



Received April 4, 2018, accepted May 16, 2018, date of publication May 28, 2018, date of current version June 20, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2841415

Robust Batch Steganography in Social Networks With Non-Uniform Payload and Data Decomposition

FENGYONG LI¹ , KUI WU², (Senior Member, IEEE), XINPENG ZHANG³ , (Member, IEEE), JIANG YU⁴, JINGSHENG LEI¹, AND MI WEN¹, (Member, IEEE)

¹College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai 200090, China

²Computer Science Department, University of Victoria, Victoria, BC V8W 3P6, Canada

³School of Communication and Information Engineering, Shanghai University, Shanghai 200072, China

⁴School of Information and Computer, Shanghai Business School, Shanghai 200235, China

Corresponding author: Fengyong Li (fyli@shiep.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61602295 and in part by the Natural Science Foundation of Shanghai under Grants 16ZR1413100 and 18ZR1427500.

ABSTRACT Batch steganography refers to a steganography method where a user tries to hide confidential payload within a batch of images from social networks. It is significantly different from the traditional *laboratory steganography* where a user only considers an individual image. To apply batch steganography in social media networks, we are faced with two nontrivial problems: 1) how to assign payload to multiple images? and 2) how to recover the hidden payload if some images are lost during transmission? We tackle the problems by: 1) developing an optimal payload embedding strategy and 2) designing a special type of data decomposition. In the former, an optimal non-uniform payload distribution for multiple images is obtained by iterative feature back replacement. In the later, we employ special matrix operation to expand original data and split them into multiple shares. These shares are then embedded into different covers following the optimal non-uniform payload distribution. Our solution is robust in the sense that the recipient can recover the hidden data even if some images are intercepted or lost during delivery. Comprehensive experimental results show that our method outperforms the state-of-the-art in terms of anti-detectability and robustness.

INDEX TERMS Information hiding, batch steganography, embedding strategy, data decomposition, robustness, social networks.

I. INTRODUCTION

Steganography [1]–[5] aims to embed secret messages into digital media, such as digital images, video or audio files, for covert communication. Traditional *laboratory steganography* [1], [2] mainly focuses on embedding messages in only one image. Clearly, it does not work if the size of secret messages is too big. One natural way is to embed the message in video [6], [7], which consists of multiple image frames and thus has a higher capacity for information hiding. Nevertheless, video steganography faces extra challenges since the perceptual quality of the video should remain the same before and after message embedding [6], [7]. To avoid the problems in video steganography, the user may tap the opportunity in social media networks, such as Flickr [39] and Instagram [40], where people upload and share a batch of images. The user can hide secret messages among a batch

of images in social networks. This way of steganography is referred as *batch steganography*, which was firstly studied in [8] and later developed by [9] and [10].

Batch steganography avoids the many problems in video steganography because the user does not need to maintain video quality. It, however, has to face with a problem like this: what if a batch of images uploaded in social network are lost or intercepted? Actually, this is a nontrivial problem. For example, information deliverers hide some secret messages in natural pictures, which might be from various image acquisition devices. These images containing secret information are finally uploaded to social networks. Unfortunately, as we known, social networks are unreliable due to a lot of noise and warden who might detect and intercept the suspicious images, and then remove the hidden data by many kinds of means. Thus, the assumption that all stego images can be

received correctly by recipient is really questionable. To solve the above problem in *the context of social networks*, it needs to answer two key questions: (1) how to distribute the payload among multiple images, and (2) how to recover the hidden information if some images are intercepted or lost during transmission?

Regarding the first question, several theoretical results have been obtained [8]–[10], [30]–[34]. Some of them indicate that the optimal batch embedding strategy may be from two extremes [8]: concentrating the payload into the fewest number of covers, or, spreading the payload into the covers as more as possible. Nevertheless, these theoretical results may not be useful in our context because they assume that the covers are homogeneous with respect to size and compression factors, which is usually unrealistic in social networks.

Another thread of research concerning the first question is *adaptive steganography* [3], [4], [18], [28], [29], which employs a heuristically-defined distortion function to enforce embedding changes in some areas where they will be least detectable, and then the embedding processing with minimum distortion can be located randomly by a Gibbs distribution [36]. Nevertheless, adaptive steganography is hard, at least inconvenient, to apply in our context for the following reasons: (1) The heuristically-defined distortion function in traditional adaptive steganography mainly focus on single image, but the function definition for a batch of images is not straightforward, perhaps needing a more sophisticated definition. (2) The covers from social networks are usually inhomogeneous due to different sizes or different compression factors, the entire distortion for batch images may be very hard to calculate. (3) Even if a whole distortion function for batch images is given, batch steganography is still hard to work, because the implementation of Gibbs embedding in multiple images is as-yet not available [10].

Regarding the second question, existing batch steganography methods [8]–[10], [30]–[34] offer no good answer, because they do not consider social media networks where images transmission over the Internet may be intercepted or lost.

Facing the aforementioned problems, we make the following novel contributions in the context of batch steganography in social networks:

- We propose a new solution for batch steganography, which can not only optimally dispatch payload among multiple images but also ensure the recipient to recover the hidden messages completely even if some images are lost.
- The proposed new solution searches for the optimal non-uniform payload distribution by iteratively using feature back replacement. It is secure in terms of anti-detectability, since the proposed batch steganography effectively narrows the gap between stego distribution and cover distribution.
- To achieve robust batch steganography, we propose a special matrix operation that is used to divide original data into multiple shares, which are hidden into multiple

covers, respectively. The special way of data decomposition guarantees that the recipient can recover the original data perfectly, even if partial data are intercepted or lost during delivery.

- We perform comprehensive experiments with images from real-world social networks. The experimental results demonstrate that our solution significantly outperforms existing batch steganography methods in terms of anti-detectability and robustness.

The rest of this paper is organized as follows. Section II presents secure payload bound and reviews maximum mean discrepancy (MMD). In Section III, we provide the details of non-uniform embedding strategy and introduce data decomposition. Subsequently, comprehensive experiments are performed to evaluate the performance of proposed scheme. The experimental results and corresponding discussions are presented in Sections IV and V, respectively. Finally, Section VI concludes the paper.

II. MOTIVATION AND RELATED WORKS

A. SECURE PAYLOAD BOUND FOR INDIVIDUAL IMAGE

In most traditional laboratory steganographic techniques, the secret message is distributed evenly into all covers according to their capacity. Detecting hidden data is usually restricted in scenarios where only a single source is considered, i.e., to detect whether or not objects from the same source are cover or stego. However, this processing has a serious pitfall: all covers are assumed to have equal payloads and sensitivity to steganography. Actually, this is infeasible in real-world social networks, because even if the covers have equal relative payloads, they may have different detectability due to diversity content. In other words, each cover has the different secure payload bound. Note that in the context of steganography, secure payload mean that the image embedded with the payload cannot be easily detected as a stego. If the relative payload of one cover is higher than its secure payload bound, its stego version will be easily identified. Figure 1 shows the different secure payload bounds for three JPEG images from the BOSSBase v1.01 [38]. In this experiment, we randomly select 5000 images from this database (excluding these three images) to construct training set (including 5000 images and their stego versions). nsF5 steganographic algorithm and PEV-274 feature set [15] are used. We train total 26 generalization classification models from payloads 0 to 0.25 bits per nonzero AC DCT coefficients (*bpnc*) with step size 0.01 and use these models to identify three images until the testing result is *stego*. In order to have a sufficient comparison, we repeat this experiment 50 times and the results are shown in Figure 1(d). Although the three images have the same attributes (from the same source), the secure payload bounds have a large gap, approximately 0 bpnc, 0.11 bpnc and 0.19 bpnc, respectively.

In traditional steganography, we embed the same payload in different images. In fact, according to the above experiments, some covers cannot be embedded so much information, even we have reasons to believe that some covers, not

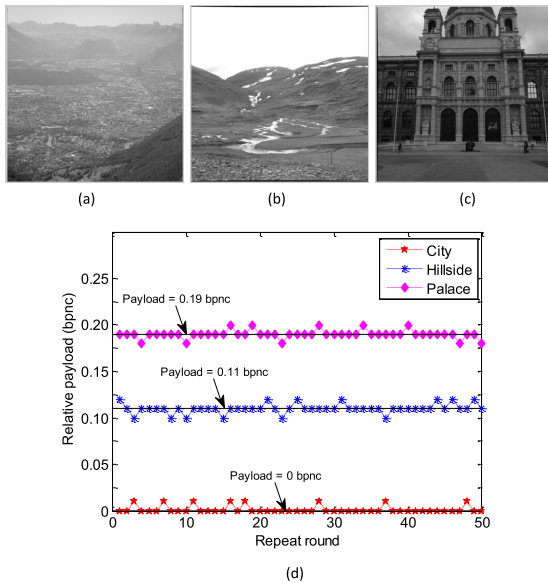


FIGURE 1. Secure payload bound for three images from the BOSSBase v1.01. (a) City. (b) Hillside. (c) Palace. (d) Secure payload bound.

embedded with any messages, may be still classified to *stego*, e.g., Figure 1(a), whereas some stegos may be judged as *cover*, even though they are embedded with more messages, e.g., Figure 1(b) and (c).

