_1$. Where $\rho_{i,j}$ is calculated from image and $\lambda_1$ is an unknown parameter. We denote the number of embedding modifications in first layer as $N_1$ and the length of the theoretical embedding payload in the first layer as $|\, m_1|$.

$$N_1 = \sum_{i,j} \mathrm{c}_{i,j}, \quad (8)$$

$$E(N_1) = \sum_{i,j} E(\mathrm{c}_{i,j}) = \sum_{i,j} \frac{e^{-\lambda_1 \rho_{i,j}}}{1+e^{-\lambda_1 \rho_{i,j}}}, \quad (9)$$

$$|\, m_1| = \sum_{i,j} -\frac{e^{-\lambda_1 \rho_{i,j}}}{1+e^{-\lambda_1 \rho_{i,j}}} \log_2 \frac{e^{-\lambda_1 \rho_{i,j}}}{1+e^{-\lambda_1 \rho_{i,j}}} + \sum_{i,j} -\left(1-\frac{e^{-\lambda_1 \rho_{i,j}}}{1+e^{-\lambda_1 \rho_{i,j}}}\right) \log_2 \left(1-\frac{e^{-\lambda_1 \rho_{i,j}}}{1+e^{-\lambda_1 \rho_{i,j}}}\right). \quad (10)$$

Yet we are unable to figure out $\lambda_1$ because $|\, m_1|$ is still an unknown parameter. Suppose that after embedding in the first layer, we can get the embedding modifications $C_1$ with $|C_1| = E(N_1)$. From the section 2.1, we know that flipping two bits can be realized with modifying pixels by $\pm 1$ and we want the embedding modifications in the second layer overlap with the first layer as large as possible. Therefore, for the embedding in the second layer, we set the cost of pixels in $C_1$ to zero and get the new cost $\rho'$. This indicates that for pixels in $C_1$, the modifications in the second layer is "free of cost". We once more use the binary entropy to analyze STCs encoding in second layer. The number of bits in $m_2$ can be expressed like (10), and ideally the bits in the second layer can be divided into two parts $C_1$ and $\widetilde{C_1}$. Therefore we formulate it in following equation.

$$|m_2| = \sum_{(i,j)\in C_1} \begin{bmatrix} -\frac{e^{-\lambda_2 \rho'_{i,j}}}{1+e^{-\lambda_2 \rho'_{i,j}}} \log_2 \frac{e^{-\lambda_2 \rho'_{i,j}}}{1+e^{-\lambda_2 \rho'_{i,j}}} \\ -\left(1-\frac{e^{-\lambda_2 \rho'_{i,j}}}{1+e^{-\lambda_2 \rho'_{i,j}}}\right) \log_2 \left(1-\frac{e^{-\lambda_2 \rho'_{i,j}}}{1+e^{-\lambda_2 \rho'_{i,j}}}\right) \end{bmatrix} + \sum_{(i,j)\in \widetilde{C_1}} \begin{bmatrix} -\frac{e^{-\lambda_2 \rho'_{i,j}}}{1+e^{-\lambda_2 \rho'_{i,j}}} \log_2 \frac{e^{-\lambda_2 \rho'_{i,j}}}{1+e^{-\lambda_2 \rho'_{i,j}}} \\ -\left(1-\frac{e^{-\lambda_2 \rho'_{i,j}}}{1+e^{-\lambda_2 \rho'_{i,j}}}\right) \log_2 \left(1-\frac{e^{-\lambda_2 \rho'_{i,j}}}{1+e^{-\lambda_2 \rho'_{i,j}}}\right) \end{bmatrix}. \quad (11)$$

Since $\rho'_{i,j} = 0,\ \forall\ (i,j) \in C_1$, we substitute it into (11) and yield:

$$|m_2| = \sum_{(i,j)\in C_1} \left[-\frac{1}{2}\log_2 \frac{1}{2} - \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right] + \sum_{(i,j)\in \widetilde{C_1}} \begin{bmatrix} -\frac{e^{-\lambda_2 \rho'_{i,j}}}{1+e^{-\lambda_2 \rho'_{i,j}}} \log_2 \frac{e^{-\lambda_2 \rho'_{i,j}}}{1+e^{-\lambda_2 \rho'_{i,j}}} \\ -\left(1-\frac{e^{-\lambda_2 \rho'_{i,j}}}{1+e^{-\lambda_2 \rho'_{i,j}}}\right) \log_2 \left(1-\frac{e^{-\lambda_2 \rho'_{i,j}}}{1+e^{-\lambda_2 \rho'_{i,j}}}\right) \end{bmatrix}$$

$$\xRightarrow{simplify} |C_1| + \sum_{(i,j)\in \widetilde{C_1}} \begin{bmatrix} -\frac{e^{-\lambda_2 \rho'_{i,j}}}{1+e^{-\lambda_2 \rho'_{i,j}}} \log_2 \frac{e^{-\lambda_2 \rho'_{i,j}}}{1+e^{-\lambda_2 \rho'_{i,j}}} \\ -\left(1-\frac{e^{-\lambda_2 \rho'_{i,j}}}{1+e^{-\lambda_2 \rho'_{i,j}}}\right) \log_2 \left(1-\frac{e^{-\lambda_2 \rho'_{i,j}}}{1+e^{-\lambda_2 \rho'_{i,j}}}\right) \end{bmatrix}. \quad (12)$$

The change probability of pixels in $\widetilde{C_1}$ relies on $\lambda_2$ which is determined by equation (11). By observing the equation (12), we can deduce that if and only if $|C_1| = |m_2|$, the following equation is satisfied:

$$\sum_{(i,j)\in\widetilde{C_1}}\left[\begin{array}{c}-\frac{e^{-\lambda_2\rho'_{i,j}}}{1+e^{-\lambda_2\rho'_{i,j}}}\log_2\frac{e^{-\lambda_2\rho'_{i,j}}}{1+e^{-\lambda_2\rho'_{i,j}}}\\-\left(1-\frac{e^{-\lambda_2\rho'_{i,j}}}{1+e^{-\lambda_2\rho'_{i,j}}}\right)\log_2\left(1-\frac{e^{-\lambda_2\rho'_{i,j}}}{1+e^{-\lambda_2\rho'_{i,j}}}\right)\end{array}\right]=0. \quad (13)$$

