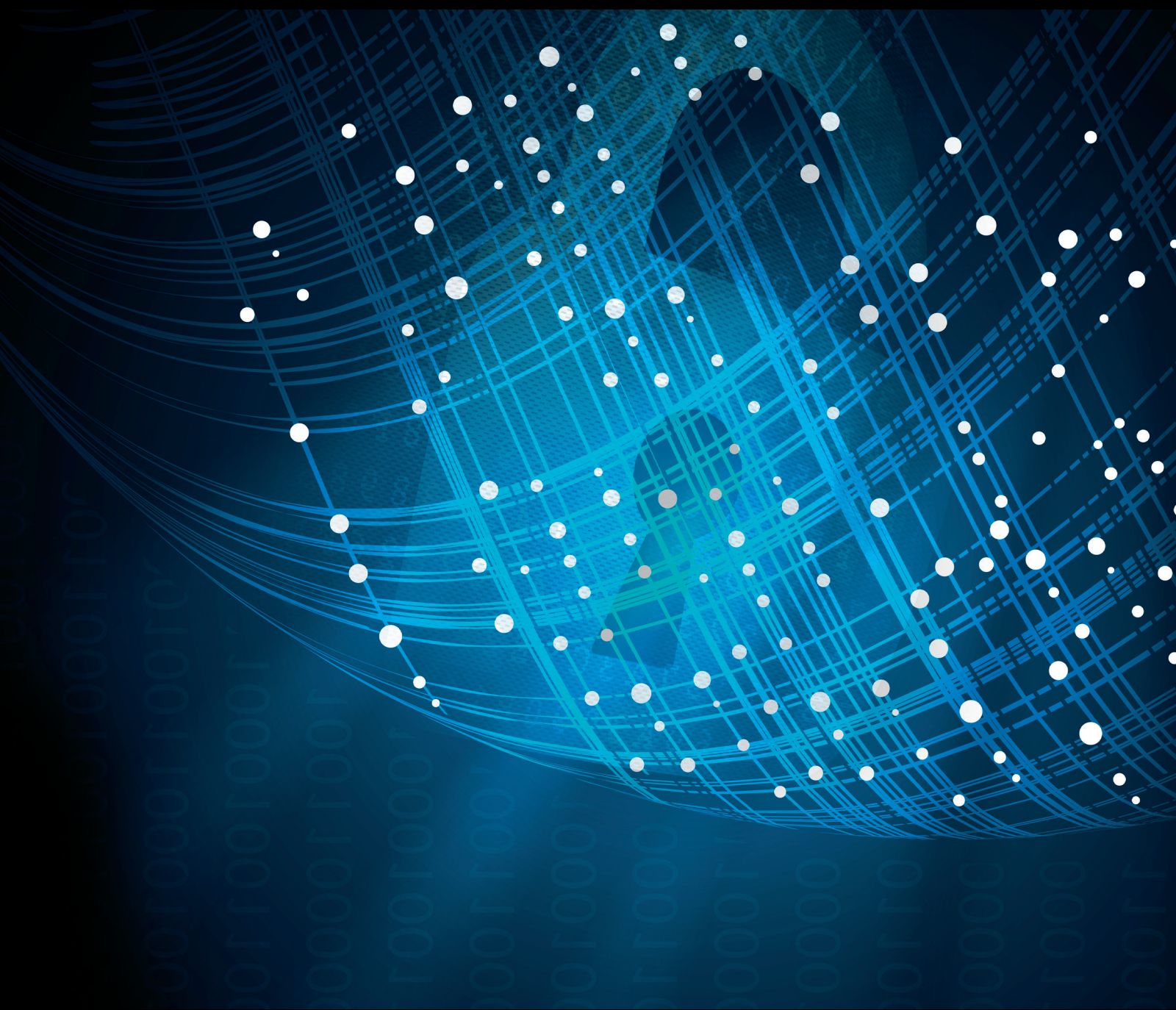
Security and Communication Networks

Exploiting the Security Aspects of Compressive Sampling

Lead Guest Editor: Junxin Chen
Guest Editors: Leo Y. Zhang, Yushu Zhang, Fabio Pareschi, and Yu-Dong Yao



Exploiting the Security Aspects of Compressive Sampling

Security and Communication Networks

Exploiting the Security Aspects of Compressive Sampling

Lead Guest Editor: Junxin Chen
Guest Editors: Leo Y. Zhang, Yushu Zhang, Fabio Pareschi, and Yu-Dong Yao

Editorial Board

Contents

Hindawi
Security and Communication Networks
Volume 2018, Article ID 4740174, 1 page
https://doi.org/10.1155/2018/4740174

Editorial

Exploiting the Security Aspects of Compressive Sampling

Junxin Chen,[1] **Leo Yu Zhang**,[2] **Yushu Zhang**,[2] **Fabio Pareschi**,[3] **and Yu-Dong Yao**[4]

[1]*Sino-Dutch Biomedical and Information Engineering School, Northeastern University, Shenyang, Liaoning, China*
[2]*School of Information Technology, Deakin University, Burwood, VIC, Australia*
[3]*Department of Engineering, University of Ferrara, Ferrara, Italy*
[4]*Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA*

Correspondence should be addressed to Junxin Chen; chenjx@bmie.neu.edu.cn

Received 20 February 2018; Accepted 21 February 2018; Published 24 April 2018

Compressive sampling (CS) has received extensive research attention in the past decade, as it allows sampling at a rate lower than that required by the Nyquist-Shannon sampling theorem. Besides, benefiting from its intrinsic simplicity, convenience and simultaneous encryption, and compression performance, CS also shows great potential in the information security field. This special issue received 13 submissions and published 5 papers which are carefully peer-reviewed by experts in the field.

The published papers of this special issue focus on the application security of compressive sampling. R. Zhang et al. extended the usage of CS for image authentication. Specifically, the primary image is firstly (DWT) transformed and then divided into important part, that is, low frequency part, and unimportant part, that is, high frequency part. For high frequency part, it is encrypted with CS to vacate space for watermark, whereas chaotic encryption is employed to conceal the low frequency phase. The innovation is that Zhang's scheme can realize not only tamper detection and localization but also tamper recovery, in comparison with existing authentication algorithms. J. Wang et al. proposed to use CS for identifying data injection attacks in a nonlinear cyber-physical system and it can also work well in linear systems. The authors conclude that only a small fraction of the observations is supposed to be attacked at a given time instance due to the property of data injection attacks. Hence the error correction problem can be formulated as a sparse optimization primitive and consequently highly relates to CS theory. M. Li et al. proposed to combine reversible data hidden (RDH) with CS and investigated a novel method for image encryption. The key idea is that RDH is applied in CS domain, which introduces a variety of benefits in terms of image sampling, communication, and security. It is demonstrated that the watermark embedding rate is significantly higher than those of other state-of-the-art schemes. Furthermore, the computational complexity of the receiver is also reduced. Two image security papers have also been included in this special issue and are expected to be beneficial for broadening the CS security research. C. Fu et al. developed a chaos-based color image encryption scheme. Different from traditional solutions, a pixel swapping based scrambling approach is developed for permutation, whereas an efficient keystream generation strategy is employed for pixel substitution. Experimental results demonstrate the satisfactory security performance. C.-J. Ouyang et al. considered the security measure in steganography and steganalysis. The proposed security measure evaluates the similarity between two vague sets of cover images and stego images in terms of *n-order* Markov chain for capturing the interpixel correlation and has been shown to have the properties of boundedness, commutativity, and unity.

Presenting these papers together in a special issue, we wish to provide better views for general readers and researchers about the state-of-the-art development of CS security and also expect that this special issue can attract more researchers into the CS security area.

Junxin Chen
Leo Yu Zhang
Yushu Zhang
Fabio Pareschi
Yu-Dong Yao

Hindawi
Security and Communication Networks
Volume 2018, Article ID 1591206, 15 pages
https://doi.org/10.1155/2018/1591206

Research Article

A Novel Image Authentication with Tamper Localization and Self-Recovery in Encrypted Domain Based on Compressive Sensing

Rui Zhang, Di Xiao, and Yanting Chang

College of Computer Science, Chongqing University, Chongqing 400044, China

Correspondence should be addressed to Di Xiao; xiaodi_cqu@hotmail.com

Received 24 July 2017; Accepted 25 September 2017; Published 29 March 2018

Academic Editor: Yu-Dong Yao

This paper proposes a novel tamper detection, localization, and recovery scheme for encrypted images with Discrete Wavelet Transformation (DWT) and Compressive Sensing (CS). The original image is first transformed into DWT domain and divided into important part, that is, low-frequency part, and unimportant part, that is, high-frequency part. For low-frequency part contains the main information of image, traditional chaotic encryption is employed. Then, high-frequency part is encrypted with CS to vacate space for watermark. The scheme takes the processed original image content as watermark, from which the characteristic digest values are generated. Comparing with the existing image authentication algorithms, the proposed scheme can realize not only tamper detection and localization but also tamper recovery. Moreover, tamper recovery is based on block division and the recovery accuracy varies with the contents that are possibly tampered. If either the watermark or low-frequency part is tampered, the recovery accuracy is 100%. The experimental results show that the scheme can not only distinguish the type of tamper and find the tampered blocks but also recover the main information of the original image. With great robustness and security, the scheme can adequately meet the need of secure image transmission under unreliable conditions.

1. Introduction

With the rapid development of data storage and digital process, more and more digital information is transformed and transmitted over the Internet day by day, which brings people a series of security problems as well as convenience. For the openness of network, digital images are vulnerable to attack during the transmission over public network. Receivers often receive tampered images unconsciously. Therefore, unpredictable results occur. Especially for the fields such as governments, military, forensics, and electronic commerce, any slight attack will lead to serious consequences. Accordingly, people pay more and more attention to the protection of privacy information. The researches on digital image security, that is, image encryption, image data hiding, and image authentication, become more important than ever.

Image encryption technique scrambles the pixels of the image and decreases the correlation among the pixels, so that the encrypted image is hard to understand [1]. However, the encrypted image may arouse an attacker's attention to guess the secret behind encryption and seek various ways to crack or break the encrypted content, which heavily threatens the security of the original information. Data hiding technique focuses on embedding some significant information or authentication information into the original cover image based on the redundancy of cover image, which makes it difficult to detect the embedded information.

With the developments of techniques, people hope not only that the data can be delivered to receiver securely but also that the receiver can detect the integrity and authenticity of the received data, which thirsts for image authentication to detect whether the image is tampered and how it is tampered. Conventional authentication techniques belong to integrity authentication, which does not allow any slight change during data transmission and therefore is not suitable for image content authentication. Different from conventional authentication, content-based digital signature and watermarking technique can detect the range of tamper and judge whether

the tamper affects the real content of image. However, most of the existing digital signature and watermarking techniques can only detect the integrity of images or conduct image content authentication with no self-recovery ability or limited self-recovery ability. More and more application scenarios require not only exact tamper detection but also tampered content identification, tamper localization, and self-recovery. Take the transmission and storage of military and medical images as an example. The content owner often encrypts the images first for avoiding information leakage. The data hider embeds secret data in encrypted images. In the receiver side, secret data can be commendably extracted and can be employed to tamper detection, localization and original data recovery. In this way, data can be securely transmitted while the authentication of data integrity and authenticity also can be conducted, which has great practical significance.

In this paper, a novel image authentication with tamper localization and self-recovery for encrypted images is proposed. Firstly, the original image is transformed into Discrete Wavelet Transformation (DWT) domain and divided into important part, that is, low-frequency part, and unimportant part, that is, high-frequency part. Then, different parts are processed differently to realize different goals. Since the low-frequency part contains the main information of image, traditional chaotic encryption is employed in encryption stage so that the low-frequency part can be fully recovered in decryption stage. Then, for the high-frequency part, Compressive Sensing (CS) is used to conduct encryption so as to vacate space for watermark embedding. The scheme takes the processed content of original image as watermark, from which the characteristic digest values are then generated. The watermark is designed mainly for tamper recovery while the digest values are considered as the standard of tamper detection. Tamper recovery is based on block division and the recovery accuracy varies with the contents that are possibly tampered. If either the watermark or low-frequency part is tampered, the recovery accuracy is 100%. If both the watermark and low-frequency part are tampered, the recovery accuracy will decrease while the tampered degree increases. However, some existing pixel prediction techniques can be used to further improve the visual quality of recovered image. The experimental results show that the proposed scheme can not only distinguish the type of tamper and find the tampered image block but also recover the main information of the original image. With great robustness and security, the scheme can adequately meet the need of secure image transportation under unreliable conditions.

The rest of this paper is organized as follows. Section 2 briefly overviews the existing image data hiding and image authentication schemes. Section 3 lists some general knowledge about CS. Section 4 presents the proposed image authentication scheme. Experimental results are demonstrated in Section 5. Finally, we conclude in Section 6.

2. Related Works

Image encryption techniques, from traditional classical encryption algorithms, such as DES and AES, to chaotic novel encryption algorithms and joint encryption algorithms, such as [2], are designed to encrypt text and images. Data hiding techniques usually go with image encryption. The embedding domain can generally be plain domain or encrypted domain. For data hiding in plain domain, including both spatial domain and transform domain, the original image is watermarked first and then encrypted. The classical algorithms in spatial domain can be divided into LSB modification and substitution based algorithms [3], error expansion based algorithms [4], and histogram shifting based algorithms [5]. The classical algorithms in transform domain include discrete cosine transform (DCT) algorithms and discrete wavelet transformation (DWT) algorithms [6–11]. For data hiding in encrypted domain, also including both spatial domain and transform domain, the original image is first encrypted and then watermarked. Data hiding in spatial domain was conducted in [12–16], while authors of [17, 18] hide data in transform domain.

Image authentication techniques can be generally divided into integrity authentication and content authentication. For integrity authentication, any slight change of image is not allowed. For content authentication, the operations that do not influence the content features of image are acceptable. The two methods of image authentication are digital signature and digital watermarking.

So far, a large number of image authentication schemes with digital signature and digital watermarking have been proposed. References [19–21] focused on digital signature, which is sophisticated and has been employed in many applications, especially in electronic commerce. Digital watermarking can be divided into spatial domain schemes and transformation domain schemes. Spatial domain schemes include block-based fragile watermarking [22] and pixel significant bit based watermarking [23]. In [22], the cover image is divided into reversible blocks and irreversible blocks. Reversible blocks are employed to embed the feature information extracted from all the blocks, while irreversible blocks are used to extract the digest information of image. The scheme can accurately locate tampers and recover images with high quality. However, the scheme is quite complicated and cannot resist quantization attack. Moreover, the number of irreversible blocks cannot be more than that of reversible blocks. Otherwise, the scheme cannot realize reversible authentication. In [23] the 7 MSBs' checksum is computed of all the pixels in the original image, which is then embedded into the LSB of each pixel. Though the scheme is of practical value and is easy to implement, the security is extremely low and the scheme is subjected to LSB substitution attack.

Since transformation domain is suitable for the extraction of image features, the authentication methods often rest on wavelet transformation coefficients or cosine transformation coefficients. In [24], a watermark in the form of a visually meaningful binary pattern is used for tamper detection. One watermark bit is embedded into each DCT block by shifting a randomly selected coefficient to have a mapped value. Though the scheme works well in resisting some attacks, there is no tamper recovery capability. In [25], Hasan and Hassan proposed a robust self-embedding watermarking scheme for self-correction and a fragile watermarking scheme for sensitive authentication. The scheme can effectively detect and

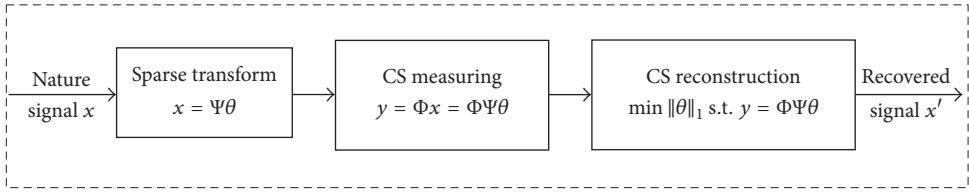


Figure 1: The framework of compressive sensing.

characterize changes and distinguish between malicious and normal manipulations and has autocorrection capabilities of local malicious alterations. In [26], a quantization and DCT based self-embedding fragile watermarking scheme with effective image authentication and restoration quality is proposed. The scheme used a small nonoverlapping block sized 2×2 to improve the accuracy of localization and can effectively remove the blocking artifacts. Unfortunately, the watermark data, which is embedded into three LSBs planes, may be destroyed by some image processing operations. In [27], Liu and Hu designed two watermarks from the low-frequency band of DWT domain and embedded the watermarks into the high-frequency bands. The scheme can resist the mild modifications of digital image and be able to detect and recover the malicious modifications precisely. In [28], the image features are extracted from the lowest-frequency coefficients of each block as the first embedded watermark and the orientation adjustment is then calculate based on the two-level wavelet coefficients in the middle-frequency subbands for image authentication. The scheme uses image feature and logo watermark as two different embedded watermarks and can realize image authentication and recovery of the tampered regions simultaneously. In [29] a semifragile and self-recoverable watermarking algorithm is proposed based on a group quantization and double authentication method. The scheme takes the generated authentication watermarks as information watermarks to reduce the amount of the embedding watermarks, enhances security by randomly permuting coefficients among a group, enhances robustness by embedding the watermarks in the largest coefficient inside a group, and employs the double authentication ring structure to effectively improve localization accuracy.

CS domain is another significant transformation domain, based on which image authentication schemes have sprung up. In [30], CS is employed to process watermark, which strengthens the security of watermark. However, due to the distortion during the process, the receiver cannot extract watermark exactly. In [31], CS is used to process watermarked image, which ensures the security of both watermark and cover image. However, the accuracy of watermark extraction is at risk. Some researchers have proved that hiding data in measurements is of strong robustness [32–37]. In [32, 33], Rachlin et al. showed the security and confidentiality of CS measurements. Without key and heuristic knowledge, attacker cannot infer the content of watermark from the measurements of encrypted image, which means that embedding watermark in CS measurements is feasible. In [34], the sender converts the original image into frequency domain with discrete wavelet transform (DWT), computes the measurements of encrypted image with compressive sensing measuring, and embeds watermark into the measurements. The watermarked encrypted image is then generated with CS reconstruction algorithm. However, only one measurement matrix is employed during the whole process, which is of huge computation and cannot resist large-scale noise attack. Moreover, the original image is needed for watermark extraction. Since the energy distribution of image is uneven and the embedding in energy-concentrated region will result in important information losing and destroying, it is better to embed watermark in energy-dispersed region.

CS based tamper authentication and recovery schemes for encrypted image can allow a certain compression ratio and effectively conduct tamper detection. The schemes can realize tamper content identification, or tamper localization or tamper recovery. However, the accuracy of tamper localization is not high and the above-mentioned three goals cannot be reached at the same time. In view of these insufficiencies, we propose a CS based image authentication scheme for the encrypted image jointly with tamper detection, localization, and recovery. The proposed scheme divides the image into important part and unimportant part and encrypts different parts with different encryption algorithms. For realizing tamper detection, localization, and recovery, the proposed scheme generates characteristic watermark from the original image and embeds the watermark into the compressive sensing measurements of the original image. The watermark is generated from the low-frequency part of DWT with CS. The reconstruction feature of CS is employed for tamper detection and recovery. And then the characteristic values extracted from watermark are considered as the tamper localization standard.

3. Compressive Sensing

In this section, we provide a brief introduction to CS. Compressive sensing, also known as compressive sampling or sparse sampling, is a new signal acquisition technology to capture and represent compressible signals at a rate significantly below the Nyquist rate. The original signals can be exactly or approximately reconstructed with a small number of measurements.

The general framework of compressive sensing, including sampling process in the encoder side and reconstruction process in the decoder side, is shown in Figure 1. Suppose that x is an $n \times 1$ natural signal which itself may or may not sparse in the canonical basis but is sparse or approximately sparse in an appropriate basis Ψ.

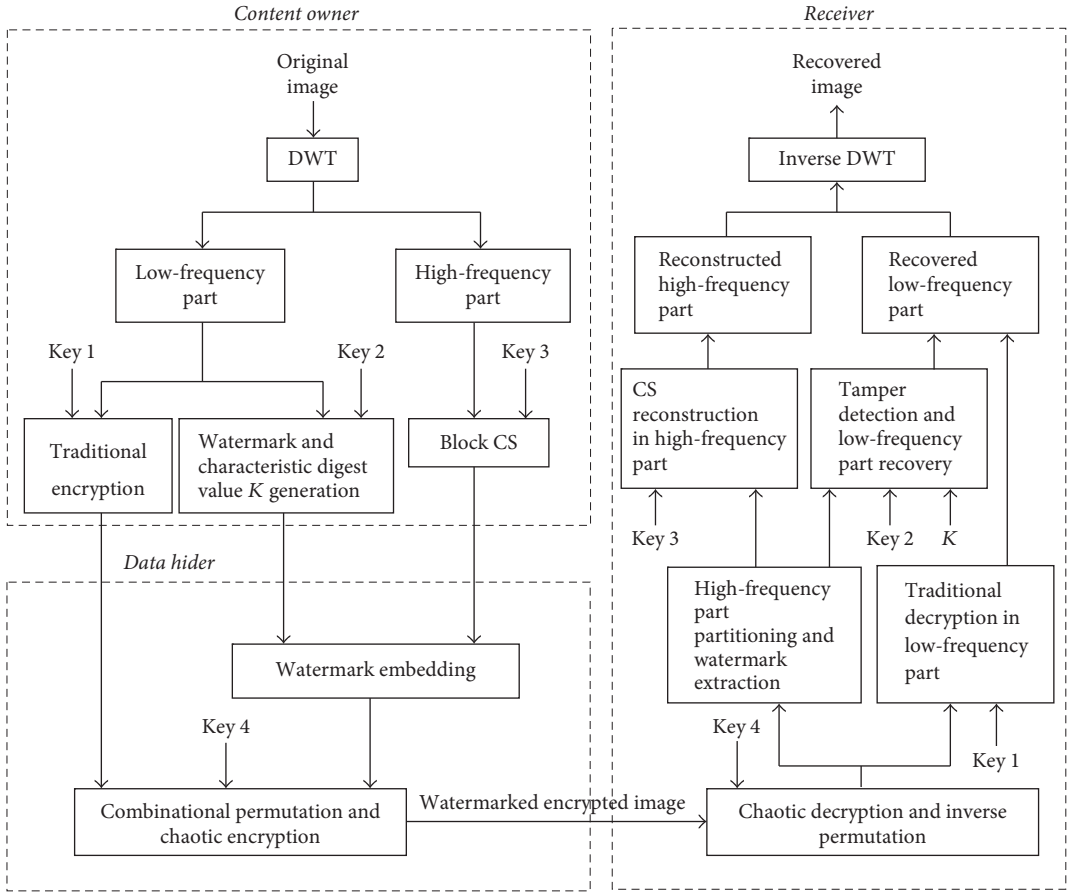


Figure 2: The general framework of the proposed scheme.

$$x = \Psi\theta, \tag{1}$$

where θ is k sparse; that is, there are exactly $k \ll n$ nonzero components. For the sampling process, measurements y can be computed through multiplying an $m \times n$ ($m \ll n$) measurement matrix Φ by x.

$$y = \Phi x, \tag{2}$$

where y, an $m \times 1$ sample vector, contains most useful information of x. Φ satisfies the restricted isometry property (RIP) of a certain order [38]. Then the sparse θ signal can be directly sampled via the following equation:

$$y = \Phi x = \Phi\Psi\theta. \tag{3}$$

For the reconstruction process, the signal x can be reconstructed from measurements y by solving an l_1 minimization problem.

$$\begin{aligned} \widehat{\theta} = \arg\min \quad & \theta_1 \\ \text{s.t.} \quad & y = \Phi\Psi\theta \\ & \widehat{x} = \Psi\widehat{\theta}, \end{aligned} \tag{4}$$

where $\widehat{x}$ is the recovered signal.

To the best of our knowledge, if the entries of matrix Φ are generated from a Gaussian distribution with zero mean and variance, Φ is a RIP matrix with overwhelming probability. In this paper, such a Gaussian distribution is employed to generate compressive sensing matrix. Moreover, the DWT is adopted to make the original signal sparse.

4. The Proposed Scheme

As illustrated in Figure 2, the proposed scheme mainly involves three parties: content owner, data hider, and receiver. The content owner generates watermark and characteristic digest values from the original image and encrypts the original image. When receiving the encrypted image and watermark, the data hider embeds the watermark into the encrypted image and transmits the watermarked encrypted image to the receiver. With relevant keys, the receiver can easily decrypt the watermarked encrypted image and conduct tamper detection, tamper localization, and image recovery.