After the secure payload bound for individual cover is calculated, one may naturally consider to use aforementioned method to distribute the optimal payload to each cover image, and then get the overall optimal embedding. However, this intuition does not work for batch steganography in social networks. This is because massive training images from the same source are difficult to gather in real-world, and it is infeasible to obtain the optimal secure payload for each cover due to the lack of generalization classification models. In addition, it is unpractical to spread the distortion to the entire set of images through a simple summation or a more sophisticated distortion definition, because the covers from social networks may be inhomogeneous and the optimal embedding is likely to depend on a close relationship between distortion and detectability [10]. Therefore, we need to find a better way for batch steganography in social networks.

B. MAXIMUM MEAN DISCREPANCY

Maximum mean discrepancy (MMD) was first proposed in [11], which has been demonstrated to be useful in distinguishing between cover features and stego features [15]. Given two distributions P and Q with domain \mathfrak{R} , MMD is defined as

$$MMD(P, Q) = \max_f |E_{X \sim P}[f(X)] - E_{X \sim Q}[f(X)]| \quad (1)$$

where f is the mapping $\mathfrak{R} \mapsto \mathbb{R}$ in a Reproducing Kernel Hilbert Space (RKHS) [11].

MMD corresponds to an L_2 distance in a Hilbert space implicitly defined through a positive definite kernel function

$k(x, y): \mathfrak{R}^d \times \mathfrak{R}^d \rightarrow \mathbb{R}$, where x and y are d -dimensional feature vectors. The kernel function k may be of different forms, for example, the linear kernel $k(x, y) = x^T y$ and the Gaussian kernel $k(x, y) = \exp(-\gamma \|x - y\|^2)$, where γ is the inverse kernel width. If we have two feature sets X and Y , each including n samples, $\{x_i\}_{i=1}^n$ and $\{y_i\}_{i=1}^n$, respectively, a sample estimate of the MMD distance can be calculated as:

$$MMD(X, Y) = \frac{2}{n(n-1)} \times \sum_{1 \leq i < j \leq n} k(x_i, x_j) - k(x_i, y_i) - k(x_j, y_j) + k(y_i, y_j). \quad (2)$$

Notably, these two feature sets X and Y need to be normalized, because the raw features may have different scales that can significantly impact the distance measure [15]. Assume that there are l sets, each having m feature vectors. By normalization, each dimensional feature is scaled to have zero mean and unit variance, that is, $\frac{1}{ml} \sum_{i=1}^{ml} \hat{F}_i = 0$ and $\frac{1}{ml} \sum_{i=1}^{ml} \hat{F}_i^2 = 1$, where F is a sequence of raw feature vector and \hat{F} is the normalized version of F .

Actually, MMD has been used not only as a measure of similarity between two feature sets, but also for evaluating the security of steganography schemes [15]. It has been confirmed that a higher MMD value indicates that stego distribution projected in RKHS is farther away from cover distribution. Since cover distribution does not change, when the MMD distance becomes larger, the stego distribution deviates more and the steganographic method would be detected easily. This conclusion can be also demonstrated by the following experiments.

We still use BOSSBase v1.01 image database and randomly select 5000 images to construct training and testing sets. Four steganographic methods, Steghide [12], JP Hide&Seek [13], F5 [1], and nsF5 [2], whose details will be shown in Section IV-B, and eight relative payloads, 0.025, 0.05, 0.075, 0.10, 0.125, 0.150, 0.175, 0.20 bpnc, are used in the experiment. Figure 2 shows the comparisons between MMD detection and SVM with Gaussian kernel (G-SVM) [14] by reporting the probability of error $P_E = (P_{FP} + P_{FN})/2$, where FP (false positives) stands for the proportion of covers that are accused as stegos, while FN (false negatives) represents the proportion of stegos that are considered as covers. Figure 2 left shows the value of $-\log_{10} MMD[X, Y]$ and Figure 2 right shows the P_E of G-SVM for four steganographic algorithms and eight payloads. Although two different ways are used in these two classification methods, the figures show a rather consistent trend, that is, the steganographic methods with a smaller MMD value will have a higher P_E in G-SVM classification and vice versa.

In general, it has been experimentally verified that the detection results of MMD method is surprisingly similar to that of G-SVM classifier [15]. Although MMD method cannot offer any ordering of the difference of probability distribution, it can be used as a replacement of traditional classifier, such as G-SVM classifier, for the steganography detection

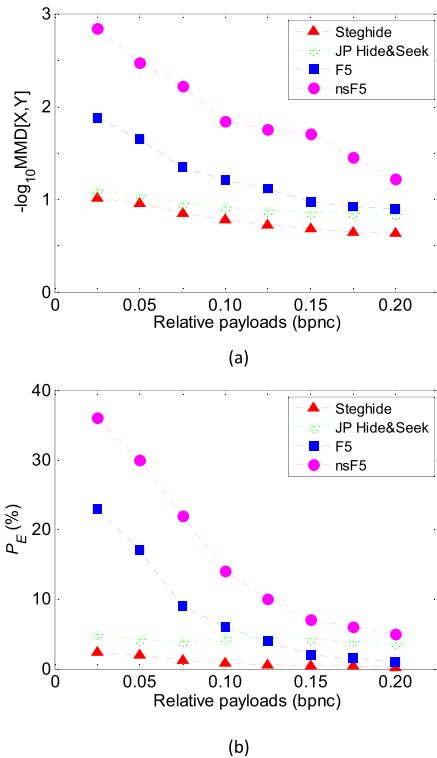


FIGURE 2. MMD detection (a) and G-SVM classification (b) for four steganographic algorithms and eight payloads. The ordinate represents the value of $-\log_{10}MMD[X, Y]$ and the probability of error P_E in two subfigures. Small MMD value means that the steganographic algorithm is more secure.

due to the following reasons. First, the MMD method has a fast convergence rate with respect to the size of samples and the dimensionality of feature space, even for high-dimensional spaces. Second, the computational complexity of MMD is relatively lower than that of G-SVM (theoretically proved with $O(n^2)$ for MMD and $O(n^3)$ for G-SVM [15]).

III. ROBUST BATCH STEGANOGRAPHY

A. THE FRAMEWORK OF PROPOSED SCHEME

The framework of our proposed robust batch steganography scheme is shown in Figure 3. The proposed scheme is mainly comprised of two parts: data embedding and data extraction. In the data embedding stage, given a batch of covers, each cover is presented as a low-dimensional feature vector. Secret messages are firstly decomposed into a number of shares by a multi-ary digital matrix and the total payload can be denoted as the sum of all shares. Furthermore, we employ total payload and feature backward replacement to iteratively calculate an optimal non-uniform secure payload distribution for all covers, and then embed these shares into each cover according to its secure payload, respectively. Finally, these images embedded with secret messages are transmitted over the public network channel. In the data extraction stage, the inverse of multi-ary digital matrix is firstly calculated. As long as enough shares are extracted correctly from the received images, the original messages can be recovered perfectly based on the calculated inverse matrix.

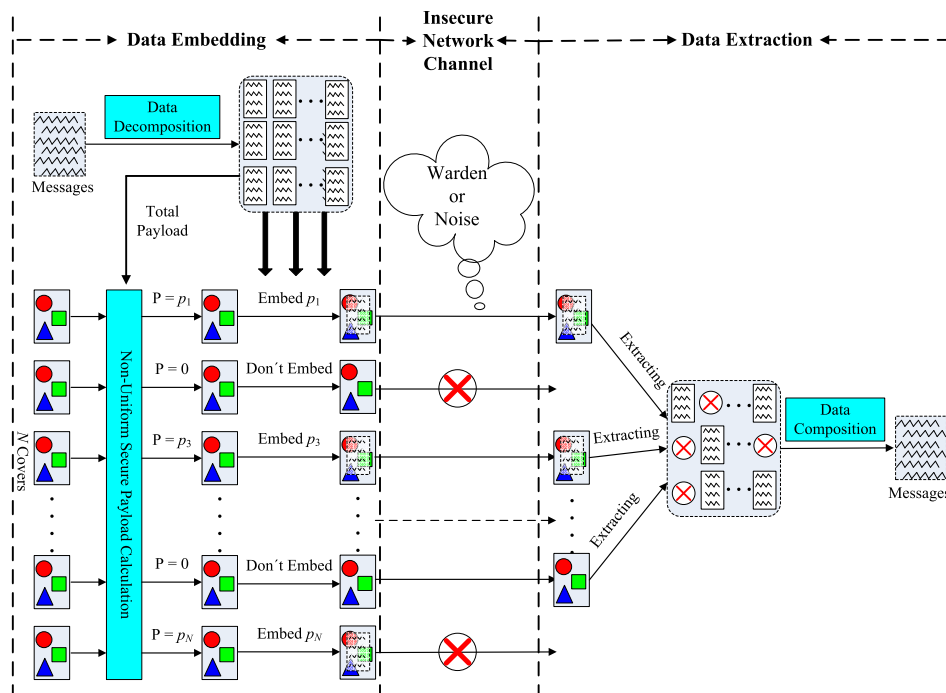


FIGURE 3. Framework of proposed robust batch steganography scheme.

B. NON-UNIFORM PAYLOAD DISTRIBUTION BASED ON FEATURE BACKWARD REPLACEMENT

Following the principle of Section II-A, different covers may have different secure payload bounds, even if they are from the same image source. Traditional batch steganography strategy uses even embedding [10] and may make some stego image easily detectable. It therefore is unsuitable for steganography in social networks. In order to improve the security of information delivery, when multiple covers are used to carry the given messages, we need a secure payload distribution to provide a better guidance for batch steganography.