While the ideal condition is that pixels in $\widetilde{C_1}$ are kept unchanged, it requires the change probability on the second layer of pixels in $\widetilde{C_1}$ equal to 0. Considering the properties of entropy and the fact that $\lambda_2 > 0$ and $\rho'_{i,j} > 0 \,\forall\, (i,j) \in \widetilde{C_1}$ , above deduction tell us that while $|C_1| = |m_2|$ is satisfied, $\frac{e^{-\lambda_2\rho'_{i,j}}}{1+e^{-\lambda_2\rho'_{i,j}}} \to 0, \forall\, (i,j) \in \widetilde{C_1}$ with $\lambda_2 \to \infty$, which means the length of the ideal embedding payload in the second layer $|m_2|$ equals to the number of embedding modifications in the first layer $N_1$, and theoretical optimal length of $m_1$ and $m_2$ can be solved. Therefore we substitute $|C_1| = |m_2|$ and $|C_1| = E(N_1)$ into (9) and together with (10), the number of the total embedding payload $|m|$ can be expressed as:

$$|m| = |m_1| + |m_2|$$

$$= \sum_{i,j}\left[\begin{array}{c}-\frac{e^{-\lambda_1\rho_{i,j}}}{1+e^{-\lambda_1\rho_{i,j}}}\log_2\frac{e^{-\lambda_1\rho_{i,j}}}{1+e^{-\lambda_1\rho_{i,j}}}\\-\left(1-\frac{e^{-\lambda_1\rho_{i,j}}}{1+e^{-\lambda_1\rho_{i,j}}}\right)\log_2\left(1-\frac{e^{-\lambda_1\rho_{i,j}}}{1+e^{-\lambda_1\rho_{i,j}}}\right)\end{array}\right] + \sum_{i,j}\frac{e^{-\lambda_1\rho_{i,j}}}{1+e^{-\lambda_1\rho_{i,j}}}, \quad (14)$$

where the only unknown parameter is the flipping lambda $\lambda_1$. A simple binary search can be used to solve this equation because both $|m_1|$ and $N_1$ are monotone with regard to $\lambda_1$. After getting the flipping lambda $\lambda_1$, we can obtain the $|m_1|$ by (10), and subsequently $|m_2|$ is obtained by $|m| - |m_1|$. See from the total derived process, the flipping lambda $\lambda_1$ is obtained on an ideal condition that the embedding modifications in the second layer totally overlap with the first layer.

So far we haven't discuss more details of the cost assignment strategy for the second layer. According to above analysis, it doesn't matter if we assign arbitrary positive values to pixels in $\widetilde{C_1}$, and their change probability is 0 since $\lambda_2 \to \infty$. Actually, if there is a trellis path which only need to flips the second layer of some pixels in $\widetilde{C_1}$, STCs encoder prefers to skip it because the overall distortion cost of $C_1$ is zero which is smaller than cost of any pixel in $\widetilde{C_1}$. However, as we mentioned in the first section, if we somehow relax the $\pm 1$ restriction and allow few $\pm 2$ modifications in $\widetilde{C_1}$ for some trials, the assignment of cost for the second layer matters. And this part is discussed in section 3.2.

## 3.2 Practical Implementation

In previous section, we derived the theoretic optimal message distribution strategy for two layers. However, in the real world the embedding process may not precisely fit the theoretic model. Based on theoretic model, it is reasonable to readjust the proportion of message bits embedded into two layers. It can be implemented by adding a scalar factor in (14). We replace the term $\sum_{i,j}\frac{e^{-\lambda_1\rho_{i,j}}}{1+e^{-\lambda_1\rho_{i,j}}}$ in (14) with $\tau \times \sum_{i,j}\frac{e^{-\lambda_1\rho_{i,j}}}{1+e^{-\lambda_1\rho_{i,j}}}$, and reformulate the equation (14) as:

$$|m| = \sum_{i,j}\left[\begin{array}{c}-\frac{e^{-\lambda_1\rho_{i,j}}}{1+e^{-\lambda_1\rho_{i,j}}}\log_2\frac{e^{-\lambda_1\rho_{i,j}}}{1+e^{-\lambda_1\rho_{i,j}}}\\-\left(1-\frac{e^{-\lambda_1\rho_{i,j}}}{1+e^{-\lambda_1\rho_{i,j}}}\right)\log_2\left(1-\frac{e^{-\lambda_1\rho_{i,j}}}{1+e^{-\lambda_1\rho_{i,j}}}\right)\end{array}\right] + \tau \times \sum_{i,j}\frac{e^{-\lambda_1\rho_{i,j}}}{1+e^{-\lambda_1\rho_{i,j}}}. \quad (15)$$

**Algorithm 1 Embedded procedure for near-optimal double-layered STCs embedding framework. Input cover image X, payload $m$, the permutation-key $\kappa$ and the adaptive steganographic algorithm $\mathcal{C}$.**

Require: $\mathrm{X} = (x_{i,j}) \in \{\mathcal{L}\}^{n_1\times n_2} \triangleq \{0,\cdots,255\}^{n_1\times n_2}$

1. Calculate the costs $\rho = \mathcal{C}(\mathrm{X})$
2. Calculate the flipping lambda $\lambda_1$ through solving the formula (15) with the inputs $\rho$ and $m$
3. Get the number of bits hidden in the first layer $|m_1|$ according formula (10) with the input $\lambda_1$
4. Embed $|m_1|$ bits messages into $X^1$ with the costs $\rho$ and the permutation-key $\kappa$ using binary STCs, then get the new vector $Y^1$.
5. If the step 4 is successful, continue the step 6. But if not, adjust the cost $\rho$ and reprocess the step 2
6. Define $\rho' = \rho$ and set $\rho'(X^1 \neq Y^1) = 0$
7. Get the number of bits hidden in the second layer $|m_2| = |m| - |m_1|$.
8. Embed $|m_2|$ bits messages into $X^2$ with costs $\rho'$ and the permutation-key $\kappa$ using binary STCs, then get the new vector $Y^2$
9. Update stego image Y with $Y^1$, $Y^2$
10. Return stego image Y and the number of bits hidden in the two layers $|m_1|$, $|m_2|$.