4.1. Image Encryption and Watermark Embedding. In this stage, the content owner first encrypts the original image and generates the watermark to be embedded. Then the data hider conducts watermark embedding into the encrypted image. The framework is illustrated in Figure 3.

4.1.1. Image Encryption and Watermark Generation. Suppose that the original image is a gray scale image I.

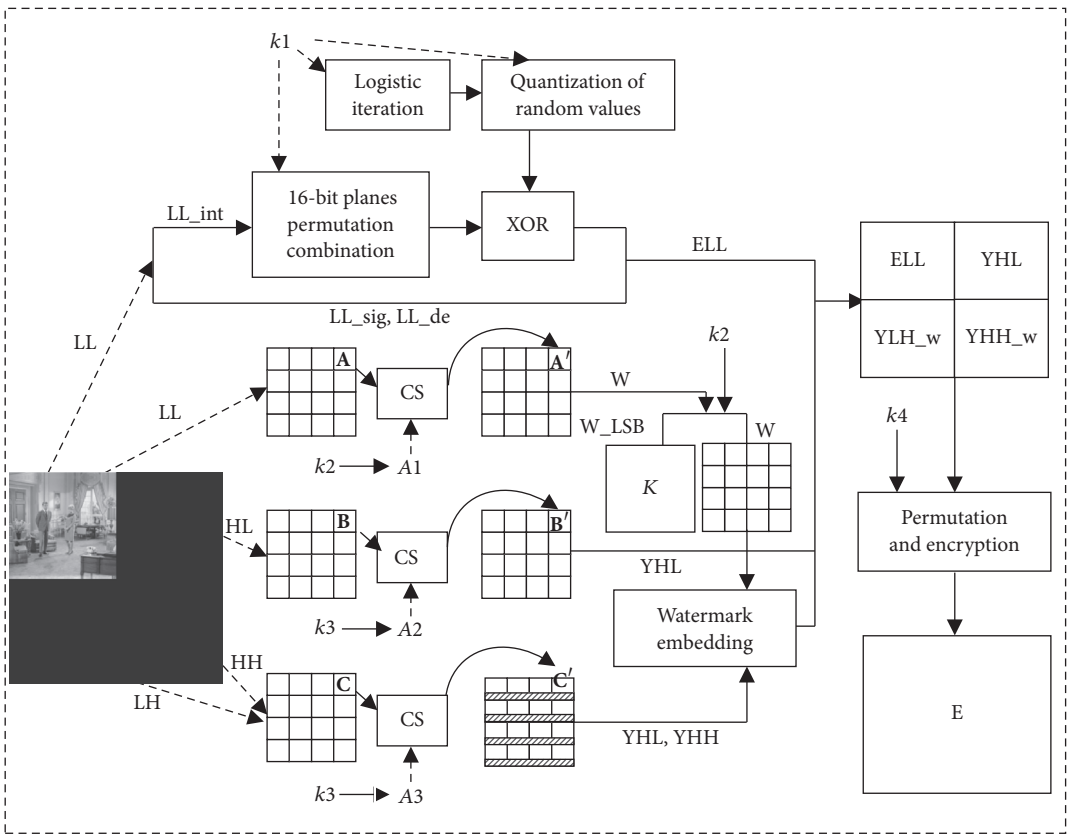


Figure 3: The framework of encryption and watermark embedding.

Step 1. The content owner decomposes the original image I with DWT and gets low-frequency part LL and high-frequency parts HL, LH, and HH. For further processing, the low-frequency part is considered as the important part while the high-frequency parts are deemed as the unimportant parts.

Step 2. There are two operations for important part LL. One is traditional image encryption. The other is watermark generation and characteristic digest value generation with direct block division and compressive sensing.

(A) Traditional Image Encryption with Arnold Scrambling and Logistic Map

(1) Separate out sign matrix LL_sig, absolute integer matrix LL_int, and decimal matrix LL_de from LL. Conduct $t1$ times Arnold scrambling to the 16-bit planes of LL_int, respectively, and then the scrambled bit panes are reassembled. The periodicity of Arnold scrambling $y1$ and the iterations $t1$ are part of the private key $k1$.

(2) According to another part of the private key $k1$, the initial values $(x0, y0)$, and a big integer P, use Logistic map to generate a random sequence with the size as LL_int, multiply it by the big integer P, and then perform modular 65536 operation.

(3) XOR the generated pixels and the random numbers to get the encrypted low-frequency integer matrix ELL_int, which is then reassembled with the sign matrix LL_sig and decimal matrix LL_de to form the encrypted low-frequency part ELL.

(B) Watermark Generation. The watermark generated from LL with block compressive sensing in this proposed scheme is designed for tamper authentication and recovery. Therefore, the size of measurement matrix should be the same as that of block. Watermark is generated as follows.

Divide LL into nonoverlapping blocks with the size of $a \times a$. Generate chaotic measurement matrix $A1$ with the seed private key $x2$. Here a and $x2$ are part of the private key $k2$. For each block, reshape it into one-dimensional vector through Zig-Zag scanning. Then each vector is measured to get the measurement value. All the measurement values are combined to form the measurement watermark matrix W. Since the watermark will be used for low-frequency recovery, the compression ratio is set to 1 for reducing distortion.

(C) Characteristic Digest Value Generation. Characteristic digest value is generated through watermark processing, which will be used for image authentication in the coming stage. Since it is transmitted via secure channel, it can be employed for tamper authentication and localization.

Take the absolute value of W as an integer matrix. Transform each element of the matrix into 16-bit sequence

with 0 and 1, which is then permuted with the private key t. Pick out the LSB plane from the 16-bit planes to form W_LSB. Compress it with run-length encoding. Then it is considered as the characteristic digest value K and transmitted to the receiver side together with other private keys.

It should be noted that since the characteristic digest value is generated through taking a bit plane from the blocks of watermark W which originates from low-frequency part with compressive sensing, the characteristic digest value can only detect whether the watermark or low-frequency part is tampered and then find out the tampered blocks.

Step 3. For the unimportant part, that is, high-frequency parts HL, LH, and HH, different compressive sensing operation will be employed to ensure the reasonability of watermark embedding and the invariance of image size before and after encryption.

(A) For HL. Divide HL into nonoverlapping blocks with the size of $a \times a$. Reshape each block into one-dimensional vector with the same method. Generate a measurement matrix $A2$ with the private key $x3$. Then each block is measured to get a measurement value, which is then transformed and combined as the measurement value matrix YHL with compression ratio of 1.

(B) For LH and HH. Divide LH and HH into nonoverlapping blocks with the size of $a \times a$. Reshape each block into one-dimensional vector with the same method. Generate a measurement matrix $A3$ with the private key $x4$. Then each block is measured to get a measurement value, which is then transformed as the blocked measurement value matrices YLH and YHH with compression ratio of 0.5. The vacant positions are filled up with 0. Here, half of space in LH and HH after compression is vacated for watermark. Moreover, a, $x3$, and $x4$ are part of $k3$.

4.1.2. Watermark Embedding. Watermark is made up of measurement values and will be embedded into the measurement values of high-frequency part. Therefore, it not only makes watermark localization and extraction convenient but also reduces the error rate of watermark extraction. After being embedded, the watermark and other elements in high-frequency part show the same distribution features of encrypted data so that it is difficult to distinguish whether watermark is embedded.

Step 1. Separate each watermark block into two parts, that is, the upper part and the lower part. The size of each part is $a/2 \times a$. Embed these two parts into the corresponding positions of YLH and YHH, respectively, which have been filled up with "0." After watermark embedding, YLH_w and YHH_w are generated and then combined with YHL to form the watermarked high-frequency part.

Step 2. Reassemble the low-frequency part and the high-frequency part. Generate a random sequence using Logistic map with the private key $(x1, y1)$. Combine this sequence and the watermarked encrypted image with XOR operation. Then perform $t2$ times Arnold scrambling to the resultant matrix to get the final watermarked encrypted image E. Here, iteration cycle $y2$ and iteration times $t2$ and $(x1, y1)$ are part of the private key $k4$.

4.2. Watermark Extraction and Image Decryption. In this section, all the operations will be done by the receiver. As illustrated in Figure 4, the process can be divided into four stages, that is, watermark extraction, image decryption, tamper validation, and tamper localization and recovery.

4.2.1. Watermark Extraction. After receiving the watermarked encrypted image, the receiver first decrypts the image and then extracts the watermark from the image. The watermark extraction process is the inverse process of embedding.

Step 1. The receiver first separates $(x1, y1)$ from $k4$. Generate a chaotic random sequence using Logistic map with $(x1, y1)$. Then pick up iteration cycle $y2$ and iteration times $t2$. Perform $y2 - t2$ times Arnold scrambling to the watermarked encrypted image E. Perform XOR operation between this scrambled image and the generated chaotic random sequence and then divide it into low-frequency part ELL_RE and watermarked high-frequency part.

Step 2. Divide the watermarked high-frequency part into three parts, that is, YHL_RE, YLH_w_RE, and YHH_w_RE, which are then further divided into nonoverlapping blocks. From the first block of YLH_w_RE and YHH_w_RE, take the lower part of the corresponding block to get a watermark block and put it into the corresponding block position of the watermark extraction matrix W_RE. When the processing of all the blocks finishes, the watermark is fully extracted. Moreover, the high-frequency part after watermark extraction will change into YLH_RE and YHH_RE.

The extracted watermark is mainly used for tamper verification. Without tamper, a high quality image will be directly reconstructed after decryption. When tamper occurs, the characteristic digest value will be generated and then compared with the transmitted characteristic digest value for tamper localization and recovery.

4.2.2. Image Decryption

(A) Low-Frequency Part Decryption

(1) Separate out the sign matrix ELL_sig_RE, the absolute integer matrix ELL_int_RE, and the decimal matrix ELL_de_RE from ELL_RE. Take the initial value $(x0, y0)$ and the big integer P out of the key $k1$. Generate a random sequence using Logistic map with the same length as ELL_int_RE. Multiply it by the big integer P and then perform modular 65536 operation. Conduct XOR operation between the resultant sequence and the pixels of ELL_int_RE.

(2) Take the periodicity of Arnold scrambling $y1$ and the iterations $t1$ out of the key $k1$. Divide the integer part of the low frequency after XOR operation into 16-bit planes. Perform $y1 - t1$ times Arnold scrambling to

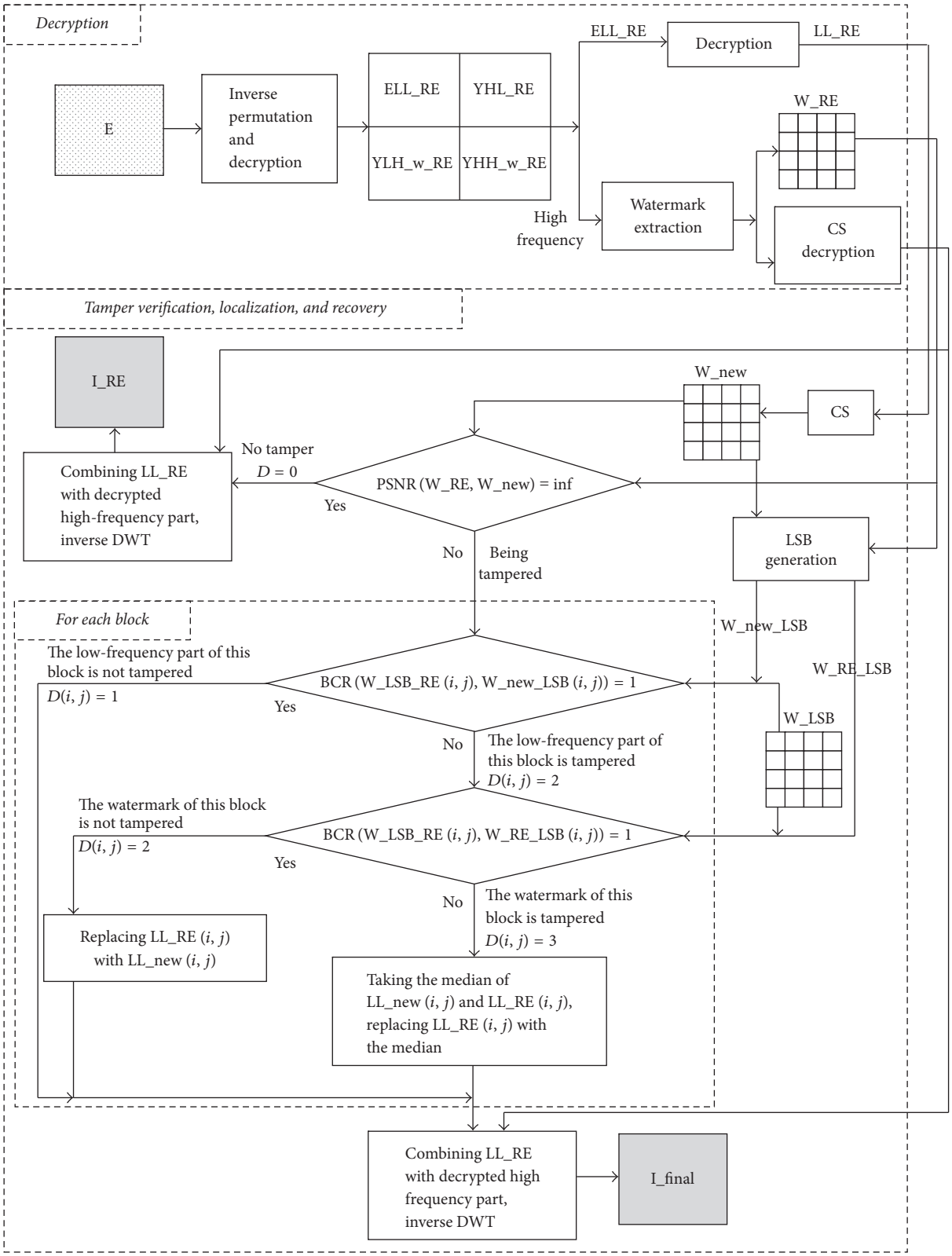


Figure 4: Watermark extraction, image decryption, tamper verification, and tamper localization and recovery.

these 16-bit planes, respectively, and then reassemble the scrambled bit panes to get the decrypted integer part of the low frequency LL_int_RE.

(3) Combine LL_int_RE with ELL_sig_RE and ELL_de_RE to form the decrypted image of the low frequency LL_RE.

(B) High-Frequency Part Decryption. According to the private key k_3, reconstruct each block of YHL_RE, YLH_RE, and YHH_RE with compressive sensing to get the recovered HL_RE, LH_RE, and HH_RE.

(C) Image Decryption. Combine LL_RE with HL_RE, LH_RE, and HH_RE. Perform inverse DWT to get the recovered image I_RE.

4.3. Tamper Detection, Localization, and Recovery

4.3.1. Tamper Detection. Since the low-frequency part contains the important information of image and the high-frequency part contains the unimportant information of image, the former is encrypted with traditional encryption algorithm while the latter is processed with CS. Due to the lossy compression of CS, it cannot perfectly recover the original image. Therefore, the low-frequency part should be recovered as completely as possible. Watermark is generated from the low-frequency part with CS, the sensibility of which can be used for tamper detection. Moreover, watermark is embedded into the high-frequency part. For the reversibility of watermark embedding, watermark is mainly used for tamper verification and recovery in low-frequency part. Generated from watermark and transmitted via secure channel, the characteristic digest value can be used for tamper localization that occurred to the watermark.

Based on the above theoretical analysis, three kinds of tampers may happen to the watermarked encrypted image during the transmission in the channel, that is, tamper with watermark, tamper with low-frequency part, and tamper with both low-frequency part and watermark.

(A) Data Preprocessing

(1) Firstly, generate a new watermark matrix W_new from the decrypted LL_RE.

(2) Secondly, recover a new image of low-frequency part LL_new from the extracted watermark W_RE.

(B) Comparison betweenW_RE and W_new

(1) If PSNR(W_RE, W_new) = Inf, W_RE is exactly the same as W_new, which means that LL_RE and W_RE are correctly recovered. Therefore, I_RE is the very image that has been correctly recovered. No tamper occurs.

For LL_RE, its related operations include traditional encryption and decryption, which are reversible. So it can be correctly recovered unless tamper occurs. If the low-frequency part is tampered, the decrypted LL_RE will vary and the newly generated watermark W_new will change. If the watermark is tampered, the extracted watermark W_RE will vary. Since these two are the same, it is believed that no tamper occurs and the low-frequency part LL_RE is correct.

(2) If PSNR(W_RE, W_new) $\neq$ Inf, the low-frequency part or the watermark is tampered. Tamper localization and tamper recovery are needed.

If the low-frequency part is partly tampered, the decrypted LL_RE will vary and the newly generated watermark W_new will change. If the watermark is tampered, the extracted watermark W_RE will vary. Under these circumstances, accurate tamper localization will greatly contribute to the recovery of image.

4.3.2. Tamper Localization and Recovery. When a tamper is detected, a comparison between the characteristic digest values generated from the extracted watermark and the one transmitted via a secure channel is needed for tamper localization, tamper content authentication, and tamper recovery. This process runs on each block. The blocks without tamper remain unchanged.

(A) Data Preprocessing

(1) Generate new characteristic digest values W_new_LSB from the watermark W_new.

(2) Generate new characteristic digest values W_RE_LSB from the extracted watermark W_RE.

(3) Suppose that D is a zero matrix with the same size as the watermark block number. Take D as tamper localization matrix. If the watermark block (i, j) is tampered, then $D(i, j) = 1$. If the low-frequency part block (i, j) is tampered, then $D(i, j) = 2$. If both the watermark block (i, j) and the low-frequency part block (i, j) are tampered, then $D(i, j) = 3$. If no tamper occurs, $D(i, j) = 0$.

(B) Tamper Localization and Recovery. Compare W_LSB with W_RE_LSB and W_new_LSB, respectively, for tamper localization. Since they are all generated directly or indirectly from the low-frequency part LL after block division and W_LSB is transmitted to the receiver side after being coded with run-length encoding, they are suitable for tamper localization.

(1) Start from the first block. For the block (i, j), take the elements W_LSB(i, j), W_RE_LSB(i, j), and W_new_LSB(i, j) from W_LSB, W_RE_LSB, and W_new_LSB, respectively.

(2) Compare W_LSB(i, j) with W_RE_LSB(i, j) and W_new_LSB(i, j), respectively:

$$s = \mathrm{BCR}\left(\mathrm{W\_LSB}\left(i, j\right), \mathrm{W\_RE\_LSB}\left(i, j\right)\right), \quad (5)$$

$$t = \mathrm{BCR}\left(\mathrm{W\_LSB}\left(i, j\right), \mathrm{W\_new\_LSB}\left(i, j\right)\right). \quad (6)$$

(a) If $s \neq 1$ and $t = 1$, it is believed that the recovered watermark W_new(i, j) of this block is correct, which means

Table 1: The PSNRs of watermarked encrypted images.

Image	Couple	Lena	Pepper	Milkdrop	Lake	Baboon	Airfield	Plane
PSNR (dB)	22.49	22.36	23.53	23.64	22.45	24.71	23.51	23.46

Table 2: Correlation coefficients of the watermarked encrypted images.

	Couple	Lena	Pepper	Milkdrop	Lake	Baboon	Airfield	Plane
Horizontal	0.0017	0.0072	−0.0040	0.0080	0.0001	−0.0027	−0.0049	0.0084
Vertical	0.0002	0.0019	−0.0022	0.0005	−0.0020	−0.0021	−0.0061	0.0039
Diagonal	−0.0027	−0.0074	−0.0036	−0.0007	0.0053	0.0009	0.0003	0.0069

that there is no tamper occurring in this low-frequency part block LL_RE(i, j). The extracted watermark block W_RE(i, j) is tampered; $D(i, j) = 1$. The low-frequency part block LL_RE(i, j) can remain unchanged.

(b) If $s = 1$ and $t \neq 1$, it is believed that the extracted watermark block W_RE(i, j) is correct while the newly generated watermark W_new_LSB(i, j) is incorrect, which means that the low-frequency part is tampered; $D(i, j) = 2$. Then replace LL_RE(i, j) with LL_new(i, j) which is recovered from W_RE(i, j).

(c) If $s \neq 1$ and $t \neq 1$, it is believed that both the extracted watermark and the low-frequency part of this block are tampered; $D(i, j) = 3$. Then replace LL_RE(i, j) with LL_mid(i, j) which is the median of LL_RE(i, j) and LL_new(i, j).

(d) Perform the above operations to each block successively. And finally tamper localization and recovery can be realized.

(C) Combine LL_RE with HL_RE, LH_RE, and HH_RE. Then inverse DWT can help to get the final recovered image I_final.

(D) If both the extracted watermark and the low frequency were tampered, some existing pixel prediction techniques [39–41] with full use of spatial correlation can be employed to further improve the visual quality of the recovered image I_final.

(a) With the help of the tamper localization matrix D, the tampered blocks and their neighboring blocks can be easily found out.

(b) For all the pixels in the tampered block, pixel prediction will begin from the tampered pixels with most nontampered neighboring pixels. If two or more neighboring blocks were tampered, theses blocks can be taken as an integrated whole to select the prediction beginning pixel. The predicted pixels can be used as nontampered pixels for next predictions.

(c) Apply corresponding pixels prediction algorithms to further improve the visual quality of image. Without loss of generality, the method in [41] is selected in this paper. After prediction, an improved image will be obtained.

5. Experimental Results and Performance Analysis

The test image set of this proposed scheme consists of 8 standard test images of size 512 × 512, that is, Couple, Lena, Peppers, Milkdrop, Lake, Baboon, Airfield, and Plane, shown in Figure 5.

5.1. Image Quality Analysis

(A) Watermarked Encrypted Image Quality. In the proposed scheme, the watermark is generated from the low-frequency part of cover image with CS and then is embedded into the encrypted high-frequency part. After encryption and permutation, the original image and watermark cannot be seen from the watermarked encrypted image any more. That is to say, the watermark and original image are well masked. Table 1 shows that the PSNR of different encrypted images are all below 25 dB. According to Table 2, the correlation coefficients of eight test images after encryption and data embedding are all close to 0. For the watermarked encrypted images, as shown in Figure 6, one can see nothing related to the original image and cannot distinguish whether a watermark is embedded into this image.

(B) Recovered Image Quality. In this proposed scheme, the encryption key and embedding key are employed during the process of image encryption and watermark embedding. Moreover, the high-frequency part is encrypted with CS while the low-frequency part is encrypted with traditional encryption algorithms. When the encrypted image is not tampered, the distortion of the recovered image only results from the reconstruction of high-frequency part with CS. Since other operations are all reversible and the watermark extraction is also reversible, the quality of the recovered image with correct keys and without tamper is reasonably high.

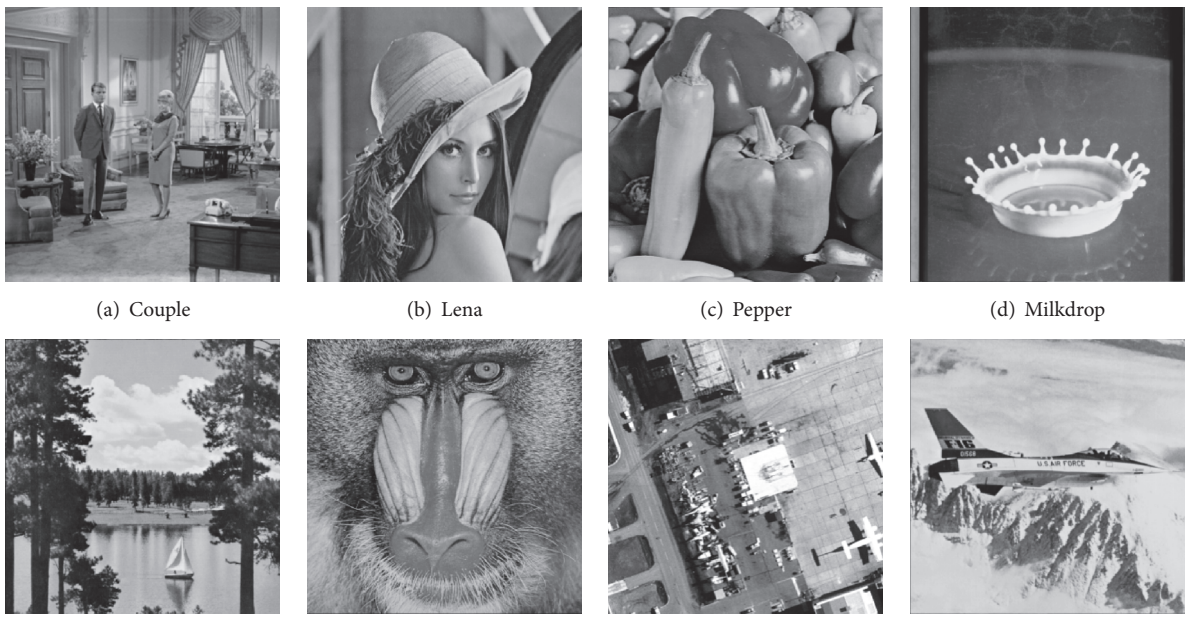
5.2. Image Authentication Performance Analysis. The watermark is generated from the low-frequency part with CS in order to perform accuracy tamper detection with the sensibility of CS and CS measurements.