Given N cover images (X_1, X_2, \dots, X_N) and the total messages of length P , we recall the traditional steganographic strategy: By using a steganographic algorithm, the steganographer would spread the messages into N images, respectively. Thus,

$$P = \sum_{i=1}^N p_i, \quad (3)$$

where (p_1, p_2, \dots, p_N) are the message fragment lengths in each image. Apparently, for even embedding,

$$p_1 = p_2 = \dots = p_N = \frac{P}{N}. \quad (4)$$

According to the principle of Section II-A, there might be an optimal non-uniform payload distribution so that batch steganography is more secure. Considering the discussions in Section II-B, if MMD distance is used as the security measurement, this non-uniform payload distribution, denoted by $(p_1^*, p_2^*, \dots, p_N^*)$, can be uniquely defined by solving the following problem

$$\begin{aligned} & (p_1^*, p_2^*, \dots, p_N^*) \\ &= \arg \min_{F_C \in \mathbb{R}, F_S \in \mathbb{R}} \{MMD(F_C, F_S), MMD(F_C, F_{S^*})\} \quad (5) \\ & \text{subject to } p_1^* + p_2^* + \dots + p_N^* = P, \quad (6) \end{aligned}$$

where F_C is the feature vector sequence from cover images, F_S is the feature vector sequence of stego images by using traditional embedding schemes, and F_{S^*} is the feature set that any feature vector of F_S is replaced. Obviously, it is likely that $p_1^*, p_2^*, \dots, p_N^*$ are different. Therefore, our goal is to find a non-uniform payload distribution $(p_1^*, p_2^*, \dots, p_N^*)$ so that there is a smaller MMD distance between the stego feature set and the cover feature set for batch steganography. Inspired by the sequential pruning of variables [16], we design a new search method, referred as feature backward replacement (FBR).

Given a batch of cover images (X_1, X_2, \dots, X_N) and the message of total size P , we hope to spread the message into N images with a distribution that makes the stego detection difficult. Assume that $F_C = \{f_C^1, f_C^2, \dots, f_C^N\}$ is the feature set of N covers, $F_S = \{f_S^1, f_S^2, \dots, f_S^N\}$ is the feature set of N stegos generated with traditional embedding strategy and with payloads (p_1, p_2, \dots, p_N) , respectively, where $P = \sum_{i=1}^N p_i$. The detailed procedure of searching for the

Algorithm 1 Feature Backward Replacement Search

Input: $F_C = \{f_C^1, f_C^2, \dots, f_C^N\}$, $F_S = \{f_S^1, f_S^2, \dots, f_S^N\}$, (p_1, p_2, \dots, p_N) , N , $\mu \leftarrow 30$
Output: $(p_1^*, p_2^*, \dots, p_N^*)$

- 1 $M \leftarrow MMD(F_C, F_S)$;
- 2 $t \leftarrow 0$;
- 3 **for** $i \leftarrow 1$ **to** N **do**
- 4 **for** $j \leftarrow 1$ **to** μ **do**
- 5 $t \leftarrow j \times 0.01$;
- 6 Embed the message with payload t bpnc to produce stego image X_i^* ;
- 7 Extract the feature vector from X_i^* as $f_{S^*}^i$;
- 8 $f_S^i \leftarrow f_{S^*}^i$;
- 9 **if** $M \geq MMD(F_C, F_{S^*})$ **then**
- 10 $p_i^* \leftarrow t$;
- 11 $M \leftarrow MMD(F_C, F_{S^*})$;
- 12 **end**
- 13 **end**
- 14 **end**
- 15 $s \leftarrow (\sum_{i=1}^N p_i^* - \sum_{i=1}^N p_i)$;
- 16 **if** $s > 0$ **then**
- 17 $p_i^* \leftarrow \min(p_1^*, p_2^*, \dots, p_N^*)$;
- 18 Repeat $s \leftarrow (s - p_i^*)$ and $p_i^* \leftarrow 0$ until $s == 0$;
- 19 **end**
- 20 **if** $s < 0$ **then**
- 21 $(p_1^*, p_2^*, \dots, p_N^*) \leftarrow (p_1, p_2, \dots, p_N)$;
- 22 **end**

optimal payload distribution, $(p_1^*, p_2^*, \dots, p_N^*)$, is illustrated in Algorithm 1.

Remark 1: Feature backward replacement is an iterative searching method. We calculate each payload in a fixed region $[0.01, \alpha]$ with a step size 0.01 .¹ The secure payload distribution $(p_1^*, p_2^*, \dots, p_N^*)$ can be solved when the minimal MMD value is found. In our implementation, we set the parameter α as $\alpha = 0.30$ (corresponding to $\mu = 30$ in Algorithm 1), because in most traditional steganographic schemes, the detection error is pretty low when the payload is more 0.30 bpnc [17], [18], [27]. This conclusion is also demonstrated in Figure 2(b). Thus, we believe that the secure payload bound for most images should be no more than 0.30 bpnc.

Remark 2: FBR is not used standalone but rather is used to improve an existing payload assignment scheme. Given an existing payload assignment scheme, FBR tries to find a more secure scheme, if any, by using interactive search. In this sense, FBR is not to find a globally optimal payload assignment scheme, but to improve existing ones. In other words, given any payload assignment, FBR will return a scheme at least as good as the given one.

¹In fact, the step size can be set to a smaller value. In this work, we set the step size as an empirical value 0.01 to be ease calculation.

C. DATA DECOMPOSITION AND DATA RECOVERY

In batch steganography, the steganographer tries to spread secret messages into multiple cover images, while the receiver needs to receive all stego images to ensure the completeness of messages. Unfortunately, social networks are unreliable in many ways, such as noise and the existence of warden who has the right to remove Internet content in her own discretion. The assumption that the receiver will correctly receive all stego images is really questionable. To improve the robustness of batch steganography and ensure the information completeness, in this section, we build Vandermonde matrix [20], [21] to decompose messages into multiple *shares* and then embed these shares into multiple images.

We can treat a given secret message as a binary stream. We denote a q -ary notational system, where q is an odd prime. The decomposition procedure is as follows.

Step 1: Segment the message into multiple pieces, each of them having L_1 bits. Convert each piece as L_2 q -ary digits and combine all q -ary digits as a sequence, where

$$L_1 = \lfloor L_2 \cdot \log_2 q \rfloor. \quad (7)$$

For example, assuming that $L_1 = 4$ and $L_2 = 2$ and that the 5-ary notational system is used, the binary sequence (1101 0110 1001) can be converted to six 5-ary digits (23 11 14).

Step 2: The steganographer segments the q -ary digit sequence into K small blocks, each of them includes m digits $\{d_{k,1}, d_{k,2}, \dots, d_{k,m}\}$, where $k \in [1, K]$.

Step 3: Build Vandermonde matrix A

$$A = \begin{bmatrix} 1 & 1 & \dots & 1 \\ a_1 & a_2 & \dots & a_n \\ a_1^2 & a_2^2 & \dots & a_n^2 \\ \vdots & \vdots & \dots & \vdots \\ a_1^{m-1} & a_2^{m-1} & \dots & a_n^{m-1} \end{bmatrix} \pmod{q} \quad (8)$$

where $a_1, a_2, \dots, a_n \in [0, q-1]$ are the indices of Vandermonde matrix A and they are different with each other. The values of m, n , and q must satisfy $m \leq n \leq q$.

Step 4: Decompose each secret digit block $\{d_{k,1}, d_{k,2}, \dots, d_{k,m}\}$ into n shares with the following equation.

$$\begin{bmatrix} t_{k,1} & t_{k,2} & t_{k,3} & \dots & t_{k,n} \end{bmatrix} = \begin{bmatrix} d_{k,1} & d_{k,2} & d_{k,3} & \dots & d_{k,m} \end{bmatrix} \cdot A, \quad (9)$$

where A is the Vandermonde matrix and the symbol “ \cdot ” in Equation (9) stands for the multiplication operator in q -ary notational system. $t_{k,1}, t_{k,2}, t_{k,3}, \dots, t_{k,n}$ correspond to their indices a_1, a_2, \dots, a_n , respectively.

According to Equations (8) and (9), m q -ary digits from the original data can be expanded to n q -ary digits. Obviously, the redundancy rate R_e can be represented as

$$R_e = 1 - \frac{m}{n}. \quad (10)$$

Since m and n satisfy $m \leq n$, even if the expanded data $t_{k,1}, t_{k,2}, t_{k,3}, \dots, t_{k,n}$ are lost $n - m$ shares, the original data

$d_{k,1}, d_{k,2}, d_{k,3}, \dots, d_{k,m}$ can still be recovered based on the property of Vandermonde matrix.

Assume that recipient receives m shares $t'_{k,1}, t'_{k,2}, t'_{k,3}, \dots, t'_{k,m}$, which correspond to the indices a'_1, a'_2, \dots, a'_m , respectively. A' is $m \times m$ Vandermonde matrix constructed by the indices a'_1, a'_2, \dots, a'_m (refer to Equation (8)). Thus, the original data $d_{k,1}, d_{k,2}, d_{k,3}, \dots, d_{k,m}$ can be recovered as follows.

$$\begin{bmatrix} d_{k,1} & d_{k,2} & d_{k,3} & \dots & d_{k,m} \end{bmatrix} = \begin{bmatrix} t'_{k,1} & t'_{k,2} & t'_{k,3} & \dots & t'_{k,m} \end{bmatrix} \cdot (A')^{-1} \quad (11)$$

where $(A')^{-1}$ is the inversion matrix of A' in q -ary notational system. The detailed proof about inversion of Vandermonde matrix is presented in **Appendix**.