Method solving this equation is same to the formula (14). We can see the only difference between the sum of embedding modifications in the first layer $N_1$ and the length of the embedding payload in the second layer $|m_2|$ is the multiplying factor $\tau$. When $\tau = 1$, it is the ideal model described in section 3.1. Larger $\tau$ means we decrease the payload and number of modifications in the first layer. The reason we put $\tau$ on the second term of (15) is that it can more smoothly adjust payload and modifications in the first layer. Since the binary STCs is a near-optimal coding scheme thus inevitably there is a small gap between optimal coding and real STCs coding scheme. When embed bits in each layer, it will increase the modifications comparing to the optimal, which means the real number of modifications is larger than theoretic result $N_1$. And this leads to the enlargement of $\tau$. We will give the best value of $\tau$ based on experiment and interpret it in section 4.

We have proposed that in order to make sure the success of the embedding, we allow a few wet pixels in the second layer to be modified by $\pm 2$, which won't effect the performance in significant manner. Since the modifications are slight, two cases [13] of assigning cost for the wet pixels can be considered here:

Case 1 – change rate minimization. The cost-value distribution of the wet bits follows the Dirac distribution, which means all cost-values are the same value. In this case, the embedding distortion is the number of modifications.

Case 2 – content adaptivity maximization. This follows the *complexity-first rule*, which requires keeping the cost-value varied according to the local content and unpredictability in an image, but leads to more modifications comparing to case 1.

Interestingly, for any cost assignment strategy, the number of modifications conforms to normal distribution whose variance and mean depend on the cost of pixels. In Case 1, although we can minimize the expectation number of modifications, we have to accept higher variance of it, which implies there is more uncertainty for number of modifications. Although Case 2 increases the number of modifications, it not only makes cost assignment follows the *complexity-first rule*, but also reduced

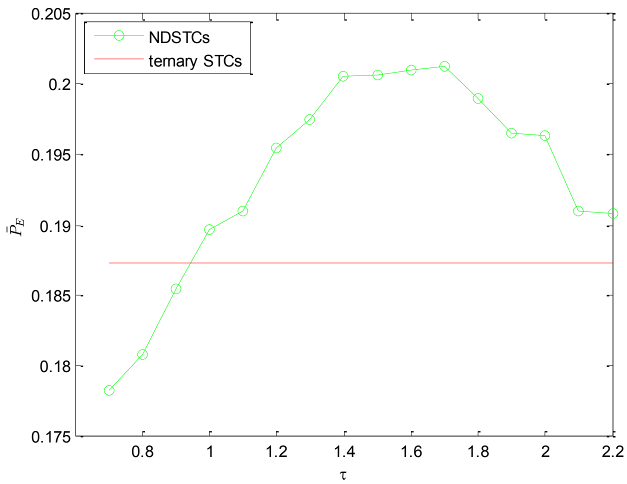


**Figure 2: Search for the optimal value of $\tau$ for S-UNIWARD with maxSRMd2 at 0.4bpp.**

uncertainty comparing to Case 1. Experimental results demonstrate that content adaptivity maximization has a better performance here. We elaborate the proof of these propositions in section 5 and appendix.

To better understand the implementation of our framework, we describe it using Algorithm 1. The inputs are the cover image X, the payload $m$, the permutation-key $\kappa$ and the adaptive steganographic algorithm $\mathcal{C}$, while the outputs are the stego image Y and the number of bits embedded in each layer.

In step 5, if the step 4 is not successful, we should adjust the cost $\rho$. This is because when apply our work to HILL, we find some images are unavailable. This situation is same to the ternary STCs. After carefully checking the source of HILL, we find that when HILL calculates the cost there is a reciprocal operation without an additive factor to avoid the cost tending to infinite. But S-UNIWARD has considered this. Since our work is to design the generic code and the unavailable images account for an extremely small amount, we just adjust the infinite cost of the unavailable images with an available value obtained by a simple wavelet filter. Moreover once the unavailable images appear in one of compared codes, the process to them should keep the same to keep the consistency for comparison. In step 9, updating the stego Y can be realized by replacing each layer of Y with $Y^1$and $Y^2$. If the least two bits of pixel is 11 or 00, it will bring about modifying pixel by $\pm 3$, which needs a correction through modifying pixel by $\pm 4$. However for the pixels with the boundary value of $\mathcal{L}$, the correction is unavailable. Due to the specialty of these pixels, we bring in the consideration of setting the uncorrectable cost to infinite when embedding in each layer.

The extract algorithm is not complex but need the payload length in each layer which can be communicated to the receiver using some pixels in the cover image or by other way. To keep the consistency for comparison, we employ the latter but don't care the communication.

# 4. EXPERIMENT AND COMPARISON

In this section, before giving the experimental result we describe some setups of the experiment. Then we test the performances of some parameters to our proposed work so that we can show the best experiment results.

## 4.1 Experimental Setup

Since Our near-optimal double-layered STCs framework is general to all adaptive steganographic algorithm in spatial domain.

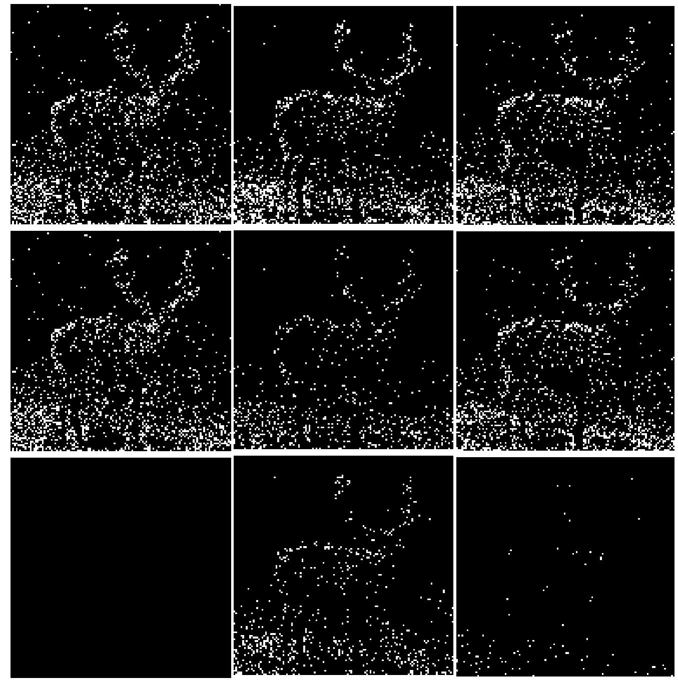

**Figure 3: The embedding modifications with S-UNIWARD algorithm at 0.4 bpp for the BOSSbase image '8.pgm'. From left to right represent the ternary STCs, the pentary STCs and the NDSTCs. From top to bottom represent the total modifications, modifications in the first layer, modifications only in the second layer, respectively.**

We utilize the BOSSbase database ver.1.01 [1] containing 10,000 512×512 8-bit grayscale images as the test image set, which has been widely used in steganography and steganalysis.