Firstly, data preprocessing is done:

(1) Conduct inverse DWT of the decrypted LL_RE and the decrypted high-frequency part to get the recovered image I_RE.

(2) According to the extracted watermark W_RE, recover the low-frequency part LL_new. Conduct inverse DWT of LL_new and the decrypted high-frequency part to get recovered image I_new.

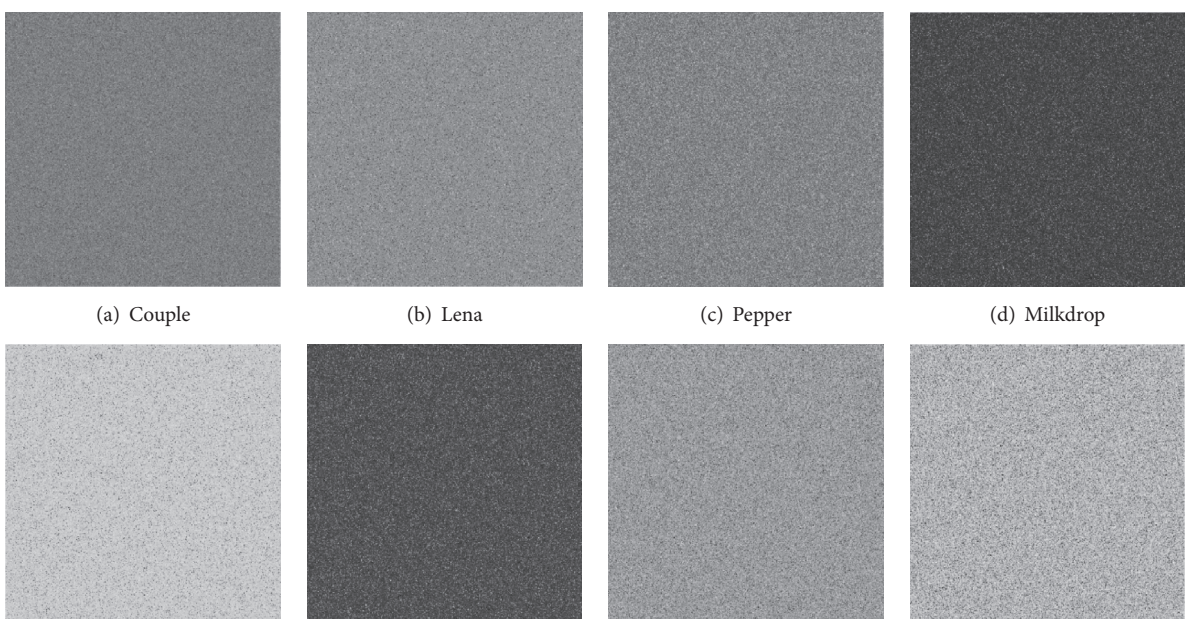
(3) According to the tamper detection and recovery method, recover the low-frequency part LL_RE.



(a) Couple (b) Lena (c) Pepper (d) Milkdrop

(e) Lake (f) Baboon (g) Airfield (h) Plane

Figure 5: The test images.



(a) Couple (b) Lena (c) Pepper (d) Milkdrop

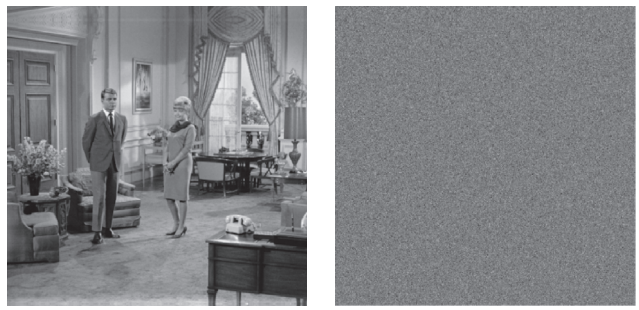
(e) Lake (f) Baboon (g) Airfield (h) Plane

Figure 6: The watermarked encrypted images.

Conduct inverse DWT of LL_RE and the decrypted high-frequency part to get tamper recovered image I_final.

In this experiment, tamper simulation is to replace the elements of some rows with 1. Figures 7–10 show the tamper localization and recovery effects when the low-frequency part or watermark is tampered. Without loss of generality, image Couple is taken as an example. Figure 7 shows the original image and watermarked encrypted image.

Figure 8 shows the tamper localization matrix, recovered image, and tamper recovered image when the low-frequency part is tampered. As can be seen, when a low-frequency part block of the watermarked encrypted image is tampered, the corresponding element of tamper localization matrix D will be changed into 2. The directly decrypted image, shown in (b), is damaged. However, since the watermark is not tampered, the recovered low-frequency part LL_new is correct and the quality of decrypted image I_new is



(a) (b)

Figure 7: The original image I (a) and the watermarked encrypted image E (b).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
10	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

(a)

(b) I_RE (c) I_new (d) I_final

Figure 8: The effect of tamper detection and recovery when low frequency is tampered.

fine, shown in (c). In other words, the proposed tamper recovery algorithm can identify that the low-frequency part is tampered and then will replace corresponding tampered block with right watermark block to get the final image I_final, shown in (d).

Figure 9 shows the tamper localization matrix, recovered image, and tamper recovered image when the watermark is tampered. As can be seen, when the watermark of a block in watermarked encrypted image is tampered, the corresponding element of tamper localization matrix D will be changed into 1. Since the low-frequency part of this block is not tampered, there is no modification to be done and it can be directly decrypted to get a good quality image I_RE, shown in (b). But since the watermark is tampered, the recovered low-frequency part LL_new is incorrect and the quality of the decrypted image I_new is damaged, shown in (c). In other words, the proposed tamper recovery algorithm can identify that the watermark is tampered and then conduct

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

(a)

(b) I_RE

(c) I_new

(d) I_final

Figure 9: The effect of tamper detection and recovery when watermark is tampered.

recovery operations to get the image I_final, shown in (d).

Figure 10 shows the tamper localization matrix, recovered image, and tamper recovered image when both the low-frequency part and the watermark are tampered. As can be seen, when both the low-frequency part and the watermark of a block in watermarked encrypted image are tampered, the corresponding element of tamper localization matrix D will be changed into 3. For the low-frequency part of this block is tampered, the directly decrypted image, shown in (b), is damaged. Since the watermark is tampered, the recovered low-frequency part LL_new is incorrect and the quality of decrypted image I_new is damaged, shown in (c). In other words, the proposed tamper recovery algorithm can identify that both the low-frequency part and the watermark of this block are tampered, then replaces LL_RE with the median of LL_RE and LL_new, and finally decrypts the image to get a relatively high quality image I_final, shown in (d). With pixel prediction techniques, the visual quality of I_final can be further improved. As shown in (e), though tampered traces still can be seen by the naked eye the heavily tampered blocks have been well improved.

In general, the proposed scheme has better tamper verification and recovery effects on this kind of local tampers. The smaller the block size is, the more accurate the tamper localization is.

5.3. Image Security Analysis. In the proposed scheme, image encryption and watermark embedding are alternate, and encryption key and embedding key are mutually bounded, which make the scheme secure. Moreover, the scheme can resist cropping attacks to a certain extent. Take Lena as an example to get the recovered images from the watermarked encrypted images with different cropping strengths. As can be seen in Table 3, when the watermarked encrypted image is cropped within a certain range, the directly recovered image and the image recovered from watermark will be affected in different degrees. However, for the image recovered with the proposed tamper recovery scheme, its PSNR will be the better one of the former two recovered images.

6. Conclusions

In this paper, we propose a novel tamper verification and recovery scheme for encrypted images with CS. After DWT, the original image can be divided into important part, that is, low-frequency part, and unimportant part, that is, high-frequency part. The watermark and characteristic digest value

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
10	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

(a)

(b) I_RE

(c) I_new

(d) I_final

(e) I_improved

Figure 10: The effect of tamper detection and recovery when watermark and low frequency are tampered.

Table 3: PSNR and NC of the recovered images through the different proportion of cropping attacks.

Cropping ratio	PSNR (dB)			NC		
	I_RE	I_new	I_final	I_RE	I_new	I_final
0	36.41	36.38	36.41	0.9987	0.9984	0.9987
1/64	34.51	33.58	34.47	0.9971	0.9921	0.9969
1/32	32.79	32.17	32.70	0.9943	0.9914	0.9986
1/16	30.90	29.51	30.09	0.9830	0.9247	0.9555
1/8	29.47	28.24	28.45	0.9797	0.9144	0.9390
1/4	28.11	26.86	27.04	0.9687	0.8166	0.8531

are generated from the low-frequency part with block CS. The characteristic digest value will be encoded and then transmitted via secure channel together with private keys. The watermark is designed mainly for tamper recovery and is embedded into the high-frequency part processed with CS. The receiver can employ the extracted watermark and characteristic digest value to perform accurate tamper detection, localization, and recovery. Theoretical analysis and experimental simulations show that in an unreliable environment the proposed scheme is robust and secure against moderate attacks, such as cropping attacks. Moreover, the tampered blocks can be accurately and effectively found out with tamper localization matrix and the tampered image can be well recovered. Comparing with the existing image

authentication algorithms, the proposed scheme can simultaneously implement tamper verification, tamper content identification, tamper localization, and tamper recovery. With great robustness and security, the scheme can adequately meet the need of secure image transmission under unreliable conditions.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work was funded by the National Natural Science Foundation of China (Grant nos. 61572089, 61502399, and 61633005), the Natural Science Foundation of Chongqing Science and Technology Commission (Grant nos. cstc2017jcyjBX0008, cstc2014jcyjA40030, and cstc2015jcyjA40039), the project supported by Graduate Student Research and Innovation Foundation of Chongqing (Grant no. CYB17026), the Chongqing Higher Education Reform Projects (Grant no. 153012), and the Fundamental Research Funds for the Central Universities (Grant nos. 106112017CDJQJ188830 and 106112017CDJXY180005).

References

[1] A. S. Rajput, N. Mishra, and S. Sharma, "Towards the growth of image encryption and authentication schemes," in *Proceedings of the 2013 2nd International Conference on Advances in Computing, Communications and Informatics, ICACCI 2013*, pp. 454–459, Mysore, India, August 2013.

[2] P. Zhen, G. Zhao, L. Min, and X. Jin, "Chaos-based image encryption scheme combining DNA coding and entropy," *Multimedia Tools and Applications*, vol. 75, no. 11, pp. 6303–6319, 2016.

[3] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," in *Proceedings of the IEEE International Conference Image Processing (ICIP '94)*, vol. 2, pp. 86–90, Austin, Tex, USA, November 1994.

[4] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.

[5] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 16, no. 3, pp. 354–362, 2006.

[6] S. Singh, T. J. Siddiqui, R. Singh, and H. V. Singh, "DCT-domain robust data hiding using chaotic sequence," in *Proceedings of the 2011 International Conference on Multimedia, Signal Processing and Communication Technologies, IMPACT 2011*, pp. 300–303, Aligarh, India, December 2011.

[7] C. C. Lin and P. F. Shiu, "High capacity data hiding scheme for dct-based images," *Journal of Information Hiding & Multimedia Signal Processing*, vol. 1, no. 3, 2010.

[8] Y. K. Lin, "High capacity reversible data hiding scheme based upon discrete cosine transformation," *Journal of Systems & Software*, vol. 85, no. 10, pp. 2395–2404, 2012.

[9] H. Liu, J. Liu, J. Huang, D. Huang, and Y. Q. Shi, "A robust DWT-based blind data hiding algorithm," *Proceedings - IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 672–675, 2002.

[10] H.-Y. Huang and S.-H. Chang, "A lossless data hiding based on discrete Haar wavelet transform," in *Proceedings of the 10th IEEE International Conference on Computer and Information Technology, CIT-2010, 7th IEEE International Conference on Embedded Software and Systems, ICESS-2010, 10th IEEE Int. Conf. Scalable Computing and Communications, ScalCom-2010*, pp. 1554–1559, Bradford, UK, July 2010.

[11] F. Li, Q. Mao, and C. C. Chang, "Reversible data hiding scheme based on the Haar discrete wavelet transform and interleaving prediction method," *Multimedia Tools & Applications*, pp. 1–20, 2017.

[12] Z. Qian, X. Zhang, Y. Ren, and G. Feng, "Block cipher based separable reversible data hiding in encrypted images," *Multimedia Tools & Applications*, vol. 75, no. 21, pp. 13749–13763, 2016.

[13] T. C. Lu, C. C. Chang, and Y. H. Huang, "High capacity reversible hiding scheme based on interpolation, difference expansion, and histogram shifting," *Multimedia Tools & Applications*, vol. 72, no. 1, pp. 417–435, 2014.

[14] X. P. Zhang, "Separable reversible data hiding in encrypted image," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 826–832, 2012.

[15] X. P. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255–258, 2011.

[16] W. Hong, T.-S. Chen, and H.-Y. Wu, "An improved reversible data hiding in encrypted images using side match," *IEEE Signal Processing Letters*, vol. 19, no. 4, pp. 199–202, 2012.

[17] T.-C. Lin and C.-M. Lin, "Wavelet-based copyright-protection scheme for digital images based on local features," *Information Sciences*, vol. 179, no. 19, pp. 3349–3358, 2009.

[18] G. S. Kalra, R. Talwar, and H. Sadawarti, "Adaptive digital image watermarking for color images in frequency domain," *Multimedia Tools & Applications*, vol. 74, no. 17, pp. 6849–6869, 2014.

[19] H.-P. Chen, X.-J. Shen, and W. Wei, "Digital signature algorithm based on hash: Round function and self-certified public key system," in *Proceedings of the 1st International Workshop on Education Technology and Computer Science, ETCS 2009*, pp. 618–624, Hubei, China, March 2009.

[20] C. Wang and X. Zhuang, "A watermarking scheme based on digital images' signatures," in *Proceedings of the 2nd International Conference on Multimedia Technology, ICMT 2011*, pp. 125–127, Hangzhou, China, July 2011.

[21] R. Kaur and A. Kaur, "Digital signature," in *Proceedings of the Turing 100 - International Conference on Computing Sciences, ICCS 2012*, pp. 295–301, Phagwara, India, September 2012.

[22] Q. Gu and T. Gao, "A new image authentication based on reversible watermarking algorithm," in *Proceedings of the 7th World Congress on Intelligent Control and Automation, WCICA'08*, pp. 2727–2731, Chongqing, China, June 2008.

[23] S. Walton, "Information authentication for a slippery new age," *Dr. Dobbs Journal*, vol. 20, no. 4, pp. 18–26, 1995.

[24] M. Al Baloshi and M. E. Al-Mualla, "A DCT-based watermarking technique for image authentication," in *Proceedings of the 2007 IEEE/ACS International Conference on Computer Systems and Applications, AICCSA 2007*, pp. 754–760, Amman, Jordan, May 2007.

[25] Y. M. Y. Hasan and A. M. Hassan, "Tamper detection with self-correction hybrid spatial-DCT domains image authentication technique," in *Proceedings of the ISSPIT 2007 - 2007 IEEE*

International Symposium on Signal Processing and Information Technology, pp. 369–374, Giza, Egypt, December 2007.

[26] D. Singh and S. K. Singh, "DCT based efficient fragile watermarking scheme for image authentication and restoration," *Multimedia Tools & Applications*, vol. 76, pp. 1–25, 2015.

[27] H. Liu and Y. Hu, "A wavelet-based watermarking scheme with authentication and recovery mechanism," in *Proceedings of the International Conference on Electrical and Control Engineering, ICECE 2010*, pp. 323–326, Wuhan, China, June 2010.

[28] L.-J. Wang and M.-Y. Syue, "Image authentication and recovery using wavelet-based multipurpose watermarking," in *Proceedings of the 2013 10th International Joint Conference on Computer Science and Software Engineering, JCSSE 2013*, pp. 31–36, Maha Sarakham, Thailand, May 2013.

[29] C. L. Li, A. H. Zhang, Z. F. Liu, L. Liao, and D. Huang, "Semi-fragile self-recoverable watermarking algorithm based on wavelet group quantization and double authentication," *Multimedia Tools & Applications*, vol. 74, no. 23, pp. 10581–10604, 2015.

[30] G. Valenzise, M. Tagliasacchi, S. Tubaro, G. Cancelli, and M. Barni, "A compressive-sensing basedwatermarking scheme for sparse image tampering identification," in *Proceedings of the 2009 IEEE International Conference on Image Processing, ICIP 2009*, pp. 1265–1268, Cairo, Egypt, November 2009.

[31] V. K. Veena, G. Jyothish Lal, S. Vishnu Prabhu, S. Sachin Kumar, and K. P. Soman, "A robust watermarking method based on Compressed Sensing and Arnold scrambling," in *Proceedings of the 2012 International Conference on Machine Vision and Image Processing, MVIP 2012*, pp. 105–108, Taipei, Taiwan, December 2012.

[32] Y. Rachlin and R. D. Baron, "The secrecy of compressed sensing measurements," in *Proceedings of the 46th Annual Allerton Conference on Communication, Control, and Computing*, pp. 813–817, IEEE, Urbana, Ill, USA, September 2008.

[33] S. A. Hossein, A. E. Tabatabaei, and N. Zivic, "Security analysis of the joint encryption and compressed sensing," in *Proceedings of the 20th Telecommunications Forum (TELFOR '12)*, pp. 799–802, Belgrade, Serbia, November 2012.

[34] G. F. Chen, S. X. Guo, Y. Li, and L. Li, "Digital image watermark algorithm based on compressive sensing," *Modern Electronics Technique*, vol. 35, no. 13, pp. 98–104, 2012.

[35] H.-C. Huang, F.-C. Chang, C.-H. Wu, and W.-H. Lai, "Watermarking for compressive sampling applications," in *Proceedings of the 2012 8th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP 2012*, pp. 223–226, Piraeus, Greece, July 2012.

[36] J. S. Pan, W. Li, C. S. Yang, and L. J. Yan, "Image steganography based on subsampling and compressive sensing," *Multimedia Tools & Applications*, vol. 74, no. 21, pp. 9191–9205, 2015.

[37] H. C. Huang and F. C. Chang, "Robust image watermarking based on compressed sensing techniques," *Journal of Information Hiding & Multimedia Signal Processing*, vol. 5, no. 2, pp. 275–285, 2014.

[38] E. J. Candes, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9, pp. 589–592, 2008.

[39] M. Fallahpour, "Reversible image data hiding based on gradient adjusted prediction," *IEICE Electronics Express*, vol. 5, no. 20, pp. 870–876, 2008.

[40] M. Li, D. Xiao, Z. Peng, and H. Nan, "A modified reversible data hiding in encrypted images using random diffusion and accurate prediction," *ETRI Journal*, vol. 36, no. 2, pp. 325–328, 2014.

[41] X. Liao and C. Shu, "Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels," *Journal of Visual Communication & Image Representation*, vol. 28, pp. 21–27, 2015.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 2708532, 13 pages
https://doi.org/10.1155/2018/2708532

Research Article

A New Chaos-Based Color Image Encryption Scheme with an Efficient Substitution Keystream Generation Strategy

Chong Fu, Gao-yuan Zhang, Mai Zhu, Zhe Chen, and Wei-min Lei

School of Computer Science and Engineering, Northeastern University, Shenyang 110004, China

Correspondence should be addressed to Chong Fu; fuchong@mail.neu.edu.cn

Received 10 August 2017; Accepted 3 January 2018; Published 20 February 2018

Academic Editor: Leo Y. Zhang

This paper suggests a new chaos-based color image cipher with an efficient substitution keystream generation strategy. The hyperchaotic Lü system and logistic map are employed to generate the permutation and substitution keystream sequences for image data scrambling and mixing. In the permutation stage, the positions of colored subpixels in the input image are scrambled using a pixel-swapping mechanism, which avoids two main problems encountered when using the discretized version of area-preserving chaotic maps. In the substitution stage, we introduce an efficient keystream generation method that can extract three keystream elements from the current state of the iterative logistic map. Compared with conventional method, the total number of iterations is reduced by 3 times. To ensure the robustness of the proposed scheme against chosen-plaintext attack, the current state of the logistic map is perturbed during each iteration and the disturbance value is determined by plain-pixel values. The mechanism of associating the keystream sequence with plain-image also helps accelerate the diffusion process and increase the degree of randomness of the keystream sequence. Experimental results demonstrate that the proposed scheme has a satisfactory level of security and outperforms the conventional schemes in terms of computational efficiency.

1. Introduction

Nowadays, digital image information has been widely communicated over the Internet and wireless networks owing to the rapid advancements in the multimedia and communication technology. Meanwhile, the protection of digital image information against illegal usage has become an important issue. A direct and obvious way to protect image data from unauthorized eavesdropping is to employ an encryption algorithm. Unfortunately, the renowned block ciphers, such as Triple-DES, AES, and IDEA, are not suitable for practical image encryption. This is because the security of these algorithms is mainly ensured by their high computational cost, making them hard to meet the demand for online communications when dealing with digital images characterized by bulk data capacity. To meet this challenge, many different encryption technologies have been proposed. Among them, the chaos-based algorithms provide an optimal trade-off between security and efficiency. The first chaos-based image encryption scheme was suggested by Fridrich in 1998 [1]. The permutation-substitution network, introduced by Claude Shannon in his classic 1949 paper, Communication Theory of Secrecy Systems, and now a guiding principle for the design of a secure cipher, is adopted in her approach. In each round of the cipher, the pixel positions are firstly scrambled in a secret way, which leads to a great reduction in the correlation among neighboring pixels. Then, the pixel values are altered sequentially and the influence of each pixel is diffused to all its succeeding ones during the modification process. With such a structure, a minor change in one pixel of the plain-image may result in a totally different cipher-image with several overall rounds of encryption.

Conventionally, three area-preserving invertible chaotic maps, that is, the cat map, the baker map, and the standard map, are widely used for image scrambling. Unfortunately, this kind of permutation strategy suffers from two main disadvantages, that is, the periodicity of discretized version of chaotic maps and applicability to only square images [2–4]. To address these two drawbacks, Fu et al. [5] suggested an image scrambling scheme using a chaotic sequence sorting mechanism. Unfortunately, this method takes a whole row/column of an image as the scrambling unit and results in

weaker confusion effect compared with many of the existing schemes working on individual pixels. In [6], inspired by the natural ripple-like phenomenon that distorts a reflection on a water surface, Wu et al. suggested a novel scrambling algorithm that shuffle images in an n dimensional (n D) space using wave perturbations. In [7], the original pixel level matrix is considered as a natural 3D bit matrix, and a new 3D bit-level permutation algorithm is proposed. During the permutation stage, the original and target bit locations are both randomly selected to further enhance the permutation effect.

In the substitution stage, various discrete chaotic maps and continuous chaotic systems can be employed to generate keystream sequences with desired statistical properties, including the most commonly used ones like the logistic map [2], the Lorenz system [8], the Chen system [9], and varieties of high dimensional chaotic systems [10]. Obviously, low dimensional chaotic maps, especially the logistic map, have the advantages of simplicity and high efficiency but suffer from small key space; in contrast, high dimensional chaotic systems, especially the hyperchaotic systems, provide sufficiently large key space but at the expense of computations. Recently, it has been reported that many existing image encryption schemes have been successfully broken by using known/chosen-plaintext attacks [11–14]. This is due to the fact that the substitution keystream sequences used in these schemes are solely determined by the secret key. That is, the same keystream sequence is used to encrypt different plain-images unless a different secret key is used. Consequently, the keystream sequence may be determined by encrypting some specially created images (e.g., an image with all pixels having the same value) and then comparing them with their corresponding outputs. Obviously, if a keystream sequence depends on both the secret key and the plaintext, then such analysis may become impractical. For instance, in [15], the keystream elements are extracted from multiple-time iteration of the logistic map, and the iteration times are determined by plain-pixel values. Unfortunately, the redundant iteration operations downgrade the efficiency of the cryptosystem to some extent. In [16], the value of each keystream element is dynamically altered according to the plain-pixel values during the substitution process.