D. BATCH EMBEDDING AND EXTRACTING PROCEDURE

1) BATCH EMBEDDING

Following the non-uniform payload distribution and data decomposition, we design a robust batch steganographic scheme.

Denote a batch of images X_1, X_2, \dots, X_N and the original secret message P_o . Our proposed scheme delivers the message securely by spreading P_o into N images. Our goal is to ensure that the recipient can get the complete messages. The batch embedding procedure is as follows.

Step 1: According to the data decomposition in Section III-C, construct an $m \times n$ Vandermonde matrix A and decompose the original data into multiple shares $t_{k,1}, t_{k,2}, t_{k,3}, \dots, t_{k,n}$. We attach the corresponding indices a_1, a_2, \dots, a_n to the end of each share. Calculate the total payload and denote it as P .

Step 2: Using the total payload P and following the proposed strategy in Section III-B, search iteratively for a secure non-uniform payload distribution $(p_1^*, p_2^*, \dots, p_N^*)$. Notably, multiple shares may be embedded into one image if the size of shares is too small.²

Step 3: With the non-uniform payload distribution $(p_1^*, p_2^*, \dots, p_N^*)$, embed all shares into the images by using some basic steganographic algorithms.³

2) DATA RECOVERY

When the message is embedded into a batch of images, they are delivered through insecure network channel. According to our proposed data recovery scheme, the recipient can recover the original message completely, even if partial stego images are intercepted or lost during delivery. Assume that the remaining stego images contain m' shares, if $m' \geq m$, the recovery procedure can be implemented successfully as follows.

²We do not consider how the sender informs the recipient of the length of each share, or how many shares correspond to an image, because it could be solved by hiding the information in the image header or by other secret channel

³Our proposed strategy works for most of existing JPEG steganography, including LSB-based steganographic methods and adaptive steganographic methods.

Step 1: Extract m shares $t'_{k,1}, t'_{k,2}, t'_{k,3}, \dots, t'_{k,m}$ from received stego images and meanwhile extract the corresponding indices a'_1, a'_2, \dots, a'_m of these shares.

Step 2: Construct Vandermonde matrix A' by the indices a'_1, a'_2, \dots, a'_m and calculate its inversion matrix A'^{-1} .

Step 3: Use Equation (11) to recover each share until the original message is recovered completely.

Remark 3: We would like to raise the attention to readers that if too many shares are lost, our proposed scheme is not able to recover the original data. According to Equation (10), the redundancy rate R_e is controlled by two parameters m and n of Vandermonde matrix A . The maximum number of shares that are allowed to lost is $n - m$.

IV. EXPERIMENTAL SETUP

In this section, we validate the proposed scheme by simulating a scenario similar to a real-world social network. We embed a message of large size into a batch of social media images. The stego images may be lost, damaged, or intercepted when they are delivering through insecure network channel. Our goal is to ensure that the recipient can get the message without error.

A. IMAGE SOURCE

We carry out our experiments on a simulated social network image source, which contains 22673 JPEG images and is formed by two parts: (1) One part is from two popular social network sites for image sharing (<http://image.baidu.com> and <http://images.google.com>). These websites are popular and include a huge amount of shared pictures, which can be downloaded in batch by some softwares, such as “NeoDownloader” (free version from <http://www.neowise.com>). We have collected the images from 200 users, each of them including 100 images. (2) Another part is from several digital cameras with different noise models, native resolution and quantity, whose specific details can be found in Table 1. These images belong to diverse digital cameras and are very different from laboratory sources, such as BOSSBase [38].

TABLE 1. Camera model with different native resolution and quantity.

Camera Model	Native Resolution	Quantity
Panasonic DMC-FX30	3072 × 2304	407
Nikon D90	4288 × 2848	275
Canon G10	3456 × 2592	338
Nikon D60	3872 × 2592	456
Canon EOS 7D	5184 × 3456	380
Canon EOS 550D	3456 × 2304	342
Sony DSC-TX10	4608 × 2592	475

In order to avoid the influence of different quantization matrices for steganalysis features, these experimental images are firstly converted to 8-bit grayscale, and then central-cropped to the approximately size of 1024 × 1024. Finally, they are JPEG-compressed with standard quantization tables corresponding to quality factors 85. All personal information is removed due to privacy.

B. STEGANOGRAPHIC METHODS

In our experiments, we test four popular steganographic algorithms: StegHide [12], JP Hide&Seek [13], F5 [1], and nsF5 (non-shrinkage F5) [2], [37]. Each algorithm is briefly introduced below.

StegHide. StegHide tries to preserve first-order statistics by a graph-theoretic approach. Since a large message header is embedded with information, it is easily detectable even when a zero-length message is transmitted.

JP Hide&Seek. As a JPEG domain steganographic algorithm, JPHS is developed in 1998 by Latham [13]. Although its C program is available, the mechanism has not been published. The algorithm is not strong, and the created stegos can be easily detected.

F5. The F5 algorithm introduces the so-called *matrix embedding* that substantially decreases the number of embedding changes, especially for smaller payloads. F5 can preserve the overall histogram shape after embedding due to the re-embedding procedure. When the coefficients need to be modified to zero, the same bit is re-embedded on the next coefficient.

nsF5. As an improved version of F5, nsF5 employs Hamming codes to replace matrix embedding so that it is more efficiency and removes the shrinkage effect. In our experiments, we use a simulated algorithm [37] that has the theoretically-optimal efficiency.

C. EMBEDDING STRATEGIES

In our experiment, the following embedding strategies [10] are used to embed messages into a batch of images. We denote the total relative payload for batch steganography as α

$$\alpha = \frac{K}{n}, \quad (12)$$

where K is the total payload size and n is the total number of nonzero AC DCT coefficients of all images in this batch.

Even strategy (ES). The message is split and distributed evenly into all images and their secure capacity (secure payload bound) is not considered.

Linear strategy (LS). All available images are embedded into some messages. The payload size in images changes proportionately with the secure capacity of images.

Greedy strategy (GS). The image with the highest capacity is firstly embedded with the payload size up to the image's maximum capacity. If the message remains, the next image with the highest capacity is selected repeatedly, until the whole message is embedded.

V. EXPERIMENTAL RESULTS

We design a serial of experiments to test the proposed scheme from two aspects, anti-detectability and robustness. The image sources aforementioned are used to imitate network images in real-world. Our goal is to make stego images have a high undetectability and ensure that the recipient is able to recover the message completely. We choose several existing batch steganalysis methods [23]–[25] to test the performance.

A. TEST ANTI-DETECTABILITY FOR DIFFERENT EMBEDDING STRATEGIES

In this section, we show the advantages of our proposed scheme. Ker’s hierarchical clustering scheme (HC) [23], Ker’s local outlier factor detection scheme (LOF) [24], and Li’s ensemble clustering scheme (EC) [25] are used to illustrate the performance, which are recently proposed steganalysis work and are considered in the context of steganographer detection. The main goal of these three schemes is to pin down to one malicious actor (*steganographer*) hiding in multiple innocent ones. Therefore, similar to these schemes, we randomly select some actors and each of them includes 50 images. Then, an actor is randomly chosen as the steganographer who uses three embedding strategies mentioned above to hide messages, while the others act as normal users without using any steganography. The experiments are repeated 50 times. The overall identification accuracy rate (AR) is used to evaluate anti-detectability of different embedding schemes, which is presented as the number of correctly detected steganographers over the selected total number of steganographers, i.e.,

$$AR = \frac{\text{Number of correctly detection}}{\text{Total number of detection}} \times 100\% \quad (13)$$

Based on Remark 2, we should test whether or not FBR can bring a large improvement over any given payload assignment scheme. For this purpose, we apply FBR on the existing schemes, ES, LS, and GS, denoted by ES-FBR, LS-FBR, and GS-FBR, respectively, and compare the performance of the 6 embedded strategies, namely, ES, LS, GS, ES-FBR, LS-FBR, and GS-FBR. Ten different relative payloads, $\alpha \in [0.025, 0.25]$ with step length 0.025 bpnc, are used to mimic the steganographer.

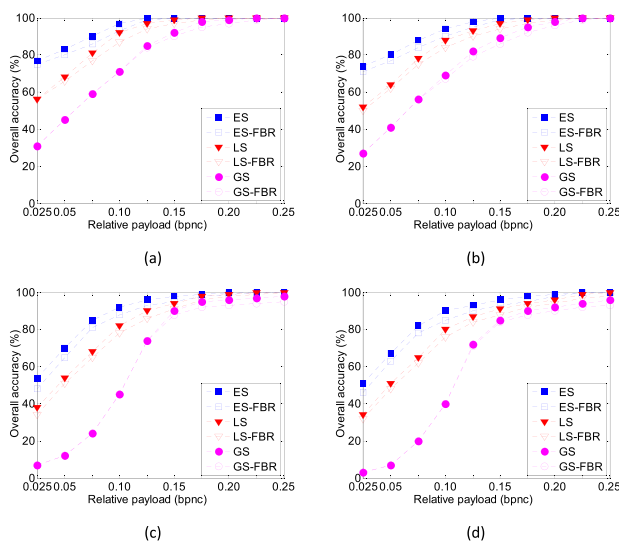


FIGURE 4. Performance comparisons of different embedding strategies for social network source. The steganalytic method is Ker’s hierarchical clustering scheme. Four steganographic algorithms, (a) StegHide method, (b) JP Hide&Seek method, (c) F5 method, and (d) nsF5 method, are used in this experiment. No more than 20 actors are selected to perform clustering in each test.