The ensemble classifier described in [2] is employed in our experiments. The parameters of ensemble classifier can be got by automatic calculation based on out-of-bag (OOB) estimating, so we needn't input any parameters but the features of cover and stego images. The classifier consists of a lot of base learners independently training on training set.

We split the whole feature set into two parts, training set and testing set. For cover samples and stego samples, there are 5000 images randomly selected as the training set, and rest 5000 images as the testing set. The security of steganographic schemes in this section is measured by the total detection error of steganalysis test. Detection error is the average error rate for two classes:

$$P_E = \min_{P_{FA}} \frac{1}{2}(P_{FA} + P_{MD}), \tag{16}$$

where $P_{FA}$, $P_{MD}$ are the probabilities of false alarms and missed detection, respectively. In fact, each experiment is repeated 10 times with different random split of training set and testing set and the average error rate $\bar{P}_E$ is utilized to evaluate the security of our proposed work.

For testing the detection error rate, the cover and stego images samples are analyzed by state-of-the-art feature-based steganalyzers, which are the Spatial Rich Model (SRM) and the maxSRMd2 [4]. The SRM consists of 39 symmetrized sub-models formed by different filters and co-occurrence matrices. It is also extended by quantifying with three different quantization factors to generate a 34,671D feature set. The maxSRMd2 is an "adaptive" version of the SRM. It makes use of the selection channel strategy which incorporates the modified probabilities obtained from the cost into co-occurrence matrix calculating. This

**Table 2: Detection error $\bar{P}_E$ for ternary STCs, pentary STCs, NDSTCs, simulator with HILL, S-UNIWARD, WOW in relative payloads**

| maxSRMd2 | | | | | | | |
|---|---|---|---|---|---|---|---|
| Embedding Strategy | 0.05 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 |
| HILL+ternary STCs | .4680±.0025 | .4237±.0022 | .3474±.0014 | .2801±.0017 | .2306±.0019 | .1893±.0018 | .1482±.0021 |
| HILL+pentary STCs | .4395±.0030 | .3811±.0023 | .2879±.0018 | .2203±.0025 | .1767±.0026 | .1364±.0023 | .1140±.0017 |
| HILL+NDSTCs | .4685±.0015 | .4265±.0020 | .3504±.0021 | .2886±.0025 | .2386±.0010 | .1961±.0025 | .1565±.0023 |
| HILL+simulator | .4694±.0026 | .4312±.0025 | .3561±.0018 | .2902±.0022 | .2379±.0020 | .1959±.0017 | .1563±.0019 |
| S-UNI+ternary STCs | .4479±.0022 | .3939±.0018 | .3061±.0024 | .2400±.0031 | .1873±.0030 | .1481±.0017 | .1120±.0017 |
| S-UNI+pentary STCs | .4157±.0020 | .3495±.0027 | .2545±.0014 | .1875±.0019 | .1440±.0028 | .1129±.0018 | .0903±.0022 |
| S-UNI+NDSTCs | .4509±.0017 | .3977±.0014 | .3133±.0017 | .2506±.0023 | .2016±.0021 | .1624±.0027 | .1287±.0019 |
| S-UNI+simulator | .4521±.0025 | .4043±.0039 | .3211±.0026 | .2519±.0018 | .1989±.0021 | .1570±.0029 | .1221±.0015 |
| WOW+ternary STCs | .4501±.0031 | .3910±.0027 | .3049±.0014 | .2376±.0018 | .1844±.0032 | .1504±.0029 | .1224±.0023 |
| WOW+pentary STCs | .4190±.0018 | .3495±.0011 | .2560±.0023 | .1911±.0020 | .1502±.0017 | .1210±.0016 | .0973±.0014 |
| WOW+NDSTCs | .4496±.0017 | .3924±.0013 | .3043±.0032 | .2434±.0022 | .1906±.0035 | .1562±.0027 | .1258±.0033 |
| WOW+simulator | .4560±.0032 | .4028±.0022 | .3171±.0027 | .2513±.0025 | .2012±.0017 | .1597±.0024 | .1273±.0031 |
| **SRM** | | | | | | | |
| Embedding Strategy | 0.05 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 |
| HILL+ternary STCs | .4695±.0028 | .4257±.0025 | .3510±.0013 | .2910±.0017 | .2350±.0015 | .1918±.0022 | .1605±.0017 |
| HILL+pentary STCs | .4437±.0020 | .3805±.0019 | .3006±.0016 | .2283±.0023 | .1830±.0034 | .1453±.0029 | .1205±.0023 |
| HILL+NDSTCs | .4683±.0021 | .4293±.0031 | .3564±.0025 | .2960±.0023 | .2458±.0015 | .2063±.0012 | .1651±.0022 |
| HILL+simulator | .4708±.0015 | .4345±.0021 | .3597±.0020 | .2965±.0018 | .2456±.0022 | .2023±.0022 | .1661±.0019 |
| S-UNI+ternary STCs | .4481±.0025 | .3971±.0019 | .3119±.0015 | .2430±.0027 | .1956±.0022 | .1569±.0017 | .1233±.0023 |
| S-UNI+pentary STCs | .4176±.0014 | .3470±.0019 | .2597±.0019 | .1950±.0027 | .1526±.0032 | .1200±.0018 | .0959±.0031 |
| S-UNI+NDSTCs | .4515±.0019 | .3985±.0024 | .3166±.0027 | .2587±0029 | .2072±.0016 | .1700±.0016 | .1384±.0037 |
| S-UNI+simulator | .4544±.0016 | .4041±.0023 | .3209±.0016 | .2552±.0019 | .2062±.0015 | .1664±.0020 | .1322±.0025 |
| WOW+ternary STCs | .4510±.0017 | .3945±.0045 | .3108±.0018 | .2491±.0015 | .1988±.0019 | .1611±.0017 | .1283±.0017 |
| WOW+pentary STCs | .4195±.0019 | .3520±.0036 | .2605±.0035 | .2001±.0015 | .1586±.0016 | .1266±.0014 | .1033±.0022 |
| WOW+NDSTCs | .4472±.0016 | .3939±.0023 | .3076±.0025 | .2491±.0033 | .2034±.0017 | .1625±.0023 | .1331±.0021 |
| WOW+simulator | .4534±.0017 | .4035±.0033 | .3192±.0024 | .2578±.0019 | .2108±.0019 | .1705±.0020 | .1389±.0026 |

style of calculating co-occurrence matrix captures more information from modified pixels comparing to ordinary matrix.