To better meet the challenge of online secure image communications, much research has been done on improving the efficiency of chaos-based image ciphers. For instance, in [17], Xiang et al. investigated the feasibility of selective image encryption on a bit-plane. It is concluded that only selectively encrypting the higher four bit-planes of an image can achieve an acceptable level of security. As only 50% of the whole image data are encrypted, the execution time is reduced. In [18], Wong et al. proposed a more efficient diffusion mechanism using simple table lookup and swapping techniques as a light-weight replacement of the 1D chaotic map iteration. Following this work, Chen et al. [19] presented an efficient image encryption scheme with confusion and diffusion operations being both performed based on a lookup table. The other advantage of their approach is that it can effectively tolerate the channel errors, which may lead to the corruption of cipher data. It has been demonstrated that images recovered from the damaged cipher data have satisfactory visual perception. In [20], Fu et al. introduced a novel bidirectional diffusion strategy to minimize the number of encryption rounds needed to spread the influence of each individual pixel over the entire cipher-image. Experimental results have demonstrated that their scheme takes one round of permutation and two rounds of substitution to obtain a satisfactory diffusion effect. In [16, 21–23], chaos-based image ciphers using a bit-level permutation were suggested. Owing to the substitution effect introduced in the permutation stage, the number of iteration rounds required by the time-consuming substitution procedure is reduced, and hence a shorter encryption time is needed. In [8], Fu et al. suggested a fast chaos-based image cipher with the permutation key determined by the hash value of the original image. Owing to the avalanche property of hash function, completely different shuffled images will be produced even if there is a tiny difference between the original ones, thereby accelerating the diffusion process. In [24], Chen et al. presented a novel image encryption scheme using a Gray-code-based permutation. Taking full advantage of (n, p, k)-Gray-code achievements, the new permutation strategy provides superior computational efficiency. In [25], Hua et al. introduced an image encryption algorithm based on a new two-dimensional sine logistic modulation map (2D-SLMM). Compared with corresponding seed maps, the new map has wider chaotic range, more parameters, and complex chaotic properties while remaining of relatively low implementation cost. Accordingly, the algorithm provides a good trade-off between security and efficiency.

Conventionally, in the substitution stage, one keystream element is obtained from the current value of a state variable of an iterative chaotic system. That is, to generate a keystream sequence of length m, a n-dimensional chaotic system should be iterated $T = \text{round}(m/n)$ times. In the present paper, we introduce an efficient logistic map-based keystream generation strategy that can simultaneously extract three keystream elements from the current state of the map. As a result, the total number of iterations is reduced by 3 times and the encryption time is shortened. In the permutation stage, the positions of subpixel in each color channel of the plain-image are scrambled across the entire color space using a pixel-swapping strategy under the control of a keystream sequence generated from the hyperchaotic Lü system. To ensure the robustness of the proposed scheme against chosen-plaintext attack, the current state of the logistic map is perturbed during each iteration and the disturbance value is determined by plain-pixel values. The mechanism of associating the keystream sequence with plain-image also helps accelerate the diffusion process and increase the degree of randomness of the keystream sequence. Experimental results demonstrate that the proposed scheme has a satisfactory level of security and outperforms the conventional schemes in terms of computational efficiency.

The rest of this paper is organized as follows. The proposed permutation and substitution algorithms are thoroughly described in Sections 2 and 3, respectively. In Section 4, the degree of randomness of the substitution keystream sequences generated using our proposed method

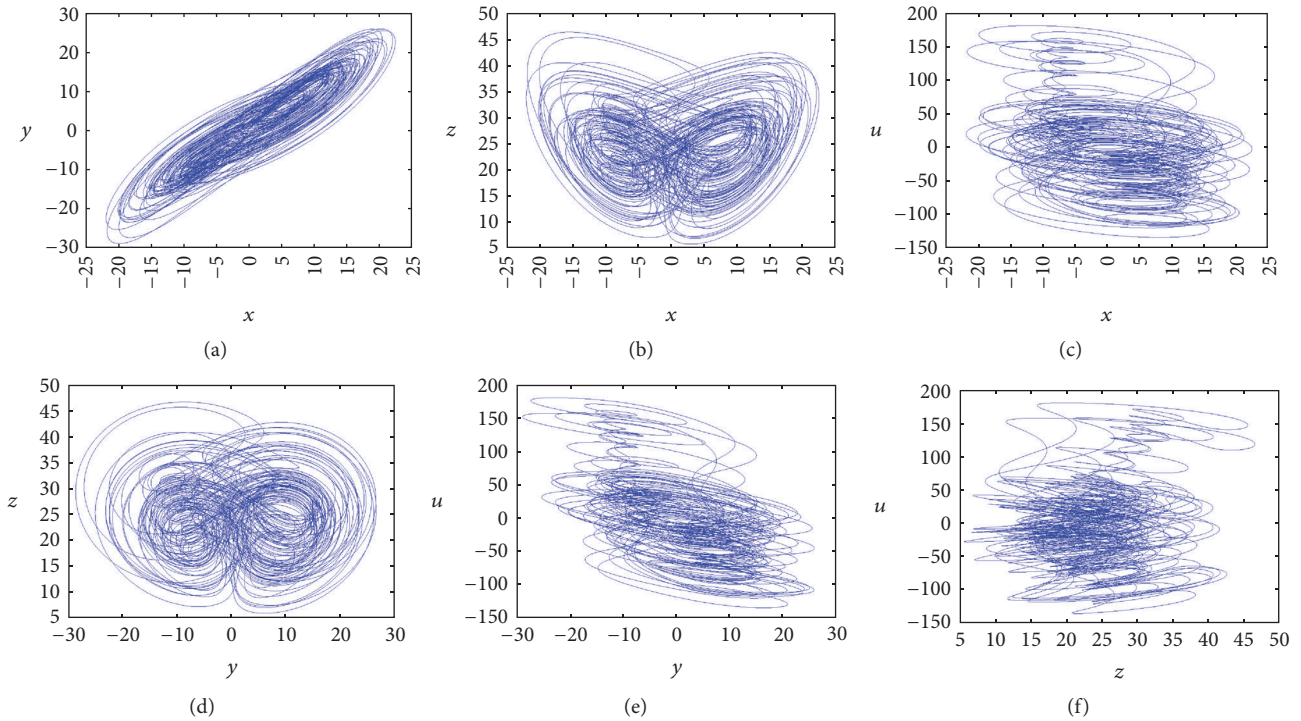


FIGURE 1: The projections of phase portrait of system (1) with $a = 36$, $b = 3$, $c = 20$, and $d = 1.3$. (a) x-y plane. (b) x-z plane. (c) x-u plane. (d) y-z plane. (e) y-u plane. (f) z-u plane.

is evaluated by using NIST test suite. In Section 5, the confusion and diffusion performance of the proposed cryptosystem are analyzed. The security and efficiency of the cryptosystem are analyzed in Sections 6 and 7, respectively. Finally, Section 8 concludes the paper.

2. Color Image Permutation Using a Pixel-Swapping Strategy

The hyperchaotic Lü system [26], which is employed in our scheme to generate the permutation keystream sequence, is described by

$$\begin{aligned} \dot{x} &= a(y - x) + u, \\ \dot{y} &= -xz + cy, \\ \dot{z} &= xy - bz, \\ \dot{u} &= xz + du, \end{aligned} \tag{1}$$

where a, b, c are the constants of Lü system [27] and d is a control parameter. When $a = 36$, $b = 3$, $c = 20$, and $-0.35 < d \leq 1.30$, the system exhibits a hyperchaotic behavior, and the projections of its phase portrait are shown in Figure 1. Evidently, the initial conditions (x_0, y_0, z_0, u_0) of the system are the immediate candidate for the secret key for permutation, as they uniquely determine a chaotic trajectory from which the permutation keystream is extracted.

Without loss of generality, a 24-bit RGB color image of size $H \times W$ is used as an input. The detailed permutation process is described as follows.

Step 1. Arrange the colored subpixels in the input image to a one-dimensional byte array imgData $= \{p_0, p_1, \ldots, p_{3\times H\times W-1}\}$ in the order from left to right, top to bottom.

Step 2. Generate a chaotic sequence of length $L_{\text{perm}} =$ len (imgData) -1 by iterating system (1), where len(x) returns the length of sequence x.

Step 2.1. Preiterate system (1) for N_0 times to avoid the harmful effect of transitional procedure, where N_0 is a constant. The system can be numerically solved by using fourth-order Runge-Kutta method, as given by

$$\begin{aligned} x_{n+1} &= x_n + \left(\frac{h}{6}\right)(K_1 + 2K_2 + 2K_3 + K_4), \\ y_{n+1} &= y_n + \left(\frac{h}{6}\right)(L_1 + 2L_2 + 2L_3 + L_4), \\ z_{n+1} &= z_n + \left(\frac{h}{6}\right)(M_1 + 2M_2 + 2M_3 + M_4), \\ u_{n+1} &= u_n + \left(\frac{h}{6}\right)(N_1 + 2N_2 + 2N_3 + N_4), \end{aligned} \tag{2}$$

where

$$
\begin{aligned}
K_j &= a\left(y_n - x_n\right) + u_n,\\
L_j &= -x_n z_n + c y_n,\\
M_j &= x_n y_n - b z_n,\\
N_j &= x_n z_n + d u_n,\\
&\qquad\qquad \left(\text{with } j = 1\right),\\
K_j &= a\left[\left(y_n + \frac{hL_{j-1}}{2}\right) - \left(x_n + \frac{hK_{j-1}}{2}\right)\right]\\
&\quad + \left(u_n + \frac{hN_{j-1}}{2}\right),\\
L_j &= -\left(x_n + \frac{hK_{j-1}}{2}\right)\left(z_n + \frac{hM_{j-1}}{2}\right)\\
&\quad + c\left(y_n + \frac{hL_{j-1}}{2}\right),\\
M_j &= \left(x_n + \frac{hK_{j-1}}{2}\right)\left(y_n + \frac{hL_{j-1}}{2}\right)\\
&\quad - b\left(z_n + \frac{hM_{j-1}}{2}\right),\\
N_j &= \left(x_n + \frac{hK_{j-1}}{2}\right)\left(z_n + \frac{hM_{j-1}}{2}\right)\\
&\quad + d\left(u_n + \frac{hN_{j-1}}{2}\right),\\
&\qquad\qquad \left(\text{with } j = 2, 3\right),\\
K_j &= a\left[\left(y_n + hL_{j-1}\right) - \left(x_n + hK_{j-1}\right)\right]\\
&\quad + \left(u_n + hN_{j-1}\right),\\
L_j &= -\left(x_n + hK_{j-1}\right)\left(z_n + hM_{j-1}\right) + c\left(y_n + hL_{j-1}\right),\\
M_j &= \left(x_n + hK_{j-1}\right)\left(y_n + hL_{j-1}\right) - b\left(z_n + hM_{j-1}\right),\\
N_j &= \left(x_n + hK_{j-1}\right)\left(z_n + hM_{j-1}\right) + d\left(u_n + hN_{j-1}\right),\\
&\qquad\qquad \left(\text{with } j = 4\right),
\end{aligned}
\tag{3}
$$

and the step h is chosen as 0.005.

Step 2.2. Continue the iteration for $I_{\text{perm}} = \lceil L_{\text{perm}}/4 \rceil$ times, where $\lceil x \rceil$ returns the least integer greater than or equal to x. For each iteration, the current values of the four state variables x, y, z, and u are in turn stored into an array permSeq $= \{ps_0, ps_1, ps_2, \ldots, ps_{3\times H\times W-2}\}$. Obviously, the last $R_{\text{perm}} = 4 \times I_{\text{perm}} - (3 \times H \times W - 1)$ element(s) produced at the final iteration step is redundant and should be discarded.

Step 3. Extract a permutation keystream sequence permKstr $= \{pk_0, pk_1, \ldots, pk_{3\times H\times W-2}\}$ from permSeq according to

$$
\begin{aligned}
pk_m = \operatorname{pos}\left(pk_m\right) + \big(1 + \operatorname{mod}\big(\operatorname{sig}\left(\operatorname{abs}\left(ps_m\right), \alpha\right),\\
\left(\left(\operatorname{len}\left(\text{imgData}\right) - 1\right) - \operatorname{pos}\left(pk_m\right)\right)\big)\big),
\end{aligned}
\tag{4}
$$

where $\operatorname{pos}(pk_m)$ returns the position of pk_m in permKstr, that is, m, $\operatorname{abs}(x)$ returns the absolute value of x, $\operatorname{sig}(x, \alpha)$ returns the α most significant decimal digits in x, and $\operatorname{mod}(x, y)$ divides x by y and returns the remainder of the division. An α value of 15 is recommended as all the state variables in our scheme are declared as double-precision type. From (4) it can be seen that pk_m is in the range between $(\operatorname{pos}(pk_m) + 1)$ and $(\operatorname{len}(\text{imgData}) - 1)$. That is, the swapping target for each colored subpixel (except the last one) in imgData will be pseudorandomly chosen from all its succeeding ones.

Step 4. Scramble imgData by swapping each subpixel p_m $(m = 0, 1, \ldots, 3\times H\times W-2)$ with another subpixel located at pk_m.

As can be seen from the above description, the proposed permutation scheme well addresses the two problems encountered when using the discretized version of area-preserving chaotic maps. First, the proposed scheme can be applied to images of arbitrary size, whereas the area-preserving chaotic maps can be only applied to square images. Secondly, though the aperiodicity nature of a chaotic system will be deteriorated in computer realization with finite computation precision, the period length of pseudorandom keystream sequence generated by a chaotic system is by far longer than that of its discretized version. A keystream sequence with a very long period can be considered practically aperiodic when applied to images of reasonable size. That is, image scrambled by the proposed method will not return to its original state even after a huge number of iterations.

3. Color Image Substitution Using a Fast Plaintext-Dependent Keystream Generation Strategy

In the substitution stage, the logistic map [28], which is the most studied example of discrete nonlinear dynamical systems that exhibit chaotic behavior, is employed to generate the keystream sequence for mixing the subpixel values. Mathematically, the map is described by

$$
x_{n+1} = r x_n \left(1 - x_n\right), \quad x_n \in [0, 1], \; r \in [0, 4], \tag{5}
$$

where x is the state variable and r is the parameter. The logistic map behaves chaotically, interrupted by small periodic windows for values of r between about 3.57 and 4. In our scheme, r is set to 4 to avoid those nonchaotic regions. Similarly, the initial condition x_0 of the map is used as the secret key for substitution.

The detailed substitution process is described as follows.

Step 1. Preiterate the logistic map for N_0 times for the same purpose mentioned above. It should be noticed that the values of 0.5 and 0.75 are two "bad" points, trapping the iterations to the fixed points 0 and 0.75, respectively. If this case is encountered, a slight perturbation should apply.

Step 2. The logistic map is iterated continuously. For each iteration, a 24-bit (3 byte) integer can be obtained from the current state of the map according to

$$\text{pseRandInt} = \text{mod}\left[\text{sig}\left(x_{n+1}, \alpha\right), (1 \ll 24)\right], \tag{6}$$

where $n = 0, 1, \ldots, H \times W - 1$ and "$\ll s$" denotes a left shift by s bit. For instance, $(1 \ll 24)$ returns bitwise representation of 1 shifted to the left by 24 bits, which is equivalent to 2^{24}.

Step 3. Extract three keystream elements from pseRandInt according to

$$\begin{aligned} \text{kstrEle}_{(\text{red})} &= \left(\text{pseRandInt} \gg 16\right) \ \& \ \text{0xFF}, \\ \text{kstrEle}_{(\text{green})} &= \left(\text{pseRandInt} \gg 8\right) \ \& \ \text{0xFF}, \\ \text{kstrEle}_{(\text{blue})} &= \text{pseRandInt} \ \& \ \text{0xFF}, \end{aligned} \tag{7}$$

where "$\gg s$" denotes a right shift by s bit and & denotes a bitwise AND operation. It can be seen from (7) that the upper, middle, and lower 8 bits of pseRandInt are assigned to $\text{kstrEle}_{(\text{red})}$, $\text{kstrEle}_{(\text{green})}$, $\text{kstrEle}_{(\text{blue})}$, respectively.

Step 4. Convert the plain-pixel to its cipher form according to

$$\begin{aligned} c_{3n} &= \text{kstrEle}_{(\text{red})} \oplus \text{mod}\left(\left(p_{3n} + \text{kstrEle}_{(\text{red})}\right), G_L\right) \\ &\quad \oplus c_{3n-1}, \\ c_{3n+1} &= \text{kstrEle}_{(\text{green})} \\ &\quad \oplus \text{mod}\left(\left(p_{3n+1} + \text{kstrEle}_{(\text{green})}\right), G_L\right) \oplus c_{3n}, \\ c_{3n+2} &= \text{kstrEle}_{(\text{blue})} \\ &\quad \oplus \text{mod}\left(\left(p_{3n+2} + \text{kstrEle}_{(\text{blue})}\right), G_L\right) \oplus c_{3n+1}, \end{aligned} \tag{8}$$

where $(p_{3n}, p_{3n+1}, p_{3n+2})$ and $(c_{3n}, c_{3n+1}, c_{3n+2})$ are the three colored subpixels of the currently operated pixel and its output cipher-pixel, respectively, $\oplus$ performs bitwise exclusive OR operation, and G_L is the number of gray levels in the input image (for a 24-bit RGB color image, $G_L = 256$).

As can be seen from (8), the modification made to a subpixel depends not only on the keystream element but also on its previous ciphered subpixel, and thereby the influence of each subpixel can be spread over all its succeeding ones. For the first subpixel p_0, the initial value c_{-1} can be set as a constant.

Step 5. Make the keystream elements depend on the plain-pixel by perturbing the state variable of logistic map according to

$$\begin{aligned} x_{n+1} &= x_{n+1} + \beta \quad \text{for } 0 < x_n < 0.5, \\ x_{n+1} &= x_{n+1} - \beta \quad \text{for } 0.5 < x_n < 1, \end{aligned} \tag{9}$$

where

$$\beta = 0.1 \times \sum_{j=0}^{2} \frac{p_{3n+j}}{\left[3 \times \left(G_L - 1\right)\right]} \tag{10}$$

is the disturbance value determined by the original values of the three colored subpixels of the currently operated pixel. It is clear that β falls within the range between 0 and 0.1, keeping the logistic map operating in chaotic region.

Step 6. Return to Step 2 until all the subpixels in imgData are encrypted.

Step 7. Perform several rounds of the overall permutation-substitution operations so as to spread the influence of each individual subpixel over the entire cipher-image.

Step 8. Produce the final output by adding a file header identical to that of the input image to imgData.

The decryption procedure is similar to that of the encryption process except that some steps are followed in a reversed order. Particularly, the inverse of (8) is given by

$$\begin{aligned} p_{3n} &= \text{mod}\left[\left(k\text{strEle}_{(\text{red})} \oplus c_{3n} \oplus c_{3n-1} + G_L \right.\right. \\ &\quad \left.\left. - \text{kstrEle}_{(\text{red})}\right), G_L\right], \\ p_{3n+1} &= \text{mod}\left[\left(\text{kstrEle}_{(\text{green})} \oplus c_{3n+1} \oplus c_{3n} + G_L \right.\right. \\ &\quad \left.\left. - \text{kstrEle}_{(\text{green})}\right), G_L\right], \\ p_{3n+2} &= \text{mod}\left[\left(\text{kstrEle}_{(\text{blue})} \oplus c_{3n+2} \oplus c_{3n+1} + G_L \right.\right. \\ &\quad \left.\left. - \text{kstrEle}_{(\text{blue})}\right), G_L\right]. \end{aligned} \tag{11}$$

In addition, as can be seen from the above description of the substitution algorithm, the perturbing of the state of the logistic map is performed after a pixel is enciphered and the disturbance value is calculated from the original values of the ciphered pixel. Accordingly, in the decryption procedure, the same disturbance value can be calculated out after a cipher-pixel is decrypted.

4. Analysis of the Randomness of the Substitution Keystream

The randomness of the keystream sequence is crucial to the security of a chaotic cipher. A cryptographically secure keystream generator should generate the keystream sequence without repetition or predictability, thus preventing different parts of a messages encrypted with the repeated parts of the keystream sequence from being intercepted or generated by an attacker. The degree of randomness of a keystream

sequence may be determined by statistical tests, and the most authoritative one is the test suite designed by the National Institute of Standards & Technology (NIST). The test suite is a statistical package consisting totally of 16 tests, which are carried out as follows: For each statistical test, a set of P values (corresponding to the set of sequences) is produced. A sequence passes a statistical test whenever the P value $\geq \alpha$ and fails otherwise, where $\alpha \in (0.001, 0.01]$ is the significance level. For each statistical test, compute the proportion of sequences that pass. The range of acceptable proportions is determined using the confidence interval defined as

$$(1-\alpha) - \sigma\sqrt{\frac{\alpha(1-\alpha)}{m}} \leq p_\alpha \leq 1.0, \tag{12}$$

where σ is the number of standard deviations and m is the sample size. In our experiments, 200 keystream sequences ($m = 200$), each with a length, in bits, as long as that of a 24-bit RGB color image of size 512×512, are generated with randomly selected substitution keys. Together with the chosen standard parameters, $\alpha = 0.01$ and $\sigma = 3$, we have $0.968893 \leq P_\alpha \leq 1.0$. If the proportion falls outside of this interval, then there is evidence that the data is nonrandom. Table 1 lists the test results for the keystream sequences generated using the proposed and conventional methods. As can be seen from this table, the proposed method has a higher pass rate in 13 of the 16 test items, and hence it generates keystream sequences with a higher degree of randomness over the conventional method.

5. Analysis of the Confusion and Diffusion Performance of the Proposed Scheme

5.1. Analysis of Confusion Performance. In order to evaluate the confusion performance of the proposed permutation method, we apply it to the standard "peppers" test image (512×512 pixels, 24-bit RGB color) and the result is compared with that of three most widely used methods based on area-preserving invertible chaotic maps, as demonstrated in Figure 2. Figure 2(a) shows the test image and Figure 2(b) is the resulting image after applying the proposed permutation method once. The permutation key, that is, the initial conditions of the hyperchaotic Lü system, is randomly chosen to be $\{x_0 = 8.14723686393179,\ y_0 = -3.13375856139019,\ z_0 = 15.0794726517402,\ u_0 = -47.8753417717149\}$. Figures 2(c)–2(e), 2(f)–2(h), and 2(i)–2(k) show the resulting images after applying the cat map, the baker map, and the standard map once, twice, and three times, respectively. The permutation keys, that is, the control parameters of the three maps, are chosen to be $\{p = 40, q = 8\}$, $n_i = \{32, 16, 32, 64, 16, 32, 64, 16, 32, 32, 64, 16, 64, 32\}$, and $K = 1024$, respectively. As can be seen from Figure 2, the proposed method takes only one round to achieve a satisfactory scrambling effect, whereas more (≥2) is needed by the three conventional ones. Moreover, the colors in the scrambled image produced by the proposed permutation method are much more uniformly distributed than that produced by the three conventional methods. This is because the proposed scheme scrambles the subpixels across all the three color channels, whereas the schemes using area-preserving chaotic maps have to scramble each color channel separately, making the dominant colors of the resulting scrambled image similar to those of its original version.

5.2. Analysis of Diffusion Performance. As is known, the diffusion property is essential to ensure the security of a cryptographic algorithm against chosen-plaintext attack. The differential analysis is the most common way to implement the chosen-plaintext attack. To do this, an opponent may firstly create two plain-images with only one-bit difference and then encrypt the two images using the same secret key. By observing the differences between the two resulting cipher-images, some meaningful relationship between plain-image and cipher-image could be found out, and it further facilitates determining the keystream. Obviously, this kind of cryptanalysis may become impractical if a cryptosystem is highly sensitive to plaintext; that is, changing one bit of the plaintext affects every bit in the ciphertext.