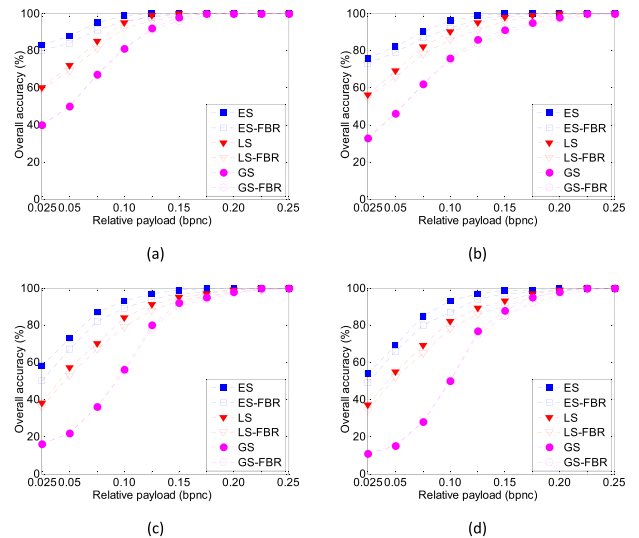


FIGURE 5. Performance comparisons of different embedding strategies for social network source. The steganalytic method is Li’s ensemble clustering scheme. Four steganographic algorithms, (a) StegHide method, (b) JP Hide&Seek method, (c) F5 method, and (d) nsF5 method, are used in this experiment. The cropping size is 192 × 192 and the rounds are fixed $L = 15$. No more than 20 actors are selected to perform clustering in each test.

Figure 4 and Figure 5 show the overall identification results for Ker’s HC scheme and Li’s EC scheme. In these two figures, the x-axis represents relative payload (bpnc), while the y-axis denotes the overall accuracy AR. Notably, for these two steganalytic schemes, since the number of actors maybe have a large impact on the performance with respect to AR measure [24], [25], we select no more than 20 actors in each test. It is easy to observe that for different embedding strategies, when FBR is used, we consistently obtain superior performance with respect to security of steganography, i.e., the detection accuracy becomes lower. No matter which steganographic method is used, the overall accuracies of proposed scheme decreases gradually with the payload increasing for two steganalytic schemes. To be specific, the average reduction is approximately 3% – 5% for even strategy (ES), 2% – 3% for linear strategy (LS) and 1% – 2% for greedy strategy (GS), respectively. In addition, in Figure 5, we find that the overall accuracies are slightly higher than that of Figure 4, the average gain is 2% – 4% for different steganographic methods. This demonstrates that the detectability of EC scheme is superior than that of HC scheme. In fact, this has been conclusively verified in [25].

In addition to HC and EC schemes, we also consider local outlier factor (LOF) to measure the undetectability for different embedding strategies. In fact, it is difficult to precisely detect a steganographer from hundreds of innocent ones. Fortunately, Ker gave another measurement: ranking all the actors and make a short list of the most guilty actors. As such, we can also measure how often the suspicious actor appears in the top n on this list, or, the top $x\%$. Figure 6 shows the performance comparisons between different embedding strategies.

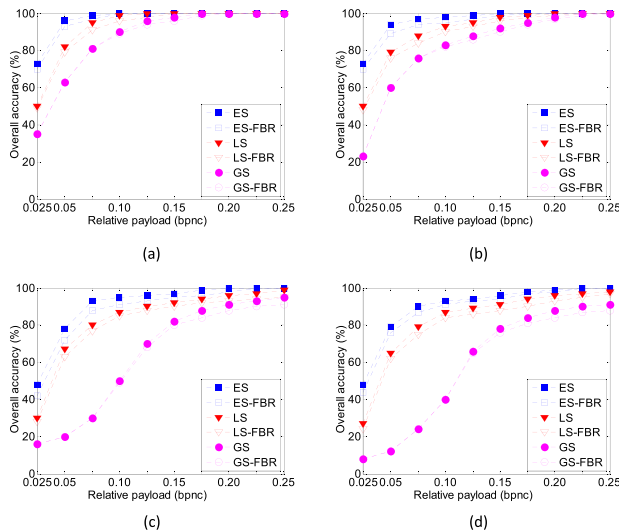


FIGURE 6. Performance comparisons of different embedding strategies for social network source. The steganalytic method is Ker’s local outlier detection scheme. Four steganographic algorithms, (a) StegHide method, (b) JP Hide&Seek method, (c) F5 method, and (d) nsF5 method, are used in this experiment. The LOF parameter of nearest neighbors k is set to 10, and the MMD measure with Gaussian kernel is used. The overall accuracy is tested when the true guilty actor was ranked in the top 10% most suspicious actors.

In this experiment, the LOF parameter of nearest neighbors is set to $k = 10$, and the MMD measure with Gaussian kernel is used. The experiments are repeated 50 times. Each time, we randomly select 50 actors and assume that the suspicious actor can be uncovered if it appears in the top 10% of the most guilty list. We can observe easily that FBR brings a significant advantage with the payload increasing, whatever the steganographic method is used, and the average reduction are more than 3% for ES and LS. Meanwhile, we can also see that the overall accuracies for proposed strategy (GS-FBR) are slightly lower than that of greedy strategy (GS) only with high payload, e.g. more than 0.15 bpnc. This implies that greedy strategy is rather secure so that it is hard to be improved further.

In addition, we also test the advantage of FBR over laboratory sources, such as BOSSBase. Figure 7 shows the performance comparison when the BOSSBase v1.01 source is used. We can observe the same phenomenon as with the social network sources, i.e., FBR consistently improves the existing embedding strategies by lowering the detection accuracy. Comparing the results from the social network source, e.g. Figure 4, and the results in Figure 7, we point out that the images in BOSSBase are 512×512 , which are obviously smaller than that of our social network source, leading to an inferior detection performance. This phenomenon has also been explained by the square root law [26], [35]. Roughly speaking, the square root law means that the secure payload grows asymptotically with the square root of cover size, rather than with the linear of cover size. Therefore, if two images have the same *relative payloads*, the larger the image, the easier the detection.

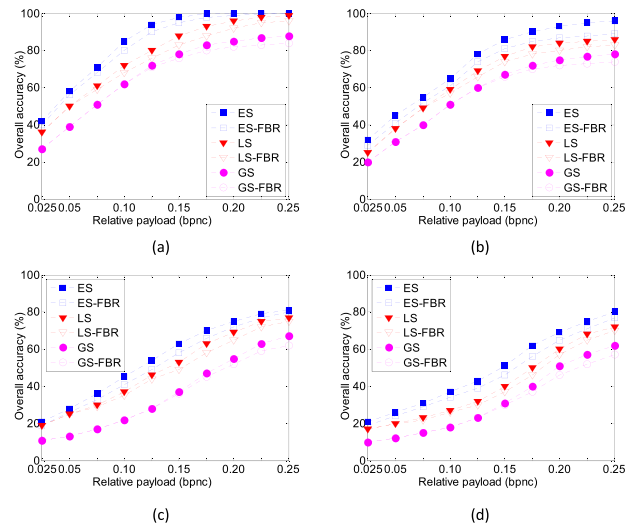


FIGURE 7. Performance comparisons of different embedding strategy for laboratory source BOSSBase. The steganalytic method is Ker’s hierarchical clustering scheme. Four steganographic algorithms, (a) StegHide method, (b) JP Hide&Seek method, (c) F5 method, and (d) nsF5 method, are used in this experiment.

B. ROBUSTNESS ANALYSIS FOR DATA DECOMPOSITION MECHANISM

Since stego images may be lost/removed due to an active warden or poor channel conditions, we design data decomposition mechanism to improve the robustness for batch steganography. In this section, we analyses the correctness of data decomposition and test the robustness with a series of experiments.

Following the property of Vandermonde matrix and Equation (9), m q -ary digits from original data can be expanded to n q -ary shares, the redundancy rate R_e can be represented in Equation (10). In other words, we can recover the original data if no more than $n - m$ out of n shares are lost. As such, we assume that the recipient has received n' shares, where $m \leq n' < n$. He selects m shares $t'_{k,1}, t'_{k,2}, t'_{k,3}, \dots, t'_{k,m}$. These m shares correspond to the indices of Vandermonde matrix, a'_1, a'_2, \dots, a'_m , respectively. According to Equation (8), we can build a Vandermonde matrix A'

$$A' = \begin{bmatrix} 1 & 1 & \dots & 1 \\ a'_1 & a'_2 & \dots & a'_m \\ (a'_1)^2 & (a'_2)^2 & \dots & (a'_m)^2 \\ \vdots & \vdots & \dots & \vdots \\ (a'_1)^{m-1} & (a'_2)^{m-1} & \dots & (a'_m)^{m-1} \end{bmatrix} \text{ mod } q. \tag{14}$$

Since a'_1, a'_2, \dots, a'_m are different with each other and A' is $m \times m$ matrix, it is thus full rank. Apparently, A' has an inverse matrix in q -ary notational system. We denote the inverse matrix as A'^{-1} , which can be obtained as shown in Appendix. Following Equation (11), the original data $d_{k,1}, d_{k,2}, d_{k,3}, \dots, d_{k,m}$ can be recovered correctly.