In order to show the performance of our near-optimal double-layered STCs, we test our double-layered STCs along with several popular adaptive steganograpic algorithms, which are the Wavelet Obtained Weights (WOW) algorithm [7], the Spatial Universal Wavelet Relative Distortion (S-UNIWARD) algorithm and the High Low Low (HILL) algorithm.

WOW algorithm uses a bank of directional filters composed of high pass decomposition wavelet filter and low pass decomposition wavelet filter to obtain the so-called directional residuals. By suitably aggregating these directional residuals, WOW constructs the cost reflecting the predictability of the image. UNIWARD algorithm bears similarity to that of WOW but is simpler and suitable for embedding in an arbitrary domain, and for the spatial domain it is the S-UNIWARD. HILL algorithm is designed by using a high-pass filter and two low-pass filters, which are used to locate the less predictable parts in an image and make the low cost values more clustered, respectively.

And the compared objects are the ternary and pentary multi-layered versions of STCs in [3]. To keep the consistency, we set the constraint height in the trellis $h$ as 10. To clearly show the gap between STCs implementation and the theoretical optimal encoding, we also test the simulator which is supposed to achieve the theoretical optimal efficiency.

## 4.2 Experimental Results

To obtain an insight as to which value of $\tau$ should be used in (15), we draw the graph showing $\bar{P}_E$ as a function of $\tau$ when embed payload at 0.4bpp (bits per pixel) with the feature maxSRMd2 and the S-UNIWARD algorithm. Figure 2 shows that when $\tau$ varies from 1.4 to 1.7, the security has the similar performance, which also demonstrates that the value of $\tau$ is larger than the ideal 1. And for uniformity, we set it 1.7. Although $\tau = 1.7$ is much larger than theoretic value $\tau = 1$, the number of modifications in the first layer wouldn't increase 0.7 times, because the solution against $\tau$ of (15) is not as sensitive as one supposes. This facilitates our tuning the parameter $\tau$.

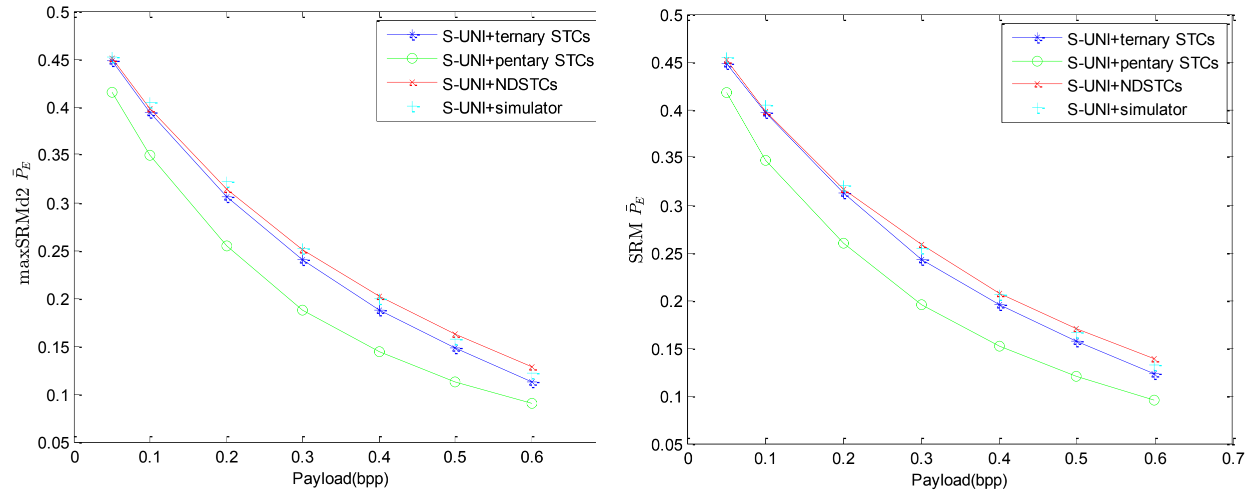


**Figure 4: The detection error $\overline{P}_E$ for ternary STCs, pentary STCs, NDSTCs, simulator with S-UNIWARD in relative payloads. Left for maxSRMd2, right for SRM.**

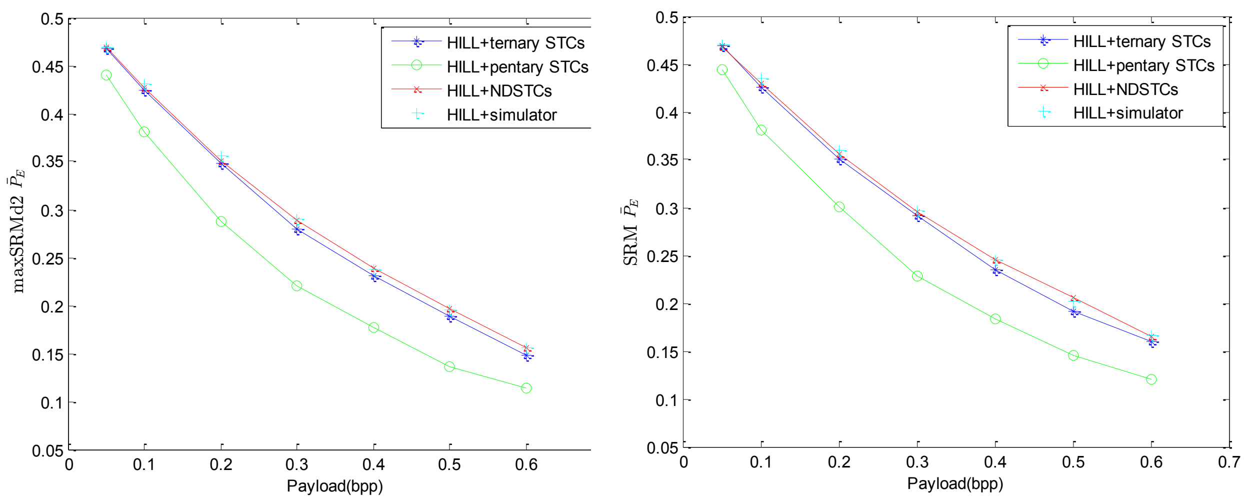


**Figure 5: The detection error $\overline{P}_E$ for ternary STCs, pentary STCs, NDSTCs, simulator with HILL in relative payloads. Left for maxSRMd2, right for SRM.**