To measure the diffusion property of an image cryptosystem, two criteria, that is, NPCR (the number of pixel change rate) and UACI (the unified average changing intensity), are commonly used. The NPCR is used to measure the percentage of different pixel numbers between two images. Let $I_1(i, j, k)$ and $I_2(i, j, k)$ be the (i, j)th pixel in kth color channel ($k = 1, 2, 3$ denotes the red, green, and blue color channels, resp.) of two images I_1 and I_2, where $0 \leq i \leq H$, $0 \leq j \leq W$; the NPCR can be defined as

$$\text{NPCR} = \frac{\sum_{k=1}^{3}\sum_{i=1}^{H}\sum_{j=1}^{W} D(i, j, k)}{3 \times H \times W} \times 100\%, \tag{13}$$

where $D(i, j, k)$ is defined as

$$D(i, j, k) = \begin{cases} 0 & \text{if } I_1(i, j, k) = I_2(i, j, k), \\ 1 & \text{if } I_1(i, j, k) \neq I_2(i, j, k). \end{cases} \tag{14}$$

The second criterion, UACI, is used to measure the average intensity of differences between the two images. It is defined as

$$\text{UACI} = \frac{1}{3 \times H \times W}\left[\sum_{k=1}^{3}\sum_{i=1}^{H}\sum_{j=1}^{W}\frac{|I_1(i, j, k) - I_2(i, j, k)|}{G_L - 1}\right] \times 100\%. \tag{15}$$

Clearly, no matter how similar the two input images are, a good image cryptosystem should produce outputs with NPCR and UACI values ideally being equal to those of two random images, which are given by

$$\begin{aligned} \text{NPCR}_{\text{random}} &= \left(1 - \frac{1}{2^{\log_2 G_L}}\right) \times 100\%, \\ \text{UACI}_{\text{expected}} &= \frac{1}{{G_L}^2}\left(\frac{\sum_{i=1}^{G_L - 1} i(i+1)}{G_L - 1}\right) \times 100\%. \end{aligned} \tag{16}$$

For instance, the NPCR and UACI values for two random color images in 24-bit RGB format ($G_L = 256$) are 99.609% and 33.464%, respectively.

Table 1: Results of NIST statistical test.

Test items	Pass rate	
	Proposed method	Conventional method
Frequency	100.0%	97.50%
Block frequency	99.00%	99.00%
Cusum-forward	100.0%	97.00%
Cusum-reverse	100.0%	97.50%
Runs	99.50%	98.50%
Long runs of ones	99.50%	97.50%
Rank	99.00%	98.50%
Spectral DFT	99.00%	99.00%
Nonoverlapping templates	99.50%	98.50%
Overlapping templates	99.00%	97.00%
Universal	98.50%	98.00%
Approximate entropy	99.00%	97.50%
Random excursions	98.21%	99.43%
Random excursions variant	99.40%	100.0%
Linear complexity	99.50%	99.50%
Serial	99.50%	99.00%

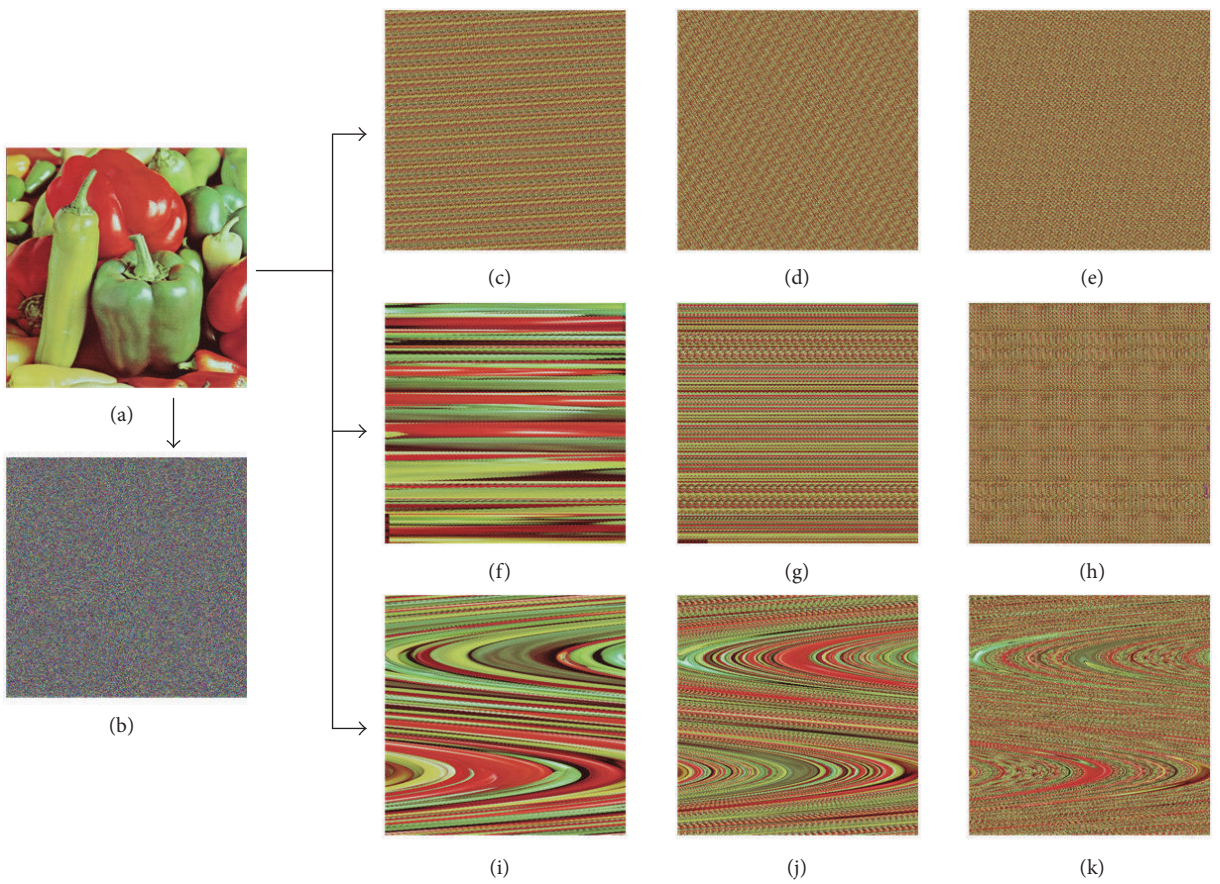


Figure 2: The applications of the proposed and three conventional chaos-based permutation methods. (a) The test image; (b) the test image after applying the proposed permutation method once; (c)–(e), (f)–(h), and (i)–(k) are the test images after applying the cat map, the baker map, and the standard map once, twice, and three times, respectively.

Table 2: Differential images used in NPCR and UACI tests.

Test image name	Color channel	Pixel position (x, y)	Pixel value	
			Original	Modified
Baboon	G	(29, 130)	66	65
House	G	(182, 179)	35	36
Lena	B	(425, 39)	121	122
Peppers	R	(428, 144)	123	122
Portofino	R	(306, 294)	64	65

Table 3: Results of NPCR and UACI tests.

Test image name	Number of encryption rounds							
	1		2		3		4	
	NPCR	UACI	NPCR	UACI	NPCR	UACI	NPCR	UACI
Baboon	0.90905	0.30349	0.99614	0.33460	0.996095	0.33445	0.99609	0.33457
House	0.83127	0.27808	0.99611	0.33483	0.9961407	0.33466	0.99608	0.33446
Lena	0.06218	0.02089	0.99620	0.33469	0.996074	0.33487	0.99619	0.33447
Peppers	0.81662	0.27345	0.99607	0.33448	0.996147	0.33457	0.99608	0.33466
Portofino	0.44400	0.14861	0.99600	0.33443	0.995989	0.33425	0.99603	0.33487

The NPCR and UACI of the proposed cryptosystem are evaluated using five standard 24-bit color test images of size 512×512 taken from the USC-SIPI image database. The differential images are created by randomly changing 1 bit in the original ones, as listed in Table 2. The two images in each test pair are encrypted using the same secret key, and their NPCR and UACI values under different number of cipher rounds are given in Table 3. It can be concluded from Table 3 that the diffusion efficiency of the proposed scheme is competitive with that of existing optimal substitution strategies, which take a minimum of two encryption rounds to achieve desired NPCR and UACI values.

6. Security Analysis

In this section, thorough security analysis has been carried out, including the most important ones like brute-force analysis, statistical analysis, and key sensitivity analysis, to demonstrate the high security of the proposed scheme.

6.1. Brute-Force Analysis. In cryptography, a brute-force attack is a cryptanalytic attack that attempts to break a cipher by systematically checking all possible keys until the correct one is found. Obviously, a cipher with a key length of n bits can be broken in a worst-case time proportional to 2^n and an average time of half that. A key should therefore be long enough that this line of attack is impractical, that is, would take too long to execute. As mentioned above, the initial conditions of the hyperchaotic Lü system and the logistic map, which consist of four and one state variables, are used as the permutation key and substitution keys, respectively. The two keys are independent of each other and a 64-bit double-precision type gives 53 bits of precision, and therefore the key length of the proposed scheme is $5 \times 53 = 265$ bits. Generally, cryptographic algorithms using keys with a length greater than 100 bits are considered to be "computational security" as the number of operations required to try all possible 2^{100} keys is widely considered out of reach for conventional digital computing techniques for the foreseeable future. Therefore, the proposed scheme is secure against brute-force attack.

6.2. Statistical Analysis

6.2.1. Frequency Distribution of Pixel Values. A good image cryptosystem should sufficiently mask the distribution of pixel values in the plain-images so as to make frequency analysis infeasible. That is, the redundancy of plain-image or the relationship between plain-image and cipher-image should not be observed from the cipher-image as such information has the potential to be exploited in a statistical attack. The frequency distribution of pixel values in an image can be easily determined by using histogram analysis. An image histogram is a graph showing the number of pixels in an image at each different intensity value found in that image. The histograms of the RGB color channels of the "peppers" test image and its output cipher-image produced by the proposed scheme are shown in Figure 3. It is clear from Figures 3(l)–3(n) that the pixel values in all the three color channels of the output cipher-image are fairly evenly distributed over the whole intensity range, and therefore no information about the plain-image can be gathered through histogram analysis.

The distribution of pixel values can be further quantitatively determined by calculating the information entropy of the image. Information entropy, introduced by Shannon in his classic paper "A Mathematical Theory of Communication" [29], is a key measure of the randomness or unpredictability of information content. The information entropy is usually expressed by the average number of bits needed to store or communicate one symbol in a message, as described by

$$H(S) = -\sum_{i=1}^{N} P(s_i) \log_2 P(s_i), \quad (17)$$

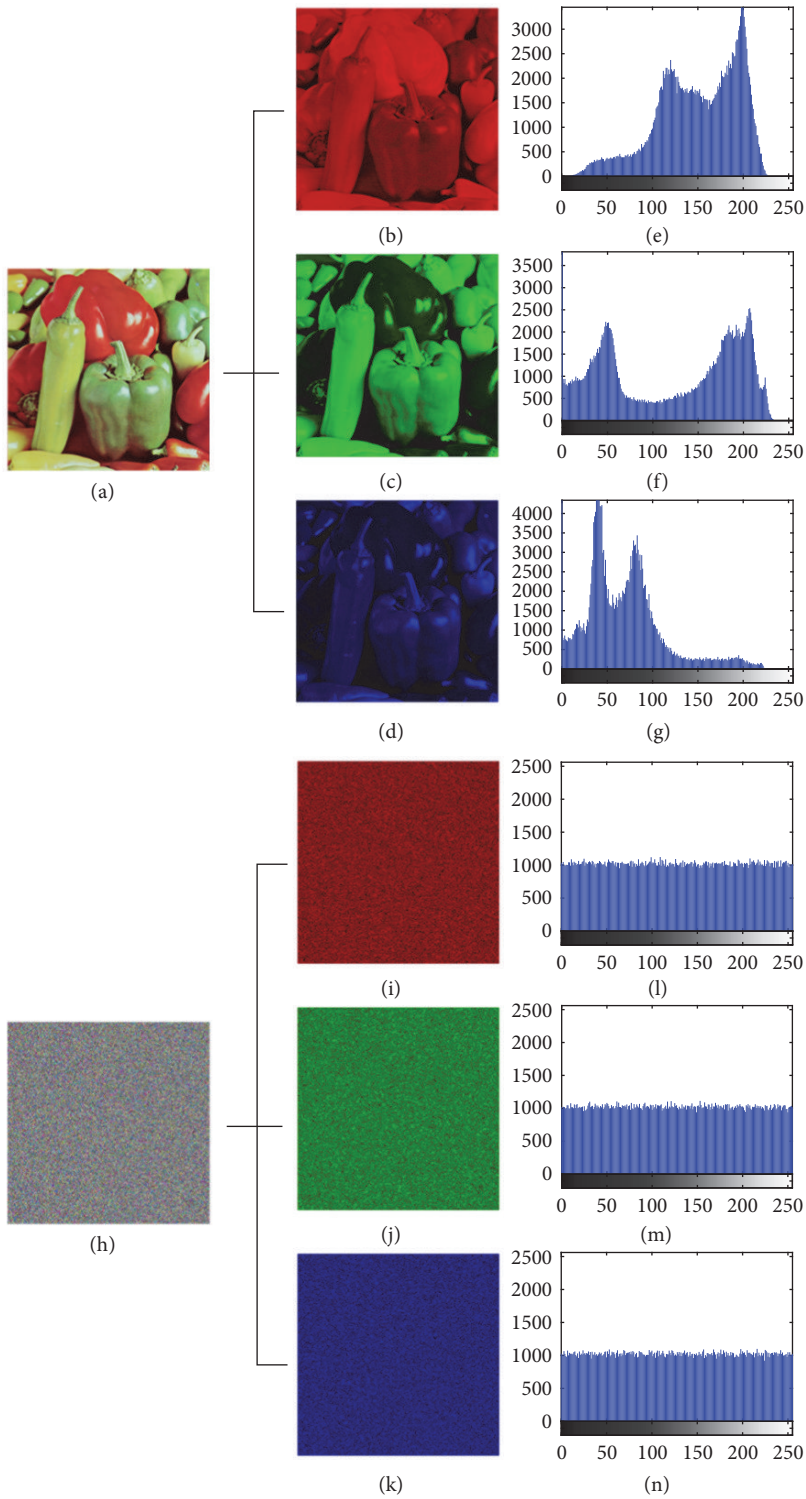


FIGURE 3: Histogram analysis. (a) and (h) are the test image and its output cipher-image, respectively. (b)–(d) and (i)–(k) are the three color channels of (a) and (h), respectively. (e)–(g) and (l)–(n) are the histograms of (b)–(d) and (i)–(k), respectively.

Table 4: Information entropies of the test images and their output cipher-images.

Test image name	Information entropy	
	Plain-image	Cipher-image
Baboon	7.762436	7.999778
House	7.485787	7.999747
Lena	7.750197	7.999772
Peppers	7.669826	7.999788
Portofino	7.306934	7.999766

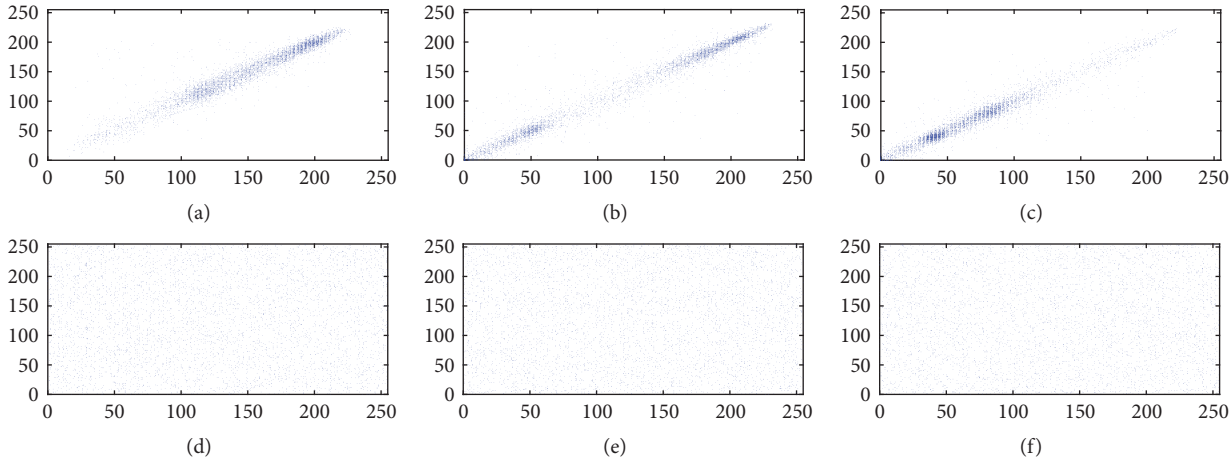


Figure 4: Graphical analysis for correlation of neighboring pixels. (a)–(c) and (d)–(f) are scatter diagrams for horizontally neighboring pixels in the three color channels of the "peppers" test image and its output cipher-image, respectively.

where S is a random variable with N outcomes $\{s1, \ldots, s_N\}$ and $P(s_i)$ is the probability mass function of outcome s_i. It is obvious from (17) that the entropy for a random source emitting N symbols is $\log_2 N$. For instance, for a ciphered image with 256 color levels per channel, the entropy should ideally be 8; otherwise, there exists certain degree of predictability which threatens its security.

The information entropies of the five test images and their output cipher-images are calculated, and the results are listed in Table 4. As can be seen from Table 4, the entropy of all the output cipher-images are very close to the theoretical value of 8. This means the proposed scheme produces outputs with perfect randomness and hence is robust against frequency analysis.

6.2.2. Correlation between Neighboring Pixels. Pixels in an ordinary image are usually highly correlated with their neighbors either in horizontal, vertical, or diagonal direction. However, an effective image cryptosystem should procedure cipher-images with sufficiently low correlation between neighboring pixels. Scatter diagram is commonly used to qualitatively explore the possible relationship between two data sets. To plot a scatter diagram for image data, the following procedures are carried out. First, randomly select S_n pairs of neighboring pixels in each direction from a color channel of the image. Then, the selected pairs is displayed as a collection of points, each having the value of one pixel determining the position on the horizontal axis and the value of the other pixel determining the position on the vertical axis.

Figures 4(a)–4(c) and 4(d)–4(f) show the scatter diagrams for horizontally neighboring pixels in the three color channels of the "peppers" test image and its output cipher-image with $S_n = 5000$, respectively. Similar results can be obtained for the other two directions. As can be seen from this figure, most points in (a)–(c) are clustered around the main diagonal, whereas those in (d)–(f) are fairly evenly distributed. The results indicate that the proposed scheme can effectively eliminate the correlation between neighboring pixels in an original image.

To further quantitatively measure the correlation between neighboring pixels in an image, the correlation coefficients r_{xy} for the sampled pairs are calculated according to the following three formulas:

$$r_{xy} = \frac{(1/S_n)\sum_{i=1}^{S_n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\left((1/S_n)\sum_{i=1}^{S_n}(x_i - \overline{x})^2\right)\left((1/S_n)\sum_{i=1}^{S_n}(y_i - \overline{y})^2\right)}},$$
$$\overline{x} = \frac{1}{S_n}\sum_{i=1}^{S_n} x_i, \quad (18)$$
$$\overline{y} = \frac{1}{S_n}\sum_{i=1}^{S_n} y_i,$$

where x_i and y_i form the ith pair of neighboring pixels.

Table 5 lists the calculated correlation coefficients for neighboring pixels in the three color channels of the five test

Table 5: Correlation coefficients for neighboring pixels in the test images and their output cipher-images.

Test image name	Direction	Plain-image			Cipher-image		
		R	G	B	R	G	B
Baboon	horizontal	0.8719	0.7544	0.8853	−0.0324	−0.0125	−0.0129
	vertical	0.9245	0.8642	0.9140	−0.0142	0.0027	−0.0036
	diagonal	0.8607	0.7324	0.8481	0.0173	0.0207	0.0034
House	horizontal	0.9601	0.9442	0.9619	−0.0055	0.0363	0.0140
	vertical	0.9570	0.9414	0.9688	0.0132	−0.0080	−0.0001
	diagonal	0.9260	0.8959	0.9340	0.0003	−0.0107	−0.0375
Lena	horizontal	0.9892	0.9833	0.9586	0.0033	0.0294	0.0086
	vertical	0.9796	0.9700	0.9357	0.0155	0.0146	−0.0229
	diagonal	0.9690	0.9571	0.9142	0.0158	0.0102	−0.0366
Peppers	horizontal	0.9640	0.9853	0.9698	0.0133	0.0016	−0.0112
	vertical	0.9622	0.9835	0.9691	0.0146	−0.0082	0.0115
	diagonal	0.9545	0.9737	0.9525	0.0008	−0.0255	0.0109
Portofino	horizontal	0.9530	0.9527	0.9209	0.0141	0.0011	−0.0109
	vertical	0.9423	0.8918	0.9119	0.0023	0.0332	0.0120
	diagonal	0.9346	0.8639	0.8954	−0.0020	−0.0048	0.0106

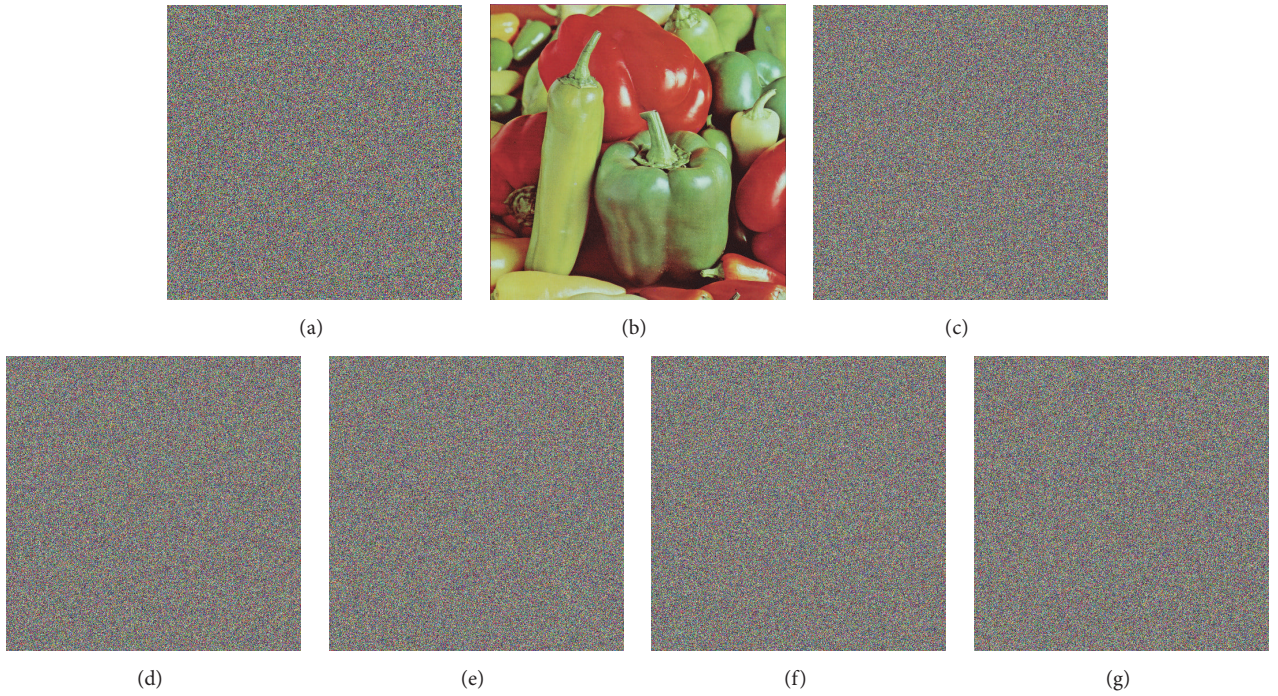


Figure 5: Results of key sensitivity test.

images and their output cipher-images. As can be seen from this table, the correlation coefficients for neighboring pixels in all the three color channels of the output cipher-images are practically zero. The results further support the conclusion drawn from Figure 4.

6.3. Key Sensitivity Analysis. Key sensitivity, another basic design principle of cryptographic algorithms, ensures that no information about the plaintext can be revealed even if there is only a slight difference between the decryption and encryption keys. To evaluate the key sensitivity property of the proposed scheme, the "peppers" test image is firstly encrypted with a randomly generated secret key: hyperchaotic Lü system with initial conditions (x_0 = 9.05791937075619, y_0 = 2.53973632587012, z_0 = 25.2943698490164, u_0 = −28.5802537020945) and logistic map with initial condition x_0 = 0.278498218867048, and the resulting cipher-image is shown in Figure 5(a). Then the ciphered image is tried to be decrypted using six decryption keys, one of which is exactly the same as the encryption key and the other five have only one-bit difference from it, as listed in Table 6. The resulting deciphered images are shown in Figures 5(b)–5(g), respectively, from which we can see that even an almost perfect guess of the key does not reveal any

Table 6: Decryption keys used for key sensitivity test.