Actually, we can observe easily that the data recovery is influenced by two parameters of Vandermonde matrix

TABLE 2. The recovery capability for different parameter combinations (m, n) and different redundancy rate R_e .

m	n	q	R_e	Data Lost Ratio		
				20%	50%	80%
5	9	11	44.44%	Yes	No	No
5	19	23	73.68%	Yes	Yes	No
5	31	37	83.87%	Yes	Yes	Yes

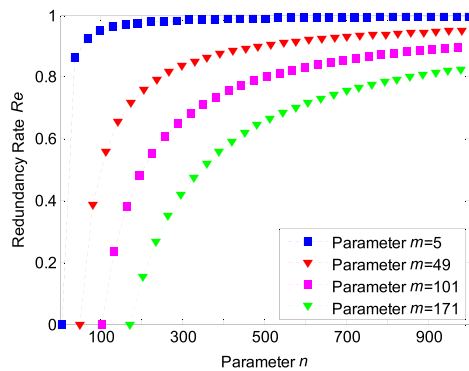


FIGURE 8. The relationship between redundancy rate R_e and two parameters m and n . Four different values, $m = 5, m = 49, m = 101, m = 171$, are tested and satisfy the condition $m \leq n \leq q$.

m and n . If the actual data lost ratio is more than $\frac{n-m}{n}$, the original data will be not recovered, otherwise, we can select the ratio $\frac{m}{n}$ data to build Vandermonde matrix and recover the original data. Theoretically, if the selected parameters m and n are appropriate, the lost ratio can be up very high. Table 2 shows the recovery capability of different parameter combinations when the data lost ratios are 20%, 50% and 80%, respectively. It is observed easily that the larger the difference between parameters m and n , the higher the redundancy rate R_e , the more data the steganographic system is allowed to lose. This conclusion can also be validated by the results in Figure 8.

C. COMPARISON WITH THE STATE OF THE ART

We compare the proposed batch steganographic schemes with three state-of-the-art batch steganography solutions. These existing batch schemes are the combination of a classical steganographic algorithm and different embedding strategies, denoted as ES-Ste, LS-Ste, and GS-Ste, respectively. Meanwhile, since our method uses FBR and data decomposition, we denote our solutions as DD-ES-FBR, DD-LS-FBR, and DD-GS-FBR, respectively. Notably, since the original data is expanded with our method, we may need more covers to carry the expanded data. Thus, for a fair comparison, we introduce a new concept *cover lost ratio*, which is significantly different from the *data lost ratio*, because a cover may contains multiple data shares.

We first analyze the overall performance of six schemes. Table 3 provides the comparisons for anti-detectability and

TABLE 3. Performance analysis for anti-detectability and robustness. Six batch steganographic schemes are used, including three existing methods, ES-Ste, LS-Ste, GS-Ste, and three proposed new methods, DD-ES-FBR, DD-LS-FBR, and DD-GS-FBR.

Embedding Strategy	Batch Schemes	Anti-Detectability	Robustness
ES	ES-Ste	Very Weak	No
	DD-ES-FBR	Weak	Yes
LS	LS-Ste	Weak	No
	DD-LS-FBR	Normal	Yes
GS	GS-Ste	Strong	Somewhat
	DD-GS-FBR	Very Strong	Yes

robustness. No matter which proposed scheme is used, the anti-detectability of our solution is significant superior than that of existing schemes, which is also verified by a series of experiments in Section V-A. This is because, on the basis of existing embedding strategies, our proposed schemes employ FBR to search for a better payload assignment scheme, which leads to a lower overall identifying accuracy due to smaller MMD values.

We further test the robustness of six methods with a series of experiments. In these experiments, nsF5 algorithm is used to form different batch steganographic schemes. We test three payloads 0.025, 0.15, and 0.25, which stand for small payload, normal payload and large payload, respectively. For the proposed data decomposition procedure, two parameter combinations, $m = 5, n = 9, q = 11$ and $m = 5, n = 31, q = 37$, are tested, and their corresponding redundancy rate R_e are 44.44% and 83.87%. We run the experiment 100 times. Each time, we randomly select images from one actor and the average recovery ratio is calculated as the times that the original data can be recovered over the total testing times.

Tables 4 and 5 show the experimental results when the cover lost ratio is set to 30% and 80%, respectively. It can be

TABLE 4. Average recovery ratio for different payloads $\alpha = 0.025, 0.15, 0.25$ bpnc when cover lost ratio is set to 30%. The parameters of Vandermonde matrix is $m = 5, n = 9, q = 11$ ($R_e = 44.44\%$).

Batch Schemes	R_e	Average data lost ratio	Payload α (bpnc)		
			0.025	0.15	0.25
ES-Ste	0%	30.00%	0%	0%	0%
DD-ES-FBR	44.44%	31.25%	100%	100%	100%
LS-Ste	0%	35.41%	0%	0%	0%
DD-LS-FBR	44.44%	32.72%	100%	100%	100%
GS-Ste	0%	36.82%	23%	9%	0%
DD-GS-FBR	44.44%	38.24%	92%	84%	70%

TABLE 5. Average recovery ratio for different payloads $\alpha = 0.025, 0.15, 0.25$ bpnc when cover lost ratio is set to 80%. The parameters of Vandermonde matrix is $m = 5, n = 31, q = 37$ ($R_e = 83.87\%$).

Batch Schemes	R_e	Average data lost ratio	Payload α (bpnc)		
			0.025	0.15	0.25
ES-Ste	0%	80.00%	0%	0%	0%
DD-ES-FBR	83.87%	78.54%	100%	100%	100%
LS-Ste	0%	86.04%	0%	0%	0%
DD-LS-FBR	83.87%	80.88%	94%	80%	74%
GS-Ste	0%	76.28%	5%	0%	0%
DD-GS-FBR	83.87%	88.22%	72%	54%	40%

TABLE 6. Average computation time[‡](second) of individual image for different batch steganographic schemes. The payload is fixed to $\alpha = 0.15$ bpnc and the parameters of Vandermonde matrix are $m = 5, n = 31, q = 37$ ($R_e = 83.87\%$), $m = 11, n = 31, q = 37$ ($R_e = 64.52\%$), and $m = 17, n = 31, q = 37$ ($R_e = 45.16\%$).

Batch Schemes	Parameters (m, n, q)	R_e	Embedding Time (s)	Extracting Time (s)
ES-Ste	-	0%	1.2	1.0
DD-ES-FBR	(5, 31, 37)	83.87%	1.9	2.4
	(11, 31, 37)	64.52%	2.0	3.2
	(17, 31, 37)	45.16%	2.4	4.6
LS-Ste	-	0%	1.3	0.9
DD-LS-FBR	(5, 31, 37)	83.87%	1.9	2.5
	(11, 31, 37)	64.52%	2.2	3.3
	(17, 31, 37)	45.16%	2.4	4.9
GS-Ste	-	0%	1.7	1.3
DD-GS-FBR	(5, 31, 37)	83.87%	2.1	2.8
	(11, 31, 37)	64.52%	2.5	3.4
	(17, 31, 37)	45.16%	2.7	5.0

[‡] LENOVO machine with 8GB RAM and Intel I7 Four-Core 3.40 GHz

seen that existing batch schemes cannot recover the original data, as long as some data is lost. In contrast, our proposed schemes can successfully recover all the data, as long as the data lost ratio is no more than R_e . Moreover, we can observe that when the greedy embedding strategy is used, the existing GS-Ste scheme exhibits certain robustness at a lower payload, e.g., 0.025 bpnc in Table 4 and 0.025 bpnc in Table 5. This interesting phenomenon can be explained as follows. With the greedy embedding strategy, the image with the highest capacity is firstly used to embed part of the message up to the image’s maximum capacity, and if the message remains, the next image with the highest capacity is selected. The process repeats until the whole message has been embedded. This procedure implies that the message will be embedded in a few images with higher capacity. When this batch of images are transmitted over the insecure network channel, as long as those images containing messages are obtained by the recipient, the original data will be recovered completely.