In Figure 3, we contrast the embedding changes for NDSTCs, ternary STCs and pentary STCs. The cover image with high textured areas is selected and the adaptive steganograpic algorithm is selected as S-UNIWARD. Data displays that the placements of embedding changes for NDSTCs are more concentrative than the ternary STCs, and meanwhile the number of total modifications is a little less. Although the number of total modifications for the pentary STCs is the least, it brings about too many modifications in the second layer, which violates the principle of reducing the disturbance to high bit plane. Contrasting to the pentary STCs, NDSTCs brings about much less modifications in the second layer and ternary STCs brings about the least.

Figure 4, 5, 6 and Table 2 show the average testing error $\overline{P}_E$ corresponding to the two steganalyzers SRM and maxSRMd2. Figure 4, 5, 6 show $\overline{P}_E$ as a function of the embedding rate. As it shows, comparing to the ternary STCs and pentary STCs, NDSTCs has promotion for all three adaptive steganographic algorithms, which means NDSTCs has a satisfactory generality. Meanwhile it has no difference with the simulator for both maxSRMd2 and SRM, which means it approaches near-optimal security. For the best performance, the detection error of NDSTCs is higher by approximately 1.7% for embedding rate 0.6bpp with the feature maxSRMd2 and the S-UNIWARD algorithm, 1.6% for embedding rate 0.3bpp with the feature SRM and the S-UNIWARD algorithm comparing to the ternary STCs. Since pentary STCs brings about too many modifications in the second layer, the security performance is lower generally. We can say that the security performance of the adaptive steganograpic algorithm sometimes is very close to the theoretical value with our proposed NDSTCs. In this experiment we find that compare to strategy of permute cover with different seed, leaving few $\pm 2$ modifications on wet pixels do not hurt the security if proper cost values are assigned for the second layer. And when we deliberately increase few $\pm 2$ modifications by tuning the parameter $\tau$ slightly large (Figure 7 shows), the security may even better. Its analysis is also addressed in the next section.

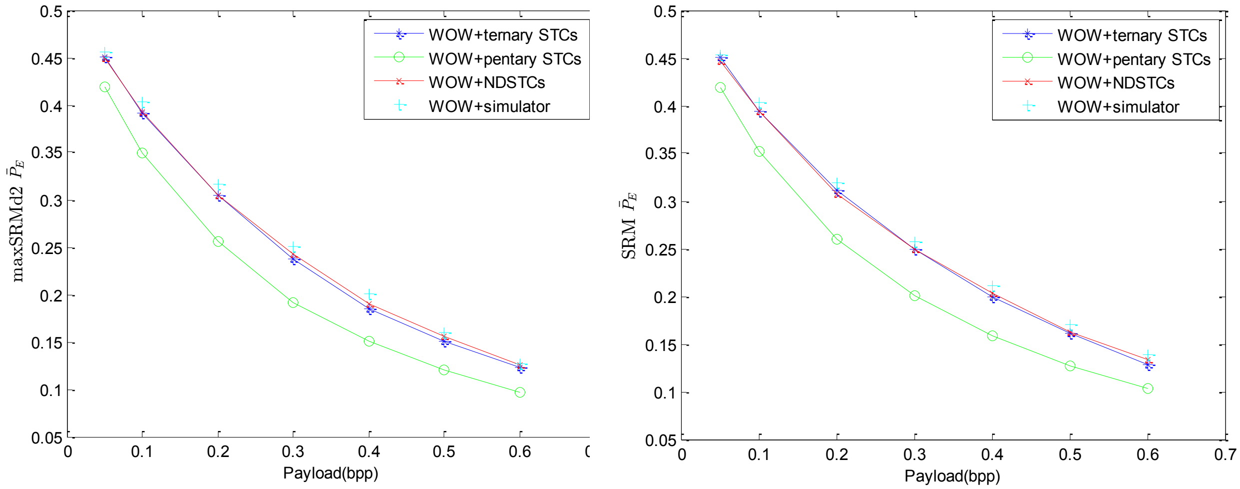


**Figure 6: The detection error $\bar{P}_E$ for ternary STCs, pentary STCs, NDSTCs, simulator with WOW in relative payloads. Left for maxSRMd2, right for SRM.**

## 5. ANALYSIS

Actually, if we want to gain more insight into the theoretic model of modification probability and payload, it is better to consider the number of modifications as a random variable. As is described in 3.1, for binary embedding, the number of modifications is a sum of random variables conforming to binary distribution. Note that there is large number of binary random variables, which amount to the number of pixels in the image. Therefore by using the Lyapunov Central Limit Theorem, we can proof that the number of modifications conforms to normal distribution whose variance and mean are $D(\beta)=\sum_{i,j}\beta_{i,j}-\beta_{i,j}^2$ and $\sum_{i,j}\beta_{i,j}$ respectively. The details of this proof are presented in appendix. The variance describes the deviance of the practical modifications from the theoretical result.

Next we analyze the upper bound of the variance. Suppose we have to carry $|m|$ bits of message, and in this condition we solve the upper bound of variance $D(\beta)$ via following problem:

$$\max_{\beta}\sum_{i,j}\beta_{i,j}-\beta_{i,j}^2$$

$$s.t.\ \ |m|=-\sum_{i,j}\beta_{i,j}\log_2(\beta_{i,j})-(1-\beta_{i,j})\log_2(1-\beta_{i,j}), \tag{17}$$

which can be solved by Lagrange Multiplier Method. After solving this problem, we get a result that when $\beta_{1,1}=\cdots=\beta_{n_1,n_2}$, variance $D(\beta)$ reaches the maximum value. This indicates that if the cost takes a constant value for all pixels, the variance of number of modifications is maximized, but in this case the embedding aims to minimize the number of modifications [13].