Figure	Decryption key		
	Permutation part		Substitution part
5(b)	$x_0 = 9.05791937075619$ $z_0 = 25.2943698490164$	$y_0 = 2.53973632587012$ $u_0 = -28.5802537020945$	$x_0 = 0.278498218867048$
5(c)	$\mathbf{x_0 = 9.05791937075618}$ $z_0 = 25.2943698490164$	$y_0 = 2.53973632587012$ $u_0 = -28.5802537020945$	$x_0 = 0.278498218867048$
5(d)	$x_0 = 9.05791937075619$ $z_0 = 25.2943698490164$	$\mathbf{y_0 = 2.53973632587011}$ $u_0 = -28.5802537020945$	$x_0 = 0.278498218867048$
5(e)	$x_0 = 9.05791937075619$ $\mathbf{z_0 = 25.2943698490163}$	$y_0 = 2.53973632587012$ $u_0 = -28.5802537020945$	$x_0 = 0.278498218867048$
5(f)	$x_0 = 9.05791937075619$ $z_0 = 25.2943698490164$	$y_0 = 2.53973632587012$ $\mathbf{u_0 = -28.5802537020944}$	$x_0 = 0.278498218867048$
5(g)	$x_0 = 9.05791937075619$ $z_0 = 25.2943698490164$	$y_0 = 2.53973632587012$ $u_0 = -28.5802537020945$	$\mathbf{x_0 = 0.278498218867047}$

Table 7: Performance comparison of two schemes using different keystream generation methods.

Image size	File size (KB)	Running speed (ms)	
		Proposed method	Conventional method
256×256	192	27.7	37.0
512×512	768	62.0	77.9
1024×1024	3072	200.9	260.1

information about the original image. It can, therefore, be concluded that the proposed scheme fully satisfies the key sensitivity requirement.

7. Speed Performance

As can be seen from the above description of the substitution keystream generation method, three keystream elements can be simultaneously extracted from the current state of the logistic map, whereas only one can be obtained using the conventional method. As a result, the total number of iterations is reduced by 3 times and the encryption time is shortened. We use three 24-bit RGB test images of different sizes to evaluate the computational efficiency of the proposed scheme and compare it with that of an identical copy of the proposed scheme except using a conventional keystream generation method. Each test image is ciphered 10 times with two rounds of permutation-substitution operations, and the average execution times are listed in Table 7. Both schemes have been implemented using C programming language on Windows 7 64-bit platform, and the tests have been done on a personal computer with an Intel Xeon E3-1230 v3 3.3 GHz processor and 8 GB RAM. As can be seen from Table 7, the proposed scheme significantly outperforms the one using a conventional keystream generation method in terms of computational efficiency, and therefore it provides a good candidate for online secure image transmission over public networks.

8. Conclusions

This paper has proposed a new permutation-substitution type color image cipher to better meet the increasing demand for real-time secure image communications. To confuse the relationship between the ciphertext and the secret key, the positions of colored subpixels in the input image are scrambled using a pixel-swapping mechanism, which avoids two main problems encountered when using the discretized version of area-preserving chaotic maps. To improve the computational efficiency of the substitution process, we introduced an efficient keystream generation method that can simultaneously extract three keystream elements from the current state of the iterative logistic map. Compared with the conventional method, the total number of iterations is reduced by 3 times. The computational efficiency comparison results have shown the superior performance of the proposed encryption scheme. To ensure the robustness of the proposed scheme against chosen-plaintext attack, the current state of the logistic map is perturbed during each iteration and the disturbance value is determined by plain-pixel values. The mechanism of associating the keystream sequence with plain-image also helps accelerate the diffusion process and increase the degree of randomness of the keystream sequence. The results of NPCR and UACI tests indicate that the proposed scheme takes only two encryption rounds to achieve a satisfactory diffusion effect. The results of NIST statistical test indicated that the substitution keystream sequences generated using the proposed method have a higher degree of randomness than that generated by conventional method. We have carried out an extensive security analysis, which demonstrates the satisfactory security level of the new scheme. It can therefore be concluded that the proposed scheme provides a good candidate for online secure image communication applications.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities (no. N150402004) and the Online Education Research Fund of MOE Research Center for Online Education (Qtone Education) (no. 2016YB123).

References

[1] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *International Journal of Bifurcation and Chaos*, vol. 8, no. 6, pp. 1259–1284, 1998.

[2] C. Fu, W.-H. Meng, Y.-F. Zhan et al., "An efficient and secure medical image protection scheme based on chaotic maps," *Computers in Biology and Medicine*, vol. 43, no. 8, pp. 1000–1010, 2013.

[3] J.-X. Chen, Z.-L. Zhu, C. Fu, H. Yu, and Y. Zhang, "Reusing the permutation matrix dynamically for efficient image cryptographic algorithm," *Signal Processing*, vol. 111, pp. 294–307, 2015.

[4] K. Wong, B. S. Kwok, and W. Law, "A fast image encryption scheme based on chaotic standard map," *Physics Letters A*, vol. 372, no. 15, pp. 2645–2652, 2008.

[5] C. Fu, B. Lin, Y. Miao, X. Liu, and J. Chen, "A novel chaos-based bit-level permutation scheme for digital image encryption," *Optics Communications*, vol. 284, no. 23, pp. 5415–5423, 2011.

[6] Y. Wu, Y. Zhou, S. Agaian, and J. P. Noonan, "A symmetric image cipher using wave perturbations," *Signal Processing*, vol. 102, pp. 122–131, 2014.

[7] W. Zhang, H. Yu, Y.-L. Zhao, and Z.-L. Zhu, "Image encryption based on three-dimensional bit matrix permutation," *Signal Processing*, vol. 118, pp. 36–50, 2016.

[8] C. Fu, O. Bian, H.-Y. Jiang, L.-H. Ge, and H.-F. Ma, "A new chaos-based image cipher using a hash function," in *Proceedings of the 15th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2016*, Japan, June 2016.

[9] J. Chen, Z. Zhu, C. Fu, H. Yu, and L. Zhang, "A fast chaos-based image encryption scheme with a dynamic state variables selection mechanism," *Communications in Nonlinear Science and Numerical Simulation*, 2014.

[10] C. Fu, Z.-K. Wen, Z.-L. Zhu, and H. Yu, "A security improved image encryption scheme based on chaotic Baker map and hyperchaotic Lorenz system," *International Journal of Computational Sciences and Engineering*, vol. 12, no. 2-3, pp. 113–123, 2016.

[11] C. Li, S. Li, and K.-T. Lo, "Breaking a modified substitution-diffusion image cipher based on chaotic standard and logistic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 2, pp. 837–843, 2011.

[12] C. Li, L. Y. Zhang, R. Ou, K.-W. Wong, and S. Shu, "Breaking a novel colour image encryption algorithm based on chaos," *Nonlinear Dynamics*, vol. 70, no. 4, pp. 2383–2388, 2012.

[13] C. Li, T. Xie, Q. Liu, and G. Cheng, "Cryptanalyzing image encryption using chaotic logistic map," *Nonlinear Dynamics*, vol. 78, no. 2, pp. 1545–1551, 2014.

[14] C. Li, Y. Liu, L. Y. Zhang, and M. Z. Q. Chen, "Breaking a chaotic image encryption algorithm based on modulo addition and XOR operation," *International Journal of Bifurcation and Chaos*, vol. 23, no. 4, Article ID 1350075, 2013.

[15] Y. Wang, K.-W. Wong, X. Liao, T. Xiang, and G. Chen, "A chaos-based image encryption algorithm with variable control parameters," *Chaos, Solitons & Fractals*, vol. 41, no. 4, pp. 1773–1783, 2009.

[16] C. Fu, J.-B. Huang, N.-N. Wang, Q.-B. Hou, and W.-M. Lei, "A symmetric chaos-based image cipher with an improved bit-level permutation strategy," *Entropy*, vol. 16, no. 2, pp. 770–788, 2014.

[17] T. Xiang, K.-W. Wong, and X. Liao, "Selective image encryption using a spatiotemporal chaotic system," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 17, no. 2, Article ID 023115, 2007.

[18] K.-W. Wong, B. S.-H. Kwok, and C.-H. Yuen, "An efficient diffusion approach for chaos-based image encryption," *Chaos, Solitons & Fractals*, vol. 41, no. 5, pp. 2652–2663, 2009.

[19] J.-X. Chen, Z.-L. Zhu, C. Fu, L.-B. Zhang, and Y. Zhang, "An efficient image encryption scheme using lookup table-based confusion and diffusion," *Nonlinear Dynamics*, vol. 81, no. 3, pp. 1151–1166, 2015.

[20] C. Fu, J. Chen, H. Zou, W. Meng, Y. Zhan, and Y. Yu, "A chaos-based digital image encryption scheme with an improved diffusion strategy," *Optics Express*, vol. 20, no. 3, pp. 2363–2378, 2012.

[21] Z. Zhu, W. Zhang, K. Wong, and H. Yu, "A chaos-based symmetric image encryption scheme using a bit-level permutation," *Information Sciences*, vol. 181, no. 6, pp. 1171–1186, 2011.

[22] J.-x. Chen, Z.-l. Zhu, C. Fu, L.-b. Zhang, and Y. Zhang, "An image encryption scheme using nonlinear inter-pixel computing and swapping based permutation approach," *Communications in Nonlinear Science and Numerical Simulation*, vol. 23, no. 1-3, pp. 294–310, 2015.

[23] J. Chen, Z.-L. Zhu, L.-B. Zhang, Y. Zhang, and B.-Q. Yang, "Exploiting self-adaptive permutation–diffusion and DNA random encoding for secure and efficient image encryption," *Signal Processing*, vol. 142, pp. 340–353, 2018.

[24] J.-X. Chen, Z.-L. Zhu, C. Fu, H. Yu, and L.-B. Zhang, "An efficient image encryption scheme using gray code based permutation approach," *Optics and Lasers in Engineering*, vol. 67, pp. 191–204, 2015.

[25] Z. Hua, Y. Zhou, C.-M. Pun, and C. L. P. Chen, "2D Sine Logistic modulation map for image encryption," *Information Sciences*, vol. 297, pp. 80–94, 2015.

[26] A. Chen, J. Lu, J. Lü, and S. Yu, "Generating hyperchaotic Lü attractor via state feedback control," *Physica A: Statistical Mechanics and its Applications*, vol. 364, pp. 103–110, 2006.

[27] J. Lü, G. Chen, and S. Zhang, "Dynamical analysis of a new chaotic attractor," *International Journal of Bifurcation and Chaos*, vol. 12, no. 5, pp. 1001–1015, 2002.

[28] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, no. 5560, pp. 459–467, 1976.

[29] C. E. Shannon, "A mathematical theory of communication," *Bibliometrics*, vol. 5, no. 1, pp. 3–55, 2001.

Hindawi
Security and Communication Networks
Volume 2018, Article ID 9803519, 12 pages
https://doi.org/10.1155/2018/9803519

Research Article

Meaningful Image Encryption Based on Reversible Data Hiding in Compressive Sensing Domain

Ming Li[1,2], **Haiju Fan**[1,3], **Hua Ren**[1], **Dandan Lu**[1], **Di Xiao**[4] and **Yang Li**[2,5,6]

[1] *College of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China*
[2] *School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China*
[3] *China National Digital Switching System Engineering and Technological R&D Center, Zhengzhou 450002, China*
[4] *College of Computer Science, Chongqing University, Chongqing 400044, China*
[5] *Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing 100191, China*
[6] *Beijing Advanced Innovation Center for Big Date-Based Precision Medicine, Beihang University, Beijing 100083, China*

Correspondence should be addressed to Yang Li; liyang@buaa.edu.cn

Received 24 July 2017; Revised 10 October 2017; Accepted 16 January 2018; Published 14 February 2018

Academic Editor: Bruce M. Kapron

A novel method of meaningful image encryption is proposed in this paper. A secret image is encrypted into another meaningful image using the algorithm of reversible data hiding (RDH). High covertness can be ensured during the communication, and the possibility of being attacked of the secret image would be reduced to a very low level. The key innovation of the proposed method is that RDH is applied to compressive sensing (CS) domain, which brings a variety of benefits in terms of image sampling, communication and security. The secret image after preliminary encryption is embedded into the sparse representation coefficients of the host image with the help of the dictionary. The embedding rate could reach 2 bpp, which is significantly higher than those of other state-of-art schemes. In addition, the computational complexity of receiver is reduced. Simulations verify our proposal.

1. Introduction

Along with the development of Internet and information science, Internet can definitely provide great convenience to information transmission and processing, but as the same time, severe security problems therein have emerged gradually [1, 2]. Data hiding, as an important method to guarantee secure information transmission, has attracted more and more people's concern [3, 4]. It has solved the easily-being-attacked problems of Cryptosystem by hiding the existence of data. Most of the methods, however, arbitrarily alter the carrier of information, which result in an unrecoverable host image after the secret information has been extracted by the receiver, and the embedded information may also not be recovered exactly. This is vital in some distortion-unacceptable scenarios. Subsequently, the problem has been solved effectively by the reversible data hiding (RDH) [5–7]. RDH technique may hide the secret messages in digital images, and it has two merits: lossless restoration of host image and lossless extraction of embedded massage. Consequently, RDH has become one effective technique to achieve copyright protection for the host image and covert communication for the embedded message [8–10].

Image encryption is a well-known technique to protect the contents of the original images [11–13]. The encrypted result appears as a random image, whose histogram is flat and information entropy is close to 8. Therefore, a secure cryptographic system with large key space and effective encryption mechanism can resist unauthorized user to obtain the original image. But the noise-like image is easily detected and attacked by the hacker during the transmission. If the encryption result presents as a meaningful image, it would be secure visually, and the possibility of being attacked would be dropped down to a very low level. Recently, many researchers contribute to the visually secure encryption scheme [14, 15]. For example, Kanso and Ghebleh [14] first decomposed the host image into four sub-bands denoted by LL, HL, LH and HH, then embedded a scrambled image into the latter three sub-bands. Chai et al. [15] proposed to compress the shuffled plain image and embed it into the host image. In fact, the

method to encrypt a plain image into another meaningful image employs the method of data hiding; therefore they have used many common algorithms. Among them, RDH based on lossless compression [6, 16], RDH using difference expansion [17, 18] and RDH using histogram shifting [19, 20] are the most influential methods. It is noticed that, if the secret image is encrypted into a meaningful image, the aim of image encryption (another kind of RDH) is not to protect the host image any more, but to protect the embedded secret image. Therefore, meaningful image encryption can be used for covert communication of the secret image.

Compressive sensing (CS), which broke through the Shannon-Nyquist sampling theorem, brings a huge impact on the area of communication. Recently, CS and sparse representation are applied in RDH [15, 21–25]. Xiao et al. [22] embedded the secret data into the AC coefficient of host image and then used compressive sensing to generate the encrypted image. Yamaç et al. [23] proposed to embed the data into the CS measurements. Hua et al. [24] have proposed an informed sparse data hiding system, which embeds the message in the sparse domain and relaxes the sparsity condition. Meanwhile, many researchers tested that CS based RDH is secure to resist multiple attacks, because the measurement matrix or dictionary in CS based RDH is set to be secret. Rachlin and Baron [26] confirmed that it is not possible to reconstruct the signal using only the measurements without knowing the measurement matrix or dictionary. Orsdemir et al. [27] argued that brute attacks to obtain the matrix above were computationally infeasible. Therefore, processing the host image by CS effectively enhanced the security of RDH. In addition, it also broadened the applications of RDH. However, RDH in CS domain is still an attempt, and the computation complexities and the embedding capacities in the existing methods are still not satisfactory.

In this paper, enlightened by the idea of encrypting a secret image into another meaningful image which can significantly improve the covertness of the secret image communication, we propose a novel method of meaningful image encryption by combining RDH with CS. The host image is processed by CS, and the key problem of the proposed method is the achievement of RDH in CS domain. The contributions of this paper can be summarized as follows:

(1) The covertness of the secret image to be encrypted can be ensured by RDH in a meaningful image.

(2) A novel method of RDH in CS domain is proposed.

(3) The embedding rate is significantly higher compared with the state-of-art schemes.

(4) The complexity of computation of receiver is reduced.

The rest of this paper is organized as follows. Section 2 gives the related works about CS. Our proposed encryption scheme is described in Section 3. Section 4 verifies the proposed scheme via simulations. Comparisons between the proposed scheme and other works are discussed in Section 5. And the last section makes a conclusion of our work.

2. The Framework of CS

In this section, we discuss the basic CS theory and the exact recovery condition. Because the host images are usually natural images, they can be compressed by CS theory. Let a host image and its sparse solution be $\mathbf{X} \in \mathbb{R}^{M\times 1}$ and $\mathbf{Y} \in \mathbb{R}^{N\times 1}$ respectively, then

$$\begin{aligned} \widehat{\mathbf{Y}} = \arg\min_{\mathbf{Y}} \quad & \|\mathbf{Y}\|_0 \\ \text{s.t.} \quad & \mathbf{X} = \mathbf{D}\mathbf{Y}, \end{aligned} \tag{1}$$

where $\mathbf{D} \in \mathbb{R}^{M\times N}$ is a matrix with full row rank. If we want to reconstruct $\mathbf{Y}$ from $\mathbf{X}$, the mutual incoherence property (RIP) [28] must be obeyed, which is showed as follows

$$\mu \triangleq \max_{i\neq j} \left|\left\langle \mathbf{d}_i, \mathbf{d}_j \right\rangle\right|, \quad i, j = 0, 1, \ldots, N-1, \tag{2}$$

where $\mathbf{d}_i$ and $\mathbf{d}_j$ are the ith and jth column vectors of $\mathbf{D}$ respectively. Tropp [28] has described the condition to reconstruct $\mathbf{Y}$ exactly as follow

$$K \triangleq \operatorname{card}\left(\widehat{\mathbf{Y}}\right) < \frac{\mu + 1}{2\mu}. \tag{3}$$

This condition requires that the host image must be sparse enough, which is impossible for natural image. Hua et al. [24] proposed that for RDH this condition in (3) can be relaxed as

$$K < M. \tag{4}$$

In the next section, we will propose how to maximize the embedding capacity by employing this condition above.

3. Description of Our Proposed Scheme

3.1. The Embedding Process. This section presents the proposed embedding algorithm, whose flowchart is shown in Figure 1. As in Figure 1, the embedding algorithm consists of four phases. In the first phase, a plain image can be encrypted by any of the existing cryptosystem. Next, the host image is represented sparsely by a given dictionary. Then, the encrypted plain image is embedded into the sparse coefficients of the host image. At last, the coefficients with message multiply with the new dictionary to generate the final cipher image, which is a meaningful image with visual security.

Step 1 (encrypt the plain image). Let a plain image and its cipher image in the first phase be $\mathbf{P} = \{P_{ij}\}_{i=1,j=1}^{m,n}$ and $\mathbf{C} = \{C_{ij}\}_{i=1,j=1}^{m,n}$. And the cipher image $\mathbf{C}$ is called Cipher image 1 in this paper. The plain image can be encrypted by any existing cryptosystem, whose keys are denoted by $A_1, A_2, \ldots, A_T$ in Figure 1. Then we can obtain a vector image $\mathbf{c} \in \mathbb{R}^{mn\times 1}$ of $\mathbf{C}$.

Step 2 (represent the host image sparsely). To compress the host image, we must choose a dictionary $\mathbf{D}$ such as DCT dictionary, wavelet dictionary, noiselets dictionary and so on, which can also be obtained from dictionary learning. Let $\mathbf{H} \in \mathbb{R}^{M\times N}$ be a host image, whose vectorized version is $\mathbf{h}$

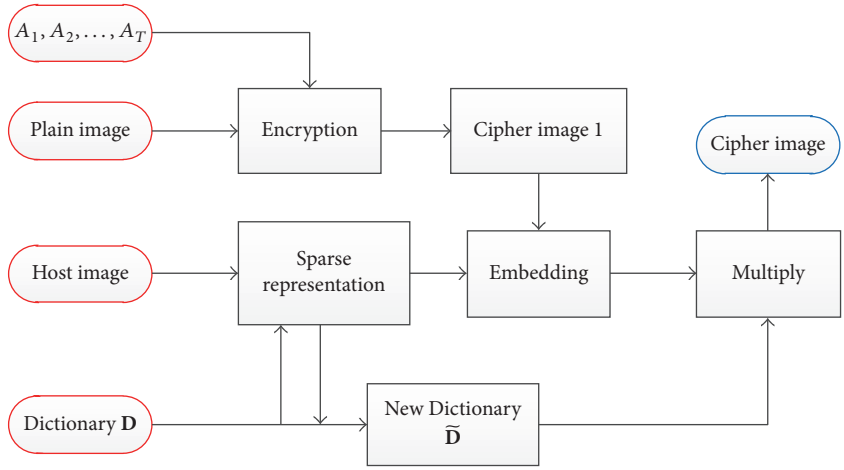


Figure 1: The flowchart of the proposed embedding method.

of length MN. The host vector $\mathbf{h}$ can be represented sparsely by $\mathbf{D} \triangleq [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_{MN}] \in \mathbb{R}^{MN \times MN}$ as

$$\dot{\mathbf{Y}} = \arg\min_{\mathbf{Y}} \quad \|\mathbf{Y}\|_0 \qquad \text{s.t.} \quad \mathbf{h} = \mathbf{D}\mathbf{Y}, \tag{5}$$

where $\dot{\mathbf{Y}} \triangleq [y_1, y_2, \ldots, y_{MN}]^T \in \mathbb{R}^{MN \times 1}$. Let the sparsity of $\mathbf{h}$ be K, then $\dot{\mathbf{Y}}$ can be obtained by orthogonal matching pursuit algorithm with the fixed sparsity K. Therefore, the vector $\dot{\mathbf{Y}}$ would have K nonzero elements. The indices of those nonzero elements are denoted by $\mathbf{k} \triangleq [k_1, k_2, \ldots, k_K]$. If selecting the K nonzero elements in order, we can get a vector $\ddot{\mathbf{Y}} \triangleq [\dot{y}_{k_1}, \dot{y}_{k_2}, \ldots, \dot{y}_{k_K}]^T \in \mathbb{R}^{K \times 1}$ satisfying

$$\dot{\mathbf{h}} = \mathbf{\Phi}\ddot{\mathbf{Y}}, \tag{6}$$

where $\mathbf{\Phi} \triangleq [\mathbf{d}_{k_1}, \mathbf{d}_{k_2}, \ldots, \mathbf{d}_{k_K}] \in \mathbb{R}^{MN \times K}$ and $\dot{\mathbf{h}}$ is a lossy equivalent of $\mathbf{h}$.

Step 3 (embed the message into the coefficients vector of host image). Let the null space of $\mathbf{\Phi}^T$ be $\mathbf{\Delta} \in \mathbb{R}^{MN \times (MN-K)}$, which satisfying $\mathbf{\Phi}^T\mathbf{\Delta} = 0$ and $\mathbf{\Delta}^T\mathbf{\Delta} = \mathbf{I}$. We can first construct a new dictionary $\widetilde{\mathbf{D}} = [\mathbf{\Phi}, \mathbf{\Delta}] \in \mathbb{R}^{MN \times MN}$. Then the Cipher image 1 is embedded in coefficients vector, which is depicted by Algorithm 1.

In Algorithm 1, the function "dec2bin()" converts the decimal number to a binary number and the function "bin2dec()" is opposite. The function "getbits($\mathbf{c}_b, i : j$)" gets the bits of $\mathbf{c}_b$ from the jth bit to the ith bit and constructs a binary number. The typical workflows are introduced as follows. First, by Line (4), each pixel of Cipher image 1 is represented by 8-bit binary number. Second, by Line (5)–(7), each 8-bit binary number is broken up into three parts and transformed to three decimal numbers $\mathbf{c}_3$, $\mathbf{c}_2$ and $\mathbf{c}_1$. These decimal numbers should be small enough to obtain a meaningful cipher image but large enough to eliminate the effect of quantization. Third, we propose parity checks

(1) **Input:** the Cipher image 1 $\mathbf{c}$, sparse coefficients $\ddot{\mathbf{Y}}$
(2) $\mathbf{c}_3 = []$; $\mathbf{c}_2 = []$; $\mathbf{c}_1 = []$;
(3) *for* $i = 1 : mn$
(4) $\quad \mathbf{c}_b = \text{dec2bin}(\mathbf{c}(i))$;
(5) $\quad s_3 = \text{bin2dec}(\text{getbits}(\mathbf{c}_b, 8 : 7))$;
(6) $\quad s_2 = \text{bin2dec}(\text{getbits}(\mathbf{c}_b, 6 : 4))$;
(7) $\quad s_1 = \text{bin2dec}(\text{getbits}(\mathbf{c}_b, 3 : 1))$;
(8) $\quad$ for $j = 1 : 3$
(9) $\quad\quad s_j = s_j + 1$;
(10) $\quad\quad$ if $(\text{mod}(s_j, 2) == 1)$
(11) $\quad\quad\quad s_j = s_j * (-1)$;
(12) $\quad\quad$ end
(13) $\quad$ end
(14) $\quad \mathbf{c}_3 = [\mathbf{c}_3; s_3]$; $\mathbf{c}_2 = [\mathbf{c}_2; s_2]$; $\mathbf{c}_1 = [\mathbf{c}_1; s_1]$;
(15) end
(16) **Output:** $\widetilde{\mathbf{Y}} = [\ddot{\mathbf{Y}}, \mathbf{c}_3, \mathbf{c}_2, \mathbf{c}_1]$

Algorithm 1: Embed the Cipher image 1 into coefficients vector.