We implement a serial of experiments to further show the average computation complexity of individual image for different embedding schemes. Since proposed schemes use data decomposition mechanism, the average data embedding and extracting time will increase accordingly. Table 6 shows the corresponding testing results. In this experiment, we use the same messages and fix the payload as $\alpha = 0.15$ bpnc. This implies that the different number of image may be involved for different embedding schemes. Moreover, to test the computation complexity of extracting stage, we fix the parameters of Vandermonde matrix ($n = 31, q = 37$) and then make another parameter m change in $m = 5, 11, 17$. As can be seen in this table, the average computation complexity of proposed schemes is significant higher than that of three other methods. To be specific, the average embedding time increases with approximately 0.5-1.5 seconds, while the addition is approximately 1.5-3.5 seconds for the average extracting time. We can explain this phenomenon as follows. Since Vandermonde matrix is controlled by the parameters

m, n and q , when n and q are fixed, the parameter m will determine the sizes of Vandermonde matrix and corresponding inversion matrix (with size $m \times m$). With m increasing, the complexity for data decomposition procedure becomes high slightly. In contrast, the computation complexity of inversion matrix increases significantly so that the data recovery procedure becomes substantial time-consuming.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a new solution for batch steganography in social networks. When steganographer employs a batch of images to deliver hidden information, our proposed scheme can effectively find a payload distribution among all images and ensure that the recipient can recover the original data successfully, even if some stego images are lost during delivery. Given any payload distribution scheme, we firstly use the feature back replacement (FBR) to iteratively search for a more secure, non-uniform payload distribution scheme. Then, we expand the original data and decompose it into multiple data shares, which are respectively embedded into each cover following the payload distribution obtained in the first step. As long as the data lost ratio does not exceed a certain limitation, the original data can be recovered perfectly. We performed extensive experiments using images collected from real-world social networks. Although the results show that our proposed schemes outperform existing batch steganography strategies in terms of anti-detectability and robustness, we should note that they are a bit time-consuming. Overall, we believe that proposed schemes provide a better guidance to non-expert for social network steganography.

Finally, we point out that FBR aims at improving a given payload distribution scheme. The obtained results may still not be globally optimal. In this sense, there may be room for further improvement. In addition, if the redundancy rate R_e is set too high, the Vandermonde matrix will become rather large and the computational complexity of calculating its inverse matrix would be high. The above two issues are left as our future work.

APPENDIX INVERSION OF THE VANDERMONDE MATRIX IN q -ARY NOTATIONAL SYSTEM

The inversion problem of Vandermonde matrix has been studied by [20] and [22]. The results, however, cannot be directly applied to our case because we set the matrix operations in the q -ary notational system.

Assume that there is a Vandermonde matrix A with size $n \times n$

$$A = \begin{bmatrix} 1 & 1 & \dots & 1 \\ a_1 & a_2 & \dots & a_n \\ a_1^2 & a_2^2 & \dots & a_n^2 \\ \vdots & \vdots & \dots & \vdots \\ a_1^{n-1} & a_2^{n-1} & \dots & a_n^{n-1} \end{bmatrix} \text{ mod } q \quad (15)$$

where $a_i \in [0, q - 1]$. Its determinant can be denoted by

$$\det(A) = V_n(a_1, a_2, \dots, a_n) \pmod q \quad (16)$$

According to the basic matrix operation, we know that

$$V_n(a_1, a_2, \dots, a_n) = \prod_{1 \leq j \leq i \leq n} (a_i - a_j) \pmod q \quad (17)$$

Assume that A_{ij} is the corresponding algebraic complement of element a_j^{i-1} in Vandermonde matrix A , $1 \leq j \leq n$ and $1 \leq i \leq n$. Then, the inversion of the element (i, j) can be represented by

$$A^{-1}(i, j) = \frac{A_{ij}}{V_n(a_1, a_2, \dots, a_n)} \pmod q \quad (18)$$

We first derive $V_n(a_1, a_2, \dots, a_n)$. Consider the determinant for $1 \leq k \leq n - 1$

$$V_n^k(a_1, a_2, \dots, a_n) = \begin{vmatrix} 1 & 1 & \dots & 1 \\ a_1 & a_2 & \dots & a_n \\ \vdots & \vdots & \dots & \vdots \\ a_1^{k-1} & a_2^{k-1} & \dots & a_n^{k-1} \\ a_1^{k+1} & a_2^{k+1} & \dots & a_n^{k+1} \\ \vdots & \vdots & \dots & \vdots \\ a_1^n & a_2^n & \dots & a_n^n \end{vmatrix} \pmod q \quad (19)$$

when $k = 0$, define

$$V_n^0(a_1, a_2, \dots, a_n) = \begin{vmatrix} a_1 & a_2 & \dots & a_n \\ a_1^2 & a_2^2 & \dots & a_n^2 \\ \vdots & \vdots & \dots & \vdots \\ a_1^n & a_2^n & \dots & a_n^n \end{vmatrix} \quad (20)$$

when $k = n$, define

$$V_n^k(a_1, a_2, \dots, a_n) = \begin{vmatrix} 1 & 1 & \dots & 1 \\ a_1 & a_2 & \dots & a_n \\ \vdots & \vdots & \dots & \vdots \\ a_1^{n-1} & a_2^{n-1} & \dots & a_n^{n-1} \end{vmatrix} \quad (21)$$

Obviously, according to Equations (15), (16) and (21),

$$V_n^n(a_1, a_2, \dots, a_n) = V_n(a_1, a_2, \dots, a_n) \quad (22)$$

Furthermore, consider the Vandermonde matrix A with order $n + 1$ in q -ary notational system, its determinant is denoted by

$$V_{n+1}(a_1, \dots, a_n, z) = \begin{vmatrix} 1 & \dots & 1 & 1 \\ a_1 & \dots & a_n & z \\ \vdots & \dots & \vdots & \vdots \\ a_1^n & \dots & a_n^n & z^n \end{vmatrix} \pmod q \quad (23)$$

where $z \in [0, q - 1]$. Then, we prove easily that

$$\begin{aligned} &V_{n+1}(a_1, \dots, a_n, z) \\ &= V_n(a_1, a_2, \dots, a_n) \times \prod_{i=1}^n (z - a_i) \pmod q \quad (24) \end{aligned}$$

Consider the symmetrical polynomials of degree k , which has variables $a_1, a_2, \dots, a_n, 0 \leq k \leq n$

$$\begin{cases} \sigma_0(a_1, a_2, \dots, a_n) = 1 \\ \sigma_1(a_1, a_2, \dots, a_n) = a_1 + a_2 + \dots + a_n \\ \sigma_2(a_1, a_2, \dots, a_n) = a_1a_2 + \dots + a_1a_n + a_2a_3 \\ \quad + \dots + a_2a_n + \dots + a_{n-1}a_n \\ \vdots \\ \sigma_n(a_1, a_2, \dots, a_n) = a_1a_2 \dots a_n \end{cases} \quad (25)$$

then

$$\begin{aligned} &\prod_{i=1}^n (z - a_i) \\ &= \sigma_0(a_1, a_2, \dots, a_n) z^n - \sigma_1(a_1, a_2, \dots, a_n) z^{n-1} \\ &\quad + \dots + (-1)^n \sigma_n(a_1, a_2, \dots, a_n) \end{aligned} \quad (26)$$

On the other hand, we develop the last column of $V_{n+1}(a_1, \dots, a_n, z)$ by applying the Laplace expansion.

$$\begin{aligned} &V_{n+1}(a_1, \dots, a_n, z) \\ &= V_n^n(a_1, a_2, \dots, a_n) z^n - V_n^{n-1}(a_1, a_2, \dots, a_n) z^{n-1} \\ &\quad + \dots + (-1)^n V_n^0(a_1, a_2, \dots, a_n) \end{aligned} \quad (27)$$

Substitute Equation (26) into Equation (24) and then combine Equation (27), we obtain

$$\begin{aligned} &V_n^k(a_1, a_2, \dots, a_n) \\ &= V_n(a_1, a_2, \dots, a_n) \sigma_{n-k}(a_1, a_2, \dots, a_n) \end{aligned} \quad (28)$$

Since A_{ij} is the corresponding algebraic complement of element a_j^{i-1} in Vandermonde matrix A , then

$$A_{ij} = (-1)^{i+j} V_{n-1}^{i-1}(a_1, \dots, a_{j-1}, a_{j+1}, \dots, a_n) \pmod q \quad (29)$$

Substitute Equation (28) into Equation (29)

$$\begin{aligned} &A_{ij} = (-1)^{i+j} V_{n-1}(a_1, \dots, a_{j-1}, a_{j+1}, \dots, a_n) \\ &\quad \cdot \sigma_{n-i}(a_1, \dots, a_{j-1}, a_{j+1}, \dots, a_n) \pmod q \end{aligned} \quad (30)$$

then, taking into account Equations (24) and (30), we obtain

$$\begin{aligned} &\frac{A_{ij}}{V_n(a_1, a_2, \dots, a_n)} \\ &= (-1)^{i+j} \frac{\sigma_{n-i}(a_1, \dots, a_{j-1}, a_{j+1}, \dots, a_n)}{\prod_{k=1}^{j-1} (a_j - a_k) \times \prod_{k=j+1}^n (a_k - a_j)} \pmod q \end{aligned} \quad (31)$$

Consequently, combining Equation (18) and Equation (31), the inversion of the element (i, j) in Vandermonde matrix A can be expressed by the following equation.

$$\begin{aligned} &A^{-1}(i, j) \\ &= \frac{A_{ij}}{V_n(a_1, a_2, \dots, a_n)} \pmod q \end{aligned}$$

$$= (-1)^{i+j} \frac{\sigma_{n-i}(a_1, \dots, a_{j-1}, a_{j+1}, \dots, a_n)}{\prod_{k=1}^{j-1} (a_j - a_k) \times \prod_{k=j+1}^n (a_k - a_j)} \bmod q \quad (32)$$