In section 3.2, we discussed two cases of cost assignment for the second layer and we suggest the *complexity-first rule*. Actually, when we concentrate on the total modifications, both strategies of case 1 and case 2 in section 3.2 are considered and tested. The testing result shows that *complexity-first rule* in case 2 is better than *reduce change rate* in case 1, and the former is very close to theoretic $\pm1$ embedding simulator. We analyze this phenomenon in following two aspects:

1) Trade-off between $\pm1$ and $\pm2$ modification. There are two measures to eliminate the $\pm2$ modification and keep wet pixels unmodified: (1). Permute cover with other seed and again use STCs encoding. (2). Increase the payload rate of the first layer. While the first measure may takes more time and is unnecessary in our method, we discuss the second one. We find that although the modification intensity of $\pm2$ is larger than that of $\pm1$, allowing few $\pm2$ modifications can reduce much more $\pm1$ modifications in our method, and finally the overall disturbance on steganalysis feature is less.
2) Variance and complexity. There are two virtues of using

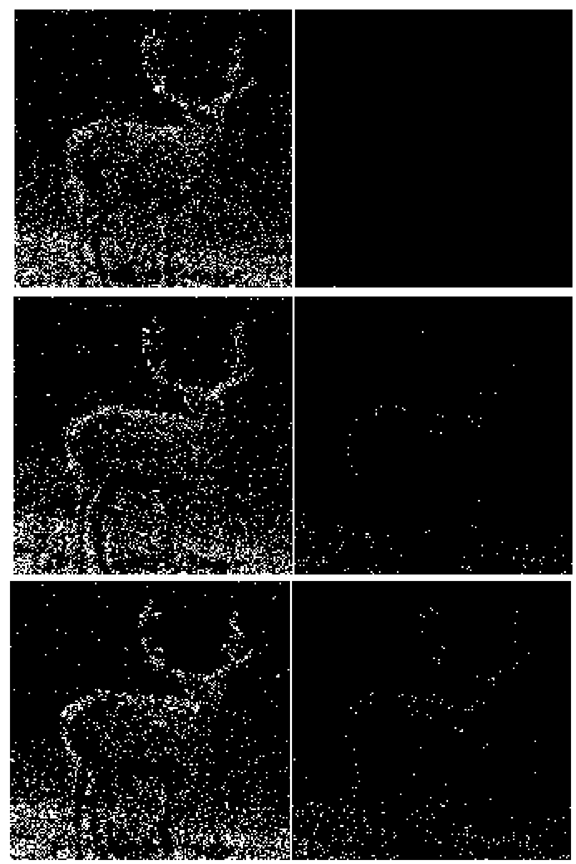

**Figure 7: The embedding modifications of NDSTCs with S-UNIWARD at 0.4 bpp in different value of $\tau$ for the BOSSbase image '8.pgm'. From top to bottom represent the value with 0.8, 1.2 and 1.6. From left to right represent the total modifications and the only modifications in the second layer.**

*complexity-first rule* to assign cost for the second layer. The first one is obvious that more complexity means more security. And the second one, as we discussed above, we can go further to see that *complexity-first rule* cost typically bears less variance of number of modifications, this helps us to more accurately control the number of $\pm 2$ modifications in our method.

And the practical number of modifications by $\pm 2$ is performed as the value of $\tau$. Although we can implement no modifications for the wet pixels through adjusting $\tau$, since this situation means more modifications in total, we don't employ it. We can combine this with Figure 2 and Figure 7. When the value of $\tau$ is small, the total modifications by $\pm 1$ will increase, which has not an ideal security performance. But when the value of $\tau$ is quite large, modifications by $\pm 2$ will increase, which violates the principle of reducing the disturbance to high bit plane and the security performance behaves going down. In our experiment, setting $\tau$ as 1.7 is a trade-off about the two situations.

# 6. CONCLUSION AND FUTURE WORK

In this paper, we propose a new kind of near-optimal double-layered STCs which can be combined with adaptive steganographic algorithm to minimize the additive distortion in spatial domain. We precisely divide the payload into two parts according to the changing probabilities of two layers. Comparing to the restriction of $\pm 1$ modifications, few pixels modified by $\pm 2$ is allowed here. With this scheme, the embedding modifications become more concentrative and the number of them becomes less. The merit of this framework is demonstrated by experiments, which utilizes the state-of-the-art feature-based steganalyzers and the result shows the proposed scheme is a general and secure method which is close to the theoretical simulator. With the performance we can draw a conclusion that it is desirable to allow few $\pm 2$ modifications and sophisticatedly balance the payload of two layers to achieve better security for spatial steganography.

There are some directions we can consider about this work in the future. First, as a general method, it may be extended to the JPEG domain with the same improvement and it may be combined with the strategies of synchronizing the selection channel [15] and clustering modification directions [16] to improve the security together. Second, the security, the number of the only modifications in each layer may be further decreased through preprocessing the costs and as a result the security can be further improved. Furthermore, we fix a value of $\tau$ for each batch of the images here. Maybe we can dynamically turn it according to different image, which is more realistic.

# 7. APPENDIX

In this appendix, we demonstrate that when we convert the cost to the probability, the number of modifications based on probability obeys the normal distribution automatically. Obviously the probability is independent non-identically distributed, so the Liapunov Central Limit Theorem is employed here.

According to Liapunov Central Limit Theorem, for a sequence of independent random variables $\{X_1,\cdots,X_n\}$, $\sum_i X_i$ conforms to normal distribution, when $\exists\delta>0$ the following conditions are satisfied:

$$E\left(|X_i-u_i|^{2+\delta}\right)<\infty,\quad \forall\, i, \tag{18}$$

and

$$\lim_{n\to\infty}\frac{1}{B_n^{2+\delta}}\sum_{i=1}^{n}E\left\{|X_i-u_i|^{2+\delta}\right\}=0, \tag{19}$$

where $u_i$ is the expectation of variable $X_i$ and $B_n^2=\sum_{i=1}^{n}\delta_i^2$ is the sum of the total variances.