(Line (10)–(12)) to reduce the quantization error, where even numbers are set to be positive and odd numbers are negative. What's more, due to the fact that zero is too small to reverse the positive and negative, we add 1 to s_j (Line (9)) to avoid the appearance of zero. Last, each group of 8-bit of Cipher image 1 is assigned to three locations of $\widetilde{\mathbf{Y}}$ for output (Line (16)). Meanwhile, the length mn of $\mathbf{c}$ must equal to $(MN - K)/3$.

Step 4 (generate a meaningful image to transmit). A vector $\mathbf{x} \in \mathbb{R}^{MN \times 1}$ can be generated by the following equation

$$\mathbf{x} = \widetilde{\mathbf{D}}\widetilde{\mathbf{Y}} = [\mathbf{\Phi}, \mathbf{\Delta}] \left[\ddot{\mathbf{Y}}, \mathbf{c}_3, \mathbf{c}_2, \mathbf{c}_1\right]^T = \mathbf{\Phi}\ddot{\mathbf{Y}} + \mathbf{\Delta}\widetilde{\mathbf{c}}, \tag{7}$$

where $\widetilde{\mathbf{c}} = [\mathbf{c}_3, \mathbf{c}_2, \mathbf{c}_1]^T \in \mathbb{R}^{(MN-K) \times 1}$.

The meaningful image $\mathbf{X}$ to transmit can be obtained by reshape the vector $\mathbf{x}$. Moreover, because the elements in

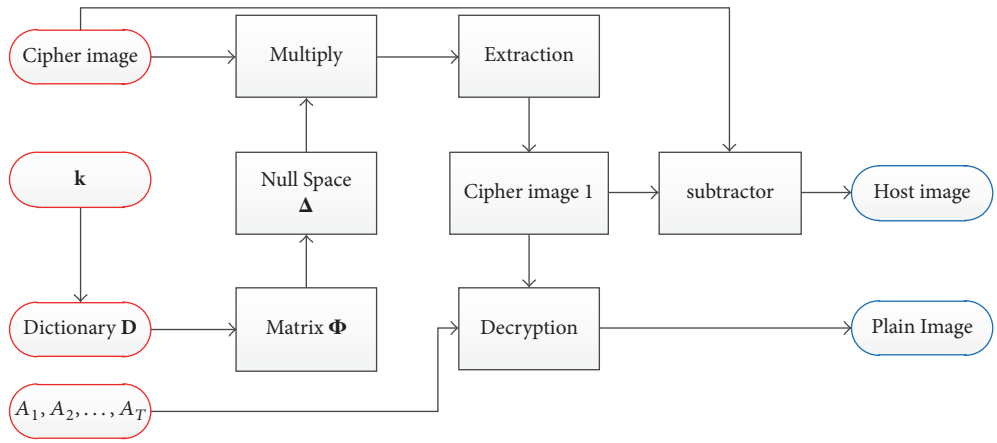


Figure 2: The flowchart of the proposed extraction and reconstruction method.

vector $\tilde{\mathbf{c}}$ are much smaller than the minimum value of $\ddot{\mathbf{Y}}$, it has only a small effect on the transmitted image

$$\mathbf{x} \doteq \mathbf{\Phi}\ddot{\mathbf{Y}} = \dot{\mathbf{h}}. \tag{8}$$

The image quality to transmit depends on the sparsity K. The bigger the sparsity K, the nearer the image approximates to the host image.

3.2. The Extraction and Reconstruction Process. This section presents the proposed extraction and reconstruction algorithm, which is shown in Figure 2. As shown in Figure 2, this algorithm includes two phases. The first phase is to obtain the plain image by extracting the Cipher image 1 and decrypting it. The second phase is to recover the host image.

Step 1 (generate the matrix $\mathbf{\Phi}$ and its null space $\mathbf{\Delta}$). Based on the vector $\mathbf{k}$ and dictionary $\mathbf{D}$, we can construct the matrix $\mathbf{\Phi} \triangleq [\mathbf{d}_{k_1}, \mathbf{d}_{k_2}, \ldots, \mathbf{d}_{k_K}]$ and compute the null space matrix $\mathbf{\Delta}$.

Step 2 (extract the plain image).

Sub-Step 1. Obtain the Cipher image 1 $\hat{\mathbf{c}}$ by the following Equation

$$\hat{\mathbf{c}} = \mathbf{\Delta}^T\mathbf{x}_r = \mathbf{\Delta}^T\left(\mathbf{\Phi}\ddot{\mathbf{Y}} + \mathbf{\Delta}\tilde{\mathbf{c}}\right) = \mathbf{\Delta}^T\mathbf{\Delta}\tilde{\mathbf{c}} = \tilde{\mathbf{c}}, \tag{9}$$

where $\mathbf{x}_r$ is the received cipher image vector. In application, quantification causes that the difference between vectors $\hat{\mathbf{c}}$ and $\tilde{\mathbf{c}}$ is 1. But, this quantification error can be eliminated by our parity checks strategy, which can have a great promoting effect on recovering the embedded image exactly. The extraction flows of the Cipher image 1 are shown in Algorithm 2.

Algorithm 2 achieves the recovery of $\tilde{\mathbf{c}}$ precisely, which is the inverse of Algorithm 1. In Algorithm 2, the function "fix()" takes out the integer part of a number, the function "sign()" takes out a positive or negative sign of a number and the function "strcat()" concatenates strings. The main flows are discussed as follows. First, by Line (4)–(16), the sign of $\tilde{\mathbf{c}}$ is extracted and its parity is determined to reduce the quantization errors, which correspond to Line (10)–(12) of Algorithm 1. Second, by Line (17), we have all elements of $\tilde{\tilde{\mathbf{c}}}$ minus 1, which correspond Line (9) of Algorithm 1. Third, by Line (18)–(20), $\tilde{\tilde{\mathbf{c}}}$ is divided into three parts (high, middle and low bit vectors), which correspond to Line (4)–(7) of Algorithm 1. Last, by Line (21)–(27), each cipher pixel is extracted, which correspond to Line (4)–(7) of Algorithm 1.

(1) **Input:** the received Cipher image $\mathbf{x}_r$, the sparsity K
(2) Generate $\mathbf{\Phi}$ and $\mathbf{\Delta}$
(3) $\mathbf{\Delta}^T\mathbf{x}_r = \mathbf{\Delta}^T(\mathbf{\Phi}\ddot{\mathbf{Y}} + \mathbf{\Delta}\tilde{\mathbf{c}}) = \mathbf{\Delta}^T\mathbf{\Delta}\tilde{\mathbf{c}} = \tilde{\mathbf{c}}$
(4) *for* $i = 1 : mn - K$
(5) $v = \text{fix}(\tilde{\mathbf{c}}(i))$;
(6) $s = \text{sign}(\tilde{\mathbf{c}}(i))$;
(7) if $(\text{mod}(v, 2) == 0)$
(8) $sv = 1$;
(9) else
(10) $sv = -1$;
(11) end
(12) if $(s * sv == -1)$
(13) $v = v + 1$;
(14) end
(15) $\tilde{\tilde{\mathbf{c}}}(i) = v$;
(16) end
(17) $\tilde{\tilde{\mathbf{c}}} = \tilde{\tilde{\mathbf{c}}} - 1$
(18) $\mathbf{c}_3 = \tilde{\tilde{\mathbf{c}}}(1 : (MN - K)/3)$
(19) $\mathbf{c}_2 = \tilde{\tilde{\mathbf{c}}}((MN - K)/3 + 1 : (MN - K)/3 * 2)$
(20) $\mathbf{c}_1 = \tilde{\tilde{\mathbf{c}}}((MN - K)/3 * 2 + 1 : MN - K)$
(21) *for* $i = 1 : (MN - K)/3$
(22) $s3 = \text{dec2bin}(\mathbf{c}_3(i))$;
(23) $s2 = \text{dec2bin}(\mathbf{c}_2(i))$;
(24) $s1 = \text{dec2bin}(\mathbf{c}_1(i))$;
(25) $s = \text{bin2dec}(\text{strcat}(s3, s2, s1))$;
(26) $\hat{\mathbf{c}}(i) = s$;
(27) end
(28) **Output:** $\hat{\mathbf{c}}$ and $\tilde{\mathbf{c}}$

Algorithm 2: Extract the Cipher image 1.

Sub-Step 2. Decrypt the Cipher image 1 $\hat{\mathbf{c}}$ and obtain the plain image.

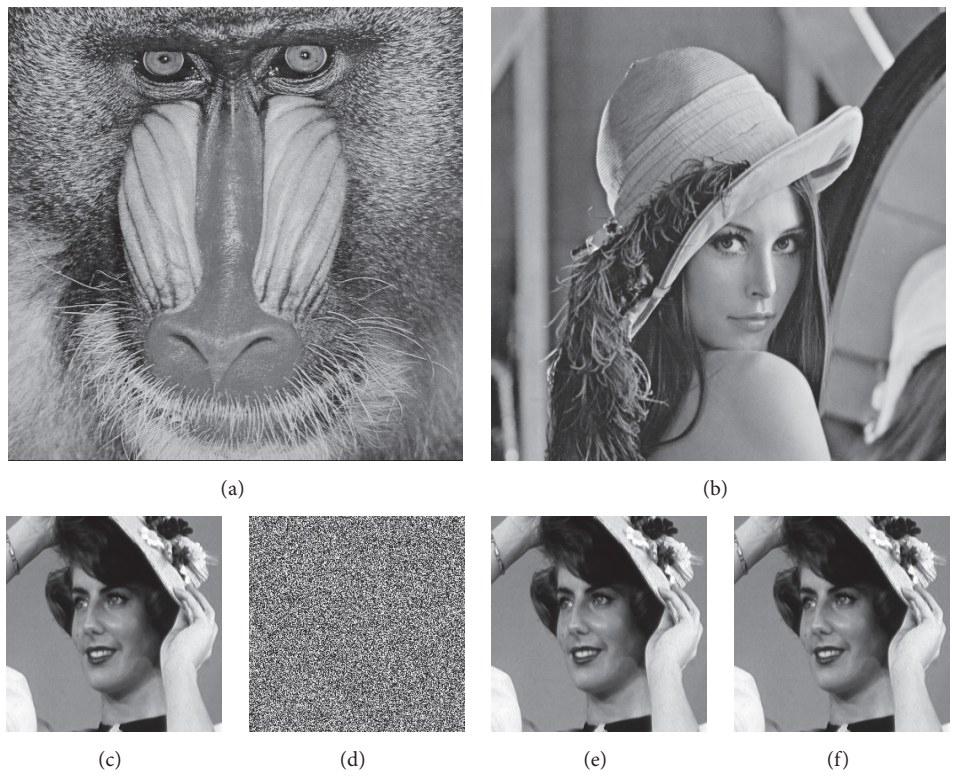


Figure 3: Host images and encryption and decryption results of the plain image Lady: (a) host image Baboon; (b) host image Lena; (c) plain image Lady; (d) Cipher image 1 of (c); (e) the extracted image; (f) the extracted image.

Step 3 (recover the host image).

$$\widehat{\mathbf{h}} = \mathbf{x}_r - \Delta\widetilde{\mathbf{c}} = \mathbf{\Phi}\ddot{\mathbf{Y}} = \dot{\mathbf{h}}. \tag{10}$$

4. Simulation Results

In this section, we verify our proposed embedding and extraction algorithm by experimental results. Without loss of generality, all host images and embedded images are with size 512×512 and 256×256, respectively. Diaconu's encryption algorithm [29] is chosen to encrypt the plain image. The Dictionary we choose is DCT dictionary.

4.1. The Performance of Extraction and Reconstruction. We choose two host images (Baboon and Lena) and one plain image (Lady) to discuss, which are shown as Figures 3(a), 3(b) and 3(c). Figure 3(c) is encrypted by Diaconu's encryption algorithm [29] and the Cipher image 1 is shown in Figure 3(d). Then, Figure 3(d) is embedded in Figures 3(a) and 3(b) respectively. The embedding results are two cipher images Figures 4(a) and 5(a), which are visually secure. Figures 4(b) and 5(b) are the recovered host images from Figures 4(a) and 5(a). Only through our visual sense, we cannot distinguish the differences among the host image, the cipher image and the recovered host image.

Figure 4(c) is the difference image between Figures 3(a) and 4(b), and Figure 5(c) is the difference image between Figures 3(b) and 5(b). From Figures 4(c) and 5(c), we can conclude that there is very tiny difference between the original host image and its recovered version. One of the applications of RDH is covertly communication, which aims to protect the embedded secret image, then, little distortion of the host image is acceptable in this scenario. To discrmminate more about the differences among the host image, the cipher image and the recovered one, we draw their histograms. Figures 4(d)-4(e) are the histograms of Figures 3(a), 4(a) and 4(b) and Figures 5(d)-5(e) are the histograms of Figures 3(b), 5(a) and 5(b). Because Lena has more grey center of clustering than Baboon, its histograms Figures 5(d)-5(e) are even more telling. From Figures 5(d)-5(e), especially the the labeled regions, we can see that the recovered image is closer to the original host image than the cipher image with message, which varifies the perfermance of our proposed algorithm to recover the host image.

The plain images embedded in two host images can be extracted precisely, which are shown in Figures 3(e) and 3(f). One can see that the extracted image is exactly the same as Figure 3(a), indicating that the secret image can be covertly transmitted without any distortion.

4.2. The Embedding Rate. From Algorithms 1 and 2, the embedding rate can be denoted by

$$\text{er} = \frac{8}{3}\frac{MN - K}{MN}. \tag{11}$$

We explain the concluded embedding rate as follows. Supposing that a host image has MN pixles and K sparsity, there are

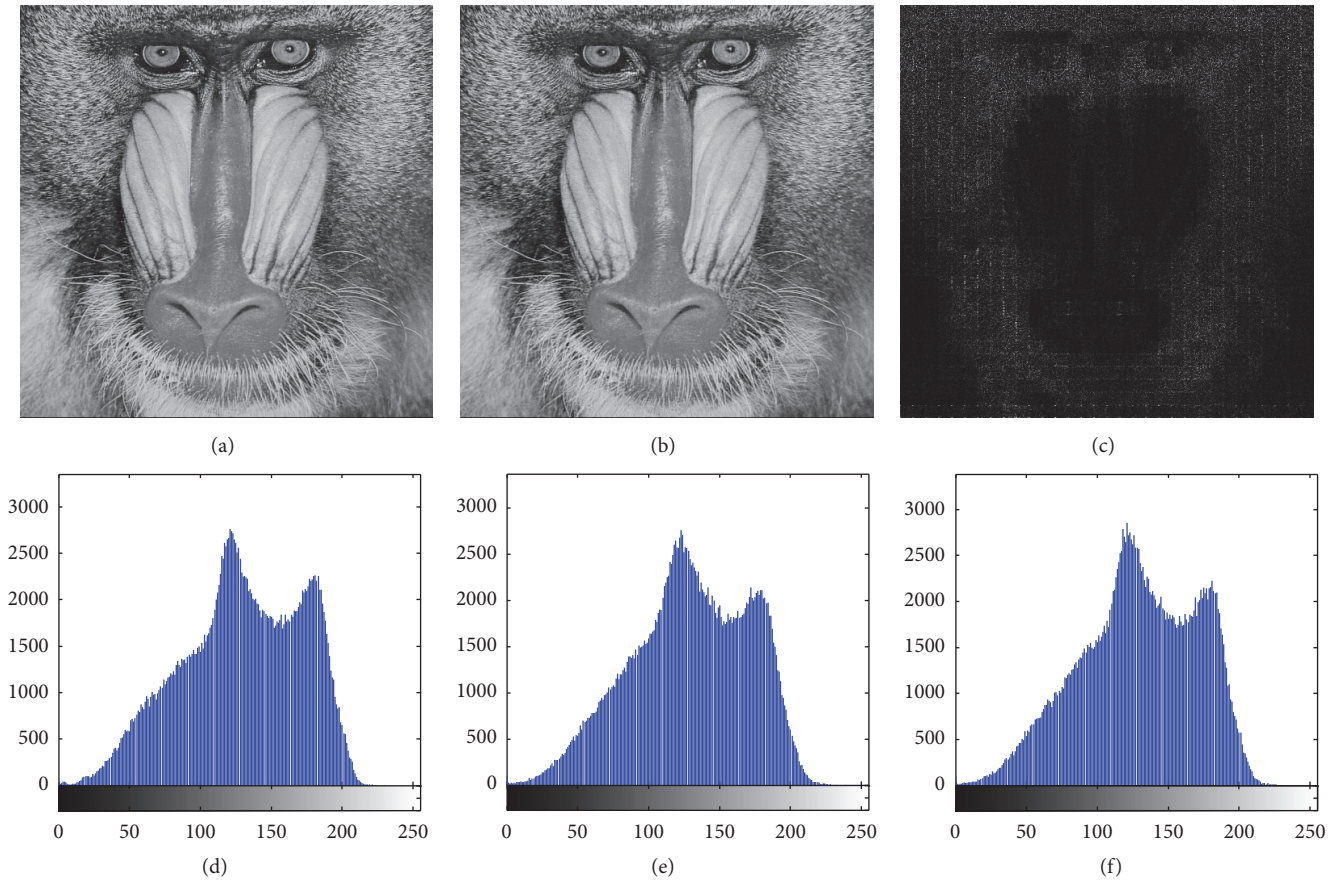


FIGURE 4: The cipher image Baboon and its reconstruction: (a) the cipher image containing message; (b) the recovered image; (c) the difference image between the original and recovered host image; (d–f) the histograms of Figures 3(a), 4(a) and 4(b).

$MN - K$ locations in coefficients vector $\widetilde{\mathbf{Y}}$ to embed message. The length mn of Cipher image 1 must equal to $(MN - K)/3$ as shown in Algorithm 1. Considering that each pixel is 8-bit, the Eq. (11) can be obtained.

By our thoroughly experiments, the embedding rate 2 bpp is a proper rate to balance the embedding capacity and reconstruction performance. If we want to achieve 2 bpp embedding rate, the compressive ratio $K/MN = 0.25$. In our experiments, for the host image sized 512×512 and er = 2 bpp, there are $MN - K = 3MN/4$ locations to embed message. For each pixel of Cipher image 1 is assigned to 3 locations, the total pixels of additional plain image should be $MN/4$. Therefore, we choose a plain image sized 256×256 to verify our algorithm as shown in Figure 3.

If we want to improve the quality of the recovered image, the parameter K can be increased. Two indexes, Peak signal to Noise Ratio (PSNR) and Structure Similarity (SSIM), are suitable to measure the quality of the recovered host image, which are shown in Eqs. (12) and (13)

$$\text{PSNR} = 10 \log_{10} \left(\frac{255^2 M^2 N^2}{\sum_{i=1}^{M^2 N^2} (H_{oi} - H_{ri})} \right), \tag{12}$$

where H_{oi} and H_{ri} denote the ith pixels of the original and recovered host images respectively

$$\text{SSIM} = \frac{(2\mu_x \mu_y + C_1)(2\sigma_x \sigma_y + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{13}$$

where μ_x and μ_y are mean values of two host images, σ_x and σ_y are standard deviations, C_1 and C_2 are two adjust parameters. For PSNR, a higher value is better; For SSIM, the nearer a value approximates to 1, the better the recovered image is.

Figure 6 shows that the PSNRs of the recovered host image increase with the raising of compressive ratio $(K/(MN))$ for three test host images, i.e., Lena, Pepper and Baboon. Meanwhile, different images have different reconstruction performances by using a fixed dictionary. For example, the smooth image Lena is more suitable for embedding data while the textured image Baboon is not. Table 1 gives the SSIMs of the recovered host images with the compressive ratios ranging in $[0.25, 0.75]$, where three images Lena, Pepper and Baoon are chosen as host images. From Table 1, one can see that the SSIMs are all close to 1, which show our proposed algorithm has good performance to recover host image. Since the closer to 1 the SSIM the better the reconstruction is, based on the Table 1 we can see Baboon

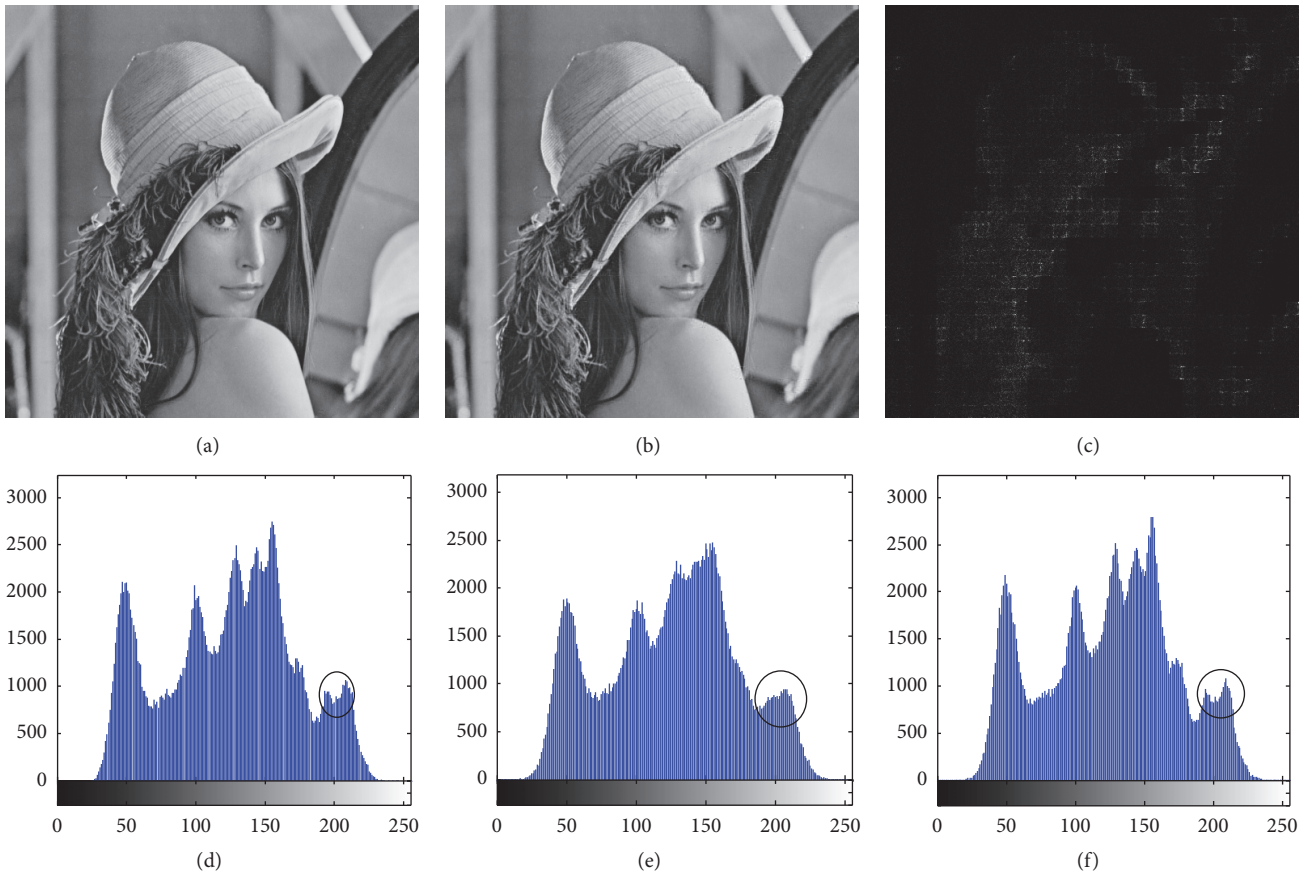


Figure 5: The cipher image Lena and its reconstruction: (a) the cipher image containing message; (b) the recovered image; (c) the difference image between the original and recovered host image; (d–f) the histograms of Figures 3(b), 5(a) and 5(b).