REFERENCES

- [1] A. Westfeld, "F5—A steganographic algorithm: High capacity despite better steganalysis," in *Proc. 4th Int. Workshop Inf. Hiding*, Pittsburgh, PA, USA, Apr. 2001, pp. 289–302. Accessed: 2017. [Online]. Available: <http://code.google.com/p/f5-steganography/>
- [2] J. Fridrich, T. Pevný, and J. Kodovský, "Statistically undetectable JPEG steganography: Dead ends challenges, and opportunities," in *Proc. 9th ACM Multimedia Secur. Workshop*, Dallas, TX, USA, Sep. 2007, pp. 3–14.
- [3] L. Guo, J. Ni, and Y. Q. Shi, "Uniform embedding for efficient JPEG steganography," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 5, pp. 814–825, May 2014.
- [4] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-adaptive steganography by minimizing statistical detectability," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 2, pp. 221–234, Feb. 2016.
- [5] T. Denemark and J. Fridrich, "Steganography with multiple JPEG images of the same scene," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 10, pp. 2308–2319, Oct. 2017.
- [6] Y. Tew and K. Wong, "An overview of information hiding in H.264/AVC compressed video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 305–319, Feb. 2014.
- [7] M. M. Sadek, A. S. Khalifa, and M. G. M. Mostafa, "Video steganography: A comprehensive review," *Multimedia Tools Appl.*, vol. 74, no. 17, pp. 7063–7094, 2015.
- [8] A. D. Ker, "Batch steganography and pooled steganalysis," in *Proc. Int. Workshop Inf. Hiding*, vol. 4437, 2007, pp. 265–281.
- [9] T. Pevný and I. Nikolaev, "Optimizing pooling function for pooled steganalysis," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, Nov. 2015, pp. 1–6.
- [10] A. D. Ker and T. Pevný, "Batch steganography in the real world," in *Proc. 14th ACM Workshop Multimedia Secur. (MM&Sec)*, 2012, pp. 1–10.
- [11] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Advances in Neural Information Processing Systems*, vol. 19. Cambridge, MA, USA: MIT Press, 2007, pp. 513–520.
- [12] S. Hetzl. (2003). *Steghide Algorithm Version 0.5.1*. Accessed: 2017. [Online]. Available: <http://steghide.sourceforge.net/>
- [13] A. Latham. (1999). *JPHide&Seek Algorithm Version 0.3*. Accessed: 2017. [Online]. Available: <http://linux01.gwdg.de/~7ealatham/stego.html>
- [14] C. Chang and C. Lin. (2001). *LIBSVM: A Library for Support Vector Machines*. Accessed: 2017. [Online]. Available: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- [15] T. Pevný, "Kernel methods in steganalysis," M.S. thesis, State Univ. New York, Binghamton, NY, USA, Apr. 2008.
- [16] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, *Feature Extraction: Foundations and Applications*. New York, NY, USA: Springer-Verlag, 2006, pp. 119–136.
- [17] J. Fridrich and J. Kodovský, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 868–882, Jun. 2012.
- [18] V. Holub, "Content adaptive steganography—Design and detection," M.S. thesis, State Univ. New York, Binghamton, NY, USA, May 2014.
- [19] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.
- [20] V. E. Neagoe, "Inversion of the Van der Monde matrix," *IEEE Signal Process. Lett.*, vol. 3, no. 4, pp. 119–120, Apr. 1996.
- [21] X. Zhang, S. Wang, and W. Zhang, "Efficient steganography based on a data decomposition mechanism," in *Proc. IEEE 3rd Int. Conf. Commun. Netw. China*, Aug. 2008, pp. 1248–1252.
- [22] X.-D. Zhang, *Matrix Analysis and Applications*, (in Chinese). Beijing, China: Tsinghua Univ. Press, 2004, pp. 161–166.
- [23] A. D. Ker and T. Pevný, "A new paradigm for steganalysis via clustering," *Proc. SPIE*, vol. 7880, p. 78800U, Feb. 2011.
- [24] A. D. Ker and T. Pevný, "The steganographer is the outlier: Realistic large-scale steganalysis," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 9, pp. 1424–1435, Sep. 2014.
- [25] F. Li, K. Wu, J. Lei, M. Wen, Z. Bi, and C. Gu, "Steganalysis over large-scale social networks with high-order joint features and clustering ensembles," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 2, pp. 344–357, Feb. 2016.
- [26] A. D. Ker, T. Pevný, J. Kodovský, and J. Fridrich, "The square root law of steganographic capacity," in *Proc. 10th ACM Multimedia Secur. Workshop*, Oxford, U.K., Sep. 2008, pp. 107–116.
- [27] J. Yu, F. Li, H. Cheng, and X. Zhang, "Spatial steganalysis using contrast of residuals," *IEEE Signal Process. Lett.*, vol. 23, no. 7, pp. 989–992, Jul. 2016.
- [28] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Paris, France, Oct. 2014, pp. 4206–4210.
- [29] T. Denemark and J. Fridrich, "Model based steganography with precover," *Electron. Imag.*, vol. 7, pp. 56–66, Jan. 2017.
- [30] H. Sajedi and M. Jamzad, "Adaptive batch steganography considering image embedding capacity," *Opt. Eng.*, vol. 48, no. 8, p. 087002, 2009.
- [31] H.-D. Yuan, "Secret sharing with multi-cover adaptive steganography," *Inf. Sci.*, vol. 254, pp. 197–212, Jan. 2014.
- [32] Z. Zhao, Q. Guan, X. Zhao, H. Yu, and C. Liu, "Embedding strategy for batch adaptive steganography," in *Proc. 15th Int. Workshop Digit. Forensics Watermarking (IWDW)*, Beijing, China, Sep. 2016, pp. 494–505.
- [33] Z. Zhao, Q. Guan, X. Zhao, H. Yu, and C. Liu, "Universal embedding strategy for batch adaptive steganography in both spatial and JPEG domain," *Multimedia Tools Appl.*, pp. 1–21, Aug. 2017, doi: [10.1007/s11042-017-5016-z](https://doi.org/10.1007/s11042-017-5016-z).
- [34] R. Cogranne, V. Sedighi, and J. Fridrich, "Practical strategies for content-adaptive batch steganography and pooled steganalysis," in *Proc. 42nd IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, New Orleans, LA, USA, Mar. 2017, pp. 2122–2126.
- [35] A. D. Ker, "The square root law of steganography: Bringing theory closer to practice," in *Proc. 5th ACM Workshop Inf. Hiding Multimedia Secur.*, Philadelphia, PA, USA, Jun. 2017, pp. 33–44.
- [36] T. Filler and J. Fridrich, "Gibbs construction in steganography," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 705–720, Dec. 2010.
- [37] *DDE Download*. Accessed: Mar. 2017. [Online]. Available: <http://dde.binghamton.edu/download/>
- [38] *BOSSBase*. Accessed: Mar. 2017. [Online]. Available: <http://agents.fel.cvut.cz/boss/>
- [39] *Flickr*. Accessed: Mar. 2017. [Online]. Available: <http://www.flickr.com>
- [40] *Instagram*. Accessed: Mar. 2017. [Online]. Available: <http://www.instagram.com>



FENGYONG LI received the M.S. degree from the School of Information and Engineering, Zhengzhou University, in 2010, and the Ph.D. degree from the School of Communication and Information Engineering, Shanghai University, in 2014. He is currently a Lecturer with the College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai, China. His research interests include multimedia security, steganography, and steganalysis.



KUI WU (S'98–M'02–SM'07) received the B.Sc. and M.Sc. degrees in computer science from Wuhan University, China, in 1990 and 1993, respectively, and the Ph.D. degree in computing science from the University of Alberta, Canada, in 2002. He joined the Department of Computer Science, University of Victoria, Canada, in 2002, where he is currently a Professor. His research interests include cloud computing, network security, mobile and wireless networks, and network performance evaluation.



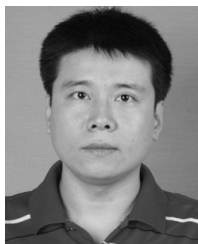
XINPENG ZHANG (M'11) received the B.S. degree in computational mathematics from Jilin University, China, in 1995, and the M.E. and Ph.D. degrees in communication and information system from Shanghai University, China, in 2001 and 2004, respectively.

He has been with the School of Communication and Information Engineering, Shanghai University, since 2004, where he is currently a Professor. His research interests include information hiding, image processing, and digital forensics.



JINGSHENG LEI received the B.S. degree in mathematics from Shanxi Normal University in 1987, and the M.S. and Ph.D. degrees in computer science from Xinjiang University in 2000 and 2003, respectively. His research interests include machine learning, data mining, pattern recognition, and cloud computing. He is the member of the Artificial Intelligence and Pattern Recognition Technical Committee of the China Computer Federation and the Machine Learning

Technical Committee of the Chinese Association of Artificial Intelligence.



JIANG YU received the M.E. degree in electronic circuit and system from the Taiyuan University of Science and Technology, China, in 2011, and the Ph.D. degree in communication and information system from Shanghai University, China, in 2016. Since 2016, he has been with the School of Information and Computer, Shanghai Business School. His research interests include multimedia security, information hiding, steganography, and steganalysis.



MI WEN (M'10) received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2008. She is currently an Associate Professor with the College of Computer Science and Technology, Shanghai University of Electric Power. From 2012 to 2013, she was a Visiting Scholar with the University of Waterloo, Canada. Her research interests include privacy preserving in wireless sensor networks and smart grid.

• • •