We assume we have got the cost and the corresponding change probability. Now we define the change of a pixel as a random variable $c_i$, and it conforms to binary distribution:

$$c_i\sim\begin{cases}1 & P(c_i)=\beta_i\\ 0 & P(c_i)=1-\beta_i\end{cases}. \tag{20}$$

With equation (20), we can get $E(c_i)=\beta_i$, $D(c_i)=\beta_i(1-\beta_i)$. Substitute the variants in equation (18), and let $\delta=1$, we have:

$$\begin{aligned}E(|c_i-E(c_i)|^{2+1})&=\beta_i(1-\beta_i)^3+{\beta_i}^3(1-\beta_i)\\&=\beta_i(1-\beta_i)\left[{\beta_i}^2+(1-\beta_i)^2\right]\\&\le\beta_i(1-\beta_i).\end{aligned} \tag{21}$$

We assume $c_{n+1},c_{n+2},\cdots,c_\infty$ are independent binary random variables with non-identically distributed mean and variance, and we obtain:

$$\begin{aligned}\frac{1}{B_n^{2+1}}\sum_{i=1}^{n}E\{|c_i-\beta_i|^{2+1}\}&\le\frac{\sum_{i=1}^{n}\beta_i(1-\beta_i)}{\left[\sum_{i=1}^{n}\beta_i(1-\beta_i)\right]^{\frac{3}{2}}}\\&=\frac{1}{\left[\sum_{i=1}^{n}\beta_i(1-\beta_i)\right]^{\frac{1}{2}}}\to 0,\ (n\to\infty).\end{aligned} \tag{22}$$

So $\{c_1,\cdots,c_i\}$ satisfy the Lyapunov Central Limit Theorem, and its final form can be expressed as

$$\lim_{n\to\infty}P\left(\frac{1}{B_n}\sum_{i=1}^{n}(c_i-\beta_i)\le x\right)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x}e^{-\frac{t^2}{2}}\mathrm{d}t,\forall \mathrm{x}\in\mathrm{R}, \tag{23}$$

which means $\sum_{i=1}^{n}c_i\sim N\left(\sum_{i=1}^{n}\beta_i,B_n^2\right)$ as $n\to\infty$.


# 8. ACKNOWLEDGMENTS

This work was supported by the NSFC under 61303259 and U1536105, the Strategic Priority Research Program of Chinese Academy of Sciences (CAS) under XDA06030600, and the Key Project of Institute of Information Engineering, CAS, under Y5Z0131201.


# 9. REFERENCES


[1] T. Filler, T. Pevný, and P. Bas. BOSS (Break Our Steganography System). http://www.agents.cz/boss, July 2010.

[2] J. Kodovský, J. Fridrich, and V. Holub. Ensemble classifiers for steganalysis of digital media. In *IEEE Transactions on Information Forensics and Security*, 7(2):432–444, 2012.

[3] T. Filler, J. Judas, and J. Fridrich. Minimizing additive distortion in steganography using syndrome-trellis codes. In *IEEE Transactions on Information Forensics and Security*, 6(3):920–935, September 2011.

[4] T. Denemark, V. Sedighi, V. Holub, R. Cogranne, and J. Fridrich. Selection-channel-aware rich model for steganalysis of digital images. In *IEEE International Workshop on Information Forensics and Security*, Atlanta, GA, December 3–5, 2014.

[5] T. Filler and J. Fridrich. Minimizing additive distortion functions with non-binary embedding operation in steganography. In *IEEE International Workshop on Information Forensics and Security*, Seattle, WA, pages 1-6, 2010.

[6] J. Fridrich and J. Kodovský. Rich models for steganalysis of digital images. In *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, June 2011.

[7] V. Holub and J. Fridrich. Designing steganographic distortion using directional filters. In *Proceedings of IEEE Workshop on Information Forensic and Security*, pp. 234–239, 2012.

[8] V. Holub, J. Fridrich, and T. Denemark. Universal distortion design for steganography in an arbitrary domain. In *EURASIP Journal on Information Security, Special Issue on Revised Selected Papers of the 1st ACM IH and MMS Workshop*, 2014:1, 2014.

[9] B. Li, M. Wang, and J. Huang. A new cost function for spatial image steganography. In *Proceedings IEEE, International Conference on Image Processing, ICIP*, Paris, France, October 27–30, 2014.

[10] W. Zhang, X. Zhang and S. Wang. A Double Layered "Plus-Minus One" Data Embedding Scheme. In *Signal Processing Letters, IEEE* , vol.14, no.11, pp.848-851, Nov. 2007

[11] T. Filler, J. Judas, and J. Fridrich. Minimizing embedding impact in steganography using trellis-coded quantization. In *Proceedings SPIE, Electronic Imaging*, volume 7541, pages 05–01–05–14, Jan. 17–21, 2010.

[12] J. Fridrich, M. Goljan, and D. Soukal. Wet paper codes with improved embedding efficiency. In *IEEE Transactions on Information Forensics and Security*, 1(1):102–110, 2006.

[13] B. Li, S. Tan, M. Wang and J. Huang. Investigation on Cost Assignment in Spatial Image Steganography. In *IEEE Transactions on Information Forensics and Security*, vol.9, no.8, pp.1264-1277, Aug. 2014.

[14] T. Pevný, T. Filler, and P. Bas. Using high-dimensional image models to perform highly undetectable steganography. *In Information Hiding (Lecture Notes in Computer Science)*, vol. 6387. Berlin, Germany, pringer-Verlag, pp. 161–177, 2010.

[15] T. Denemark, and J. Fridrich. Improving Steganographic Security by Synchronizing the Selection Channel. In *3rd ACM workshop on Information Hiding and Multimedia Security.* pp. 5-14, Portland, Oregon, June 17-19, 2015.

[16] B. Li, M. Wang, X. Li, S. Tan and J. Huang. A Strategy of Clustering Modification Directions in Spatial Image Steganography. In *IEEE Transactions on Information Forensics and Security*, vol.10, no.9, pp.1905-1917, Sept. 2015.

[17] J. Fridrich, M. Goljan, P. Lisonek and D. Soukal. Writing on wet paper. In *IEEE Transactions on Signal Processing*, vol.53, no.10, pp.3923-3935, Oct. 2005.