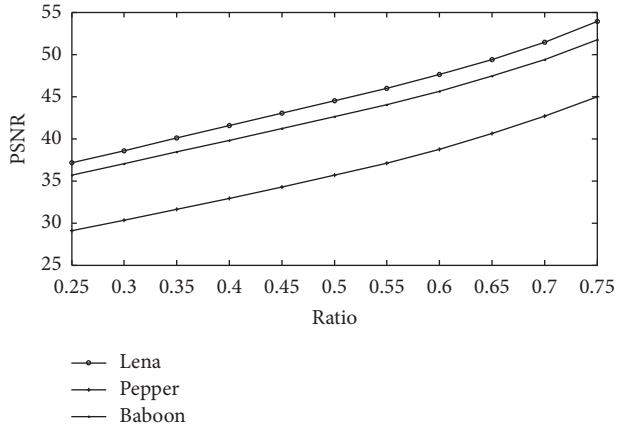


Figure 6: The PSNR VS the compressive ratio for three host images.

is the least suited to embedding data and Lena is the most suited one. The analysises from Table 1 are consistant with those from Figure 6.

Because the compressive ratio is proportional to the sparity K and the embedding rate decreases with the increasing the sparity K, the embedding rate decreases with the increasing the compressive ratio.

4.3. Against Noises. Each pixel of the embedded Cipher image 1 is transformed into 8 bits. These bits are divided into three parts and included in the coefficients vector $\tilde{\mathbf{Y}}$ by Line (4)–(7) of Algorithm 1. Therefore, these coefficients related to the embedded pixels are usually smaller than those of host image. If the cipher image is explosure in noisy invironment during transmition, the smaller coefficients would be influenced seriously. Therefore, the embedded Cipher image 1 is more sensitive to noises than the host image. Figure 7 discusses the sensitivity of the proposed algorithm to the noises. We add two levels "salt & pepper" noises to the received images. Figures 7(a) and 7(d) are the received cipher images, and they are full of "salt & pepper" noises. Figure 7(a) has noise density of 0.001 and Figure 7(d) is with 0.01. Figures 7(b) and 7(e) are the recovered host images, which are closer to the original host image. However, the quality of the extracted images are not satisfactory, as shown in Figures 7(c) and 7(f), which are in the presence of noises. These results manifest that the proposed algorithm has good reconstruction performance in noisy envioronment, but the embedded image is sensitive to noises, and this property can be used to detect the potential attacks which is unconspicuous on the stego-image.

4.4. Complexity. We would present the complexity from two phases: embedding phase, extraction and reconstruction

Table 1: The SSIM VS. compressive ratio for three host images.

Image	Compressive ratio										
	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75
Lena	0.9912	0.9939	0.9959	0.9972	0.9980	0.9987	0.9990	0.9993	0.9995	0.9997	0.9999
Pepper	0.9868	0.9907	0.9935	0.9954	0.9968	0.9978	0.9984	0.9989	0.9992	0.9995	0.9997
Baboon	0.9778	0.9845	0.9894	0.9926	0.9948	0.9964	0.9975	0.9983	0.9989	0.9993	0.9995

Figure 7: Lena and its extracted data in noisy environment: (a) the cipher images with density of 0.001 "salt & pepper" noises; (b) the recovered host image from (a); (c) the extracted plain image from (a); (d) the cipher images with density of 0.01 "salt & pepper" noises; (e) the recovered host image from (d); (c) the extracted plain image from (d).

phase. In embedding phase, the complexity of encrypting plain image is related to the size of plain image. Therefore, this complexity is $O(m \times n)$. As for embedding the Cipher image 1, the complexities are related to the bits of the plain image and the sparse presentation by OMP. Then, the complexities are $O(8 \times M \times N)$ and $O(K \times M \times N)$ respectively, where K is sparity. In summary, the complexity in the embedding phase is close to $O(K \times M \times N)$.

In extraction and reconstruction phase, the computational complexity can be effectively decreased due to the fact that matrix multiply instead of the CS recovery is executed in this phase, which is one of the main merits of this paper. Then, the complexity in extraction and reconstruction phase is related to multiplication between $\mathbf{\Delta}$ and the cipher image, which is close to $O(M \times N \times (MN - K))$. Therefore the total complexity of our algorithm is $O(M^2 \times N^2)$.

5. Comparison and Discussion

In what follows, we compare our proposed algorithm with three CS based RDH methods [22–24].

First, we discuss these algorithms from the view of the embedded image. Our proposed algorithm embeds natural image, but Refs. [22–24] embed binary image. Binary image

Figure 8: Lena and its extracted data in Gausian noisy environment: (a) the noisy cipher image containing message; (b) the recovered host image; (c) the extracted image.

is much sparser than natural image and it is much easier to be embedded.

Second, from the view of embedding location, we discuss their differences. In [22], the message is embedded in the AC coefficients, and then is compressed by CS. If one wants to extract the message precisely, the plain image must have ultra-sparse pixels. Yamaç et al. [23] proposed a method to embed data in the measurements, but our method is to embed the data in the sparse coefficients. The embedded locations in Hua et al. [24] are the same as those of our proposed algorithm, but there is drastically difference between them.

In [24], Hua et al. expanded possible locations to embed data, and they also dictated that there are $MN - K$ locations for data hiding except the coefficients of host image. At the same time they indicated that multiple chips can be inserted, but the length of multiple chips was not stated explicitly. If inserting multiple chips at the same time, using their detection algorithm multiple chips are inseparable and cannot be extracted precisely. Therefore, they did not give the method how to embed and extract multiple bits. In a word, our proposed work improves Hua's algorithm in two ways. One is to give a method to embed data into $MN-K$ locations at the same time and to extract them precisely. The other is to insert more than two bits into one location.

Last, from the view of embedding rate, our proposed work achieves the largest rate. As previously described, our embedding rate achieves 2 bpp, which is larger than those in [22–24], but it leads to the sensitivity to noises. If we insert one bit into a coefficient location, the demerit (sensitivity to noises) can be made up. The idea is discussed as follows.

In Line (16) of Algorithm 1, the output can be changed to $\widetilde{\mathbf{Y}} = [\ddot{\mathbf{Y}}, \alpha\mathbf{b}]$, where α denotes the embed strength and $\mathbf{b} = \{b_i\}_{i=1}^{MN-K}$ represents the embedded bits. If the ith embedded data is 0, then $b_i = 1$. If the ith embedded data is 1, then $b_i = -1$. Thus, the embed rate can achieve $(MN - K)/MN$. If the transmitted cipher image is polluted by noises, the receiver data is changed to

$$\widetilde{\mathbf{x}} = \mathbf{x} + \mathbf{z} = \mathbf{\Phi}\ddot{\mathbf{Y}} + \mathbf{\Delta}\widetilde{\mathbf{c}} + \mathbf{z}, \tag{14}$$

where $\mathbf{z}$ is additive noise satisfying $\|\mathbf{z}\|_2 \leq \varepsilon$. After extracting data like Line (3) of Algorithm 2, we can obtain

$$\mathbf{\Delta}^T\widetilde{\mathbf{x}} = \mathbf{\Delta}^T(\mathbf{x} + \mathbf{z}) = \mathbf{\Delta}^T\left(\mathbf{\Phi}\ddot{\mathbf{Y}} + \mathbf{\Delta}\widetilde{\mathbf{c}} + \mathbf{z}\right) = \widetilde{\mathbf{c}} + \mathbf{\Delta}^T\mathbf{z}. \tag{15}$$

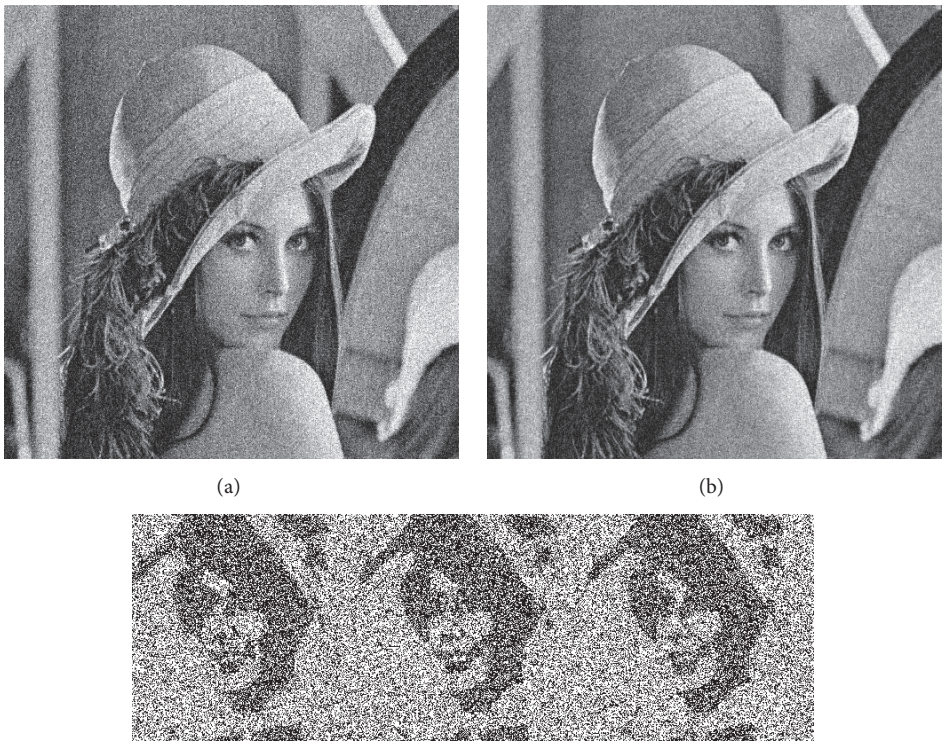
For $\|\mathbf{\Delta}^T\mathbf{z}\|_2 = (\mathbf{\Delta}^T\mathbf{z})^T(\mathbf{\Delta}^T\mathbf{z}) = \mathbf{z}^T\mathbf{\Delta}\mathbf{\Delta}^T\mathbf{z} = \mathbf{z}^T\mathbf{z} = \|\mathbf{z}\|_2 \leq \varepsilon$, we can change (15) into

$$\mathbf{\Delta}^T\widetilde{\mathbf{x}} = \widetilde{\mathbf{c}} + \widetilde{\mathbf{z}} = \alpha\mathbf{b} + \widetilde{\mathbf{z}}, \tag{16}$$

where $\widetilde{\mathbf{z}} = \mathbf{\Delta}^T\mathbf{z}$.

When extracting, we can use $\alpha\,\mathrm{sign}(\mathbf{\Delta}^T\widetilde{\mathbf{x}}) = \alpha\,\mathrm{sign}(\widetilde{\mathbf{c}} + \widetilde{\mathbf{z}}) = \alpha\,\mathrm{sign}(\alpha\mathbf{b} + \widetilde{\mathbf{z}})$ to replace Line (5) of the Algorithm 2. If $\|\widetilde{\mathbf{z}}\|_\infty < \alpha$, then $\alpha\,\mathrm{sign}(\mathbf{\Delta}^T\widetilde{\mathbf{x}}) = \alpha\mathbf{b}$, i.e., the embedded bits can be extracted precisely. In simulation, α is set to be 5.

The simulation results are shown from Figures 8–10. Figures (a) are the transmitted cipher images. Because each



(a) (b)

(c)

Figure 9: Lena and its extracted data in Gaussian noisy environment: (a) the noisy cipher image containing message; (b) the recovered host image; (c) the extracted image.

(a) (b)

(c)

Figure 10: Lena and its extracted data in "salt & pepper" noisy environment: (a) the noisy cipher image containing message; (b) the recovered host image; (c) the extracted image.

pixel in a binary image can be represented by 1 bit, we embed a binary image Lady in three Figures (a). Figure 8(a) is polluted by Gaussian noise, where noise level is smaller than the embed strength α. From Figure 8(c), we can see the extracted image is without noises. In contrast, the Gaussian noise level in Figure 9(a) is larger than the embed strength and from Figure 9(c) we can see the extracted image is degraded by Gaussian noise. In Figure 10(a), the cipher image is exposed in "salt & pepper" noisy environment. From Figure 10(c), we can see that the extracted result is degraded as same as Figure 9(c). It is because the "salt" noises are much larger than α, and the "pepper" noises cleared some information of transmitted data. In fact, these noises caused by strong Gaussian or "salt & pepper" cannot be eliminated by these methods [22–24].

We can conclude, for our proposed algorithm in this section, decreasing the embedding rate and embedding a bit in a location of coefficients vector can improve the robustness to noises. If the compressive ratio is still $K/MN = 0.25$, the embedding rate is 0.75 bpp, which is still larger than those in [22–24].

6. Conclusions

In this paper, an algorithm for meaningful image encryption is proposed. The main idea is to embed the secret image into the sparse representation of host image to achieve high covertness. Our proposed algorithm has the following merits. First, a novel method of RDH in CS domain is proposed, and the data embedding capacity is significantly higher than other similar schemes. Second, we propose the parity checks strategy to eliminate quantization error. Third, if the transmitted cipher image is not polluted by noises, the embedding rate achieves 2 bpp. And, if the transmitted cipher image is polluted by additive noise with lower level, the embedding rate $(MN - K)/(MN)$ bpp is equal to the difference between 1 and compressive ratio. Last, we use matrix multiplication in reconstruction phase to decrease computational complexity. In the future, we aim to improve the robustness of the proposed algorithm with lower computational complexity.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 61602158, 61572089, 61633005, U1604156, 61671042, 61403016), the China Postdoctoral Science Foundation (Grant no. 2016M600030), the Beijing Natural Science Foundation (Grant no. 4172037), the Open Fund Project of Fujian Provincial Key Laboratory in Minjiang University (Grant no. MJUKF201702), the Science Foundation for the Excellent Youth Scholars of Henan Normal University (Grant no. YQ201607), the Natural Science Foundation of Chongqing Science and Technology Commission (Grant no. cstc2017jcyjBX0008), and the Fundamental Research Funds for the Central Universities (Grants nos., 106112017CDJQJ188830, 106112017CDJXY180005).

References

[1] X. Liao, J. Yin, and S. Guo, "Medical JPEG image steganography based on preserving inter-block dependencies," *Computers Electrical Engineering*, 2017.

[2] X. Liao, Z. Qin, and L. Ding, "Data embedding in digital images using critical functions," *Signal Processing: Image Communication*, vol. 58, pp. 146–156, 2017.

[3] M. Wu, H. Yu, and B. Liu, "Data hiding in image and video: Part II - Designs and applications," *IEEE Transactions on Image Processing*, vol. 12, no. 6, pp. 696–705, 2003.

[4] B. Feng, W. Lu, and W. Sun, "Secure binary image steganography based on minimizing the distortion on the texture," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, article no. A3, pp. 243–255, 2015.

[5] X. Liao, G. Chen, and J. Yin, "Content-adaptive steganalysis for color images," *Security and Communication Networks*, vol. 9, no. 18, pp. 5756–5763, 2016.

[6] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 253–266, 2005.

[7] X. Liao, K. Li, and J. Yin, "Separable data hiding in encrypted image based on compressive sensing and discrete fourier transform," *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 20739–20753, 2017.

[8] G. Bhatnagar, "Robust covert communication using high capacity watermarking," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 3783–3807, 2017.

[9] B. Ma and Y. Q. Shi, "A reversible data hiding scheme based on code division multiplexing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 1914–1927, 2016.

[10] S. A. Parah, J. A. Sheikh, A. M. Hafiz, and G. M. Bhat, "A secure and robust information hiding technique for covert communication," *International Journal of Electronics*, vol. 102, no. 8, pp. 1253–1266, 2015.

[11] F. Chen, K.-w. Wong, X. Liao, and T. Xiang, "Period distribution of generalized discrete Arnold cat map," *Theoretical Computer Science*, vol. 552, pp. 13–25, 2014.

[12] Y. Zhang, J. Zhou, F. Chen et al., "Embedding cryptographic features in compressive sensing," *Neurocomputing*, vol. 205, pp. 472–480, 2016.

[13] Y. Zhang, J. Zhou, F. Chen et al., "A block compressive sensing based scalable encryption framework for protecting significant image regions," *International Journal of Bifurcation and Chaos*, vol. 26, no. 11, Article ID 1650191, 2016.

[14] A. Kanso and M. Ghebleh, "An algorithm for encryption of secret images into meaningful images," *Optics and Lasers in Engineering*, vol. 90, pp. 196–208, 2017.

[15] X. Chai, Z. Gan, Y. Chen, and Y. Zhang, "A visually secure image encryption scheme based on compressive sensing," *Signal Processing*, vol. 134, pp. 35–51, 2017.

[16] L. Kamstra and H. J. Heijmans, "Reversible data embedding into images using wavelet techniques and sorting," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2082–2090, 2005.

[17] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.

[18] X. Liao and C. Shu, "Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels," *Journal of Visual Communication and Image Representation*, vol. 28, pp. 21–27, 2015.

[19] M. Li and Y. Li, "Histogram shifting in encrypted images with public key cryptosystem for reversible data hiding," *Signal Processing*, vol. 130, pp. 190–196, 2017.

[20] M. Li, D. Xiao, Y. Zhang, and H. Nan, "Reversible data hiding in encrypted images using cross division and additive homomorphism," *Signal Processing: Image Communication*, vol. 39, pp. 234–248, 2015.

[21] Y. Zhang, L. Y. Zhang, J. Zhou, L. Liu, F. Chen, and X. He, "A review of compressive sensing in information security field," *IEEE Access*, vol. 4, pp. 2507–2519, 2016.

[22] D. Xiao, H. Cai, Y. Wang, and S. Bai, "High-capacity separable data hiding in encrypted image based on compressive sensing," *Multimedia Tools and Applications*, vol. 75, no. 21, pp. 13779–13789, 2016.

[23] M. Yamaç, Ç. Dikici, and B. Sankur, "Hiding data in compressive sensed measurements: A conditionally reversible data hiding scheme for compressively sensed measurements," *Digital Signal Processing*, vol. 48, pp. 188–200, 2016.

[24] G. Hua, Y. Xiang, and G. Bi, "When compressive sensing meets data hiding," *IEEE Signal Processing Letters*, vol. 23, no. 4, pp. 473–477, 2016.

[25] M. Li, D. Xiao, and Y. Zhang, "Reversible data hiding in block compressed sensing images," *ETRI Journal*, vol. 38, no. 1, pp. 159–163, 2016.

[26] Y. Rachlin and R. D. Baron, "The secrecy of compressed sensing measurements," in *Proceedings of the 46th Annual Allerton Conference on Communication, Control, and Computing*, pp. 813–817, IEEE, Urbana, Ill, USA, September 2008.

[27] A. Orsdemir, H. O. Altun, G. Sharma, and M. F. Bocko, "On the security and robustness of encryption via compressed sensing," in *Proceedings of the IEEE Military Communications Conference (MILCOM '08)*, pp. 1–7, San Diego, Calif, USA, November 2008.

[28] J. A. Tropp, "Greed is good: algorithmic results for sparse approximation," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[29] A.-V. Diaconu, "Circular inter-intra pixels bit-level permutation and chaos-based image encryption," *Information Sciences*, vol. 355-356, pp. 314–327, 2016.

Hindawi
Security and Communication Networks
Volume 2017, Article ID 9602357, 12 pages
https://doi.org/10.1155/2017/9602357

Research Article

F-DDIA: A Framework for Detecting Data Injection Attacks in Nonlinear Cyber-Physical Systems

Jingxuan Wang,[1] Lucas C. K. Hui,[1] S. M. Yiu,[1] Gang Zhou,[2] and Ruoqing Zhang[1]

[1]*Department of Computer Science, University of Hong Kong, Pokfulam Road, Hong Kong*
[2]*Peking University, Beijing, China*

Correspondence should be addressed to Jingxuan Wang; hongkongwangjingxuan@gmail.com

Received 10 April 2017; Accepted 7 June 2017; Published 10 August 2017

Academic Editor: Leo Y. Zhang

Data injection attacks in a cyber-physical system aim at manipulating a number of measurements to alter the estimated real-time system states. Many researchers recently focus on how to detect such attacks. However, most of the detection methods do not work well for the nonlinear systems. In this paper, we present a compressive sampling methodology to identify the attack, which allows determining how many and which measurement signals are launched. The sparsity feature is used. Generally, our methodology can be applied to both linear and nonlinear systems. The experimental testing, which includes realistic load patterns from NYISO with various attack scenarios in the IEEE 14-bus system, confirms that our detector performs remarkably well.

1. Introduction

A cyber-physical system (CPS) is a dynamical system, which integrates the computational components (i.e., real-time operations) with its physical components (i.e., hardware facilities). Examples of CPS can be large-scale distributed systems, such as smart grid, transportation networks, railway control system, and medical monitoring. The design of CPS involves various of disciplines, such as control engineering, software engineering, and mechanics and networks. Particularly, control engineering is a communication network for transmitting sensor data (measurements) so that the system operator can in real-time monitor the production process. Among the control disciplines, a scheme called bad data detector (BDD) is applied to detect whether there exists a disruption of sensor data caused by the genetic malfunction or malicious attacks. The classical BDD technique is to utilize the "residual principle," which calculates the difference between the observed readings and the computed readings based on the estimated system states. When an attack is injected into the system, BDD will remove those readings (collected from the sensors), of which residuals are larger than a threshold.

As the increased vulnerabilities proposed by the recent discoveries of system malware, concerns about the security of CPS are arising. In 2011, a malware, known as Stuxnet [1], successfully penetrated the networks of Iran's uranium enrichment infrastructure via programmable logic controllers. From this instance, we can see that it is possible for an attacker to introduce errors on physical readings. Inspired by this attacking strategy, a class of attacks named *data injection attacks* are proposed in recent years, which can affect the system control algorithms and thus lead to abnormal operations [2, 3]. Hence, sufficient attention should be paid to the detection techniques against this attack, which is easy to be implemented by strong adversaries who are quite knowledgeable about the targeted systems.

To fight against this attack, existing works focus on the detection of data injection attacks and the protection of nonlinear measurements [4, 5]. Detectors utilizing the sparsity and low rank of the system topology are proposed in [6–8]. Greedy and game theory methods have been used for optimizing the placement of devices [9], to lower the possibility of the construction of data injection attacks. Applying the machine learning techniques to conduct the classification is proposed in [10]. They propose a "first difference aware" machine learning (FDML) classifier to detect the cyber attacks. A graph theory-based algorithm is proposed in [11] to determine which measurement signals an attacker will alter. However, we notice that all detection models except [11, 12]

are conducted in a constrained setting, by assuming that the functions from system states to measurements are linear. This assumption is too stringent to fit for some nonlinear systems, for example, alternative current (AC) model in power grids.

This paper investigates an alternative approach to detect data injection attacks in the nonlinear system. We propose a detector framework named F-DDIA to reconstruct the initial states of the plant from the corrupted observations, which formulates an error correction problem. In particular, we notice that, due to the property of data injection attacks, only a small fraction of the observations are supposed to be attacked at a given time instance. Thus, we formulate the error correction problem as a sparse optimization problem which can be solved with the general ℓ_1-minimization program technique. In this paper, we apply Douglas-Rachford techniques [13] among minimization techniques. Furthermore, we employ the "divide-and-conquer" principle to construct a compressive sensing model of a linear subspace, which is interesting in the general mathematical settings.

To validate and illustrate our algorithm, we use real-world CPS power grids as a case study. In particular, we use the data injection attacks model proposed in [2], where the attacks are directed by injecting false data into the sensors. Simulations based on IEEE 14-bus test systems validate the effectiveness of our methodology. The results show that the proposed algorithm can efficiently identify the data injection attacks (i.e., with high precision and recall values) and recover the initial system states (i.e., with small average phase error).

The rest of this paper is organized as follows. Section 2 presents the system model in a nonlinear system, including preliminaries related to a broad class of attacks. Section 3 states the problem and derives a theoretical justification of the efficacy of the security algorithm in a general cyber-physical system model. Section 4 analyzes the performance of the proposed approach through simulations. Section 5 gives concluding remarks.

2. Preliminaries

2.1. System Model and Bad Data Detector. A cyber-physical system is usually described by the following widely adopted discrete-time nonlinear dynamical model:

$$x[k+1] = \delta(x[k]) + Bu[k] + w[k], \tag{1a}$$

$$z[k] = h(x[k]) + e[k], \tag{1b}$$

where at time $k \in \mathcal{J} \triangleq \{0, \ldots, T-1\}$: $x[k] \in \mathbb{R}^n$ is the system state; $u[k] \in \mathbb{R}^l$ is the bounded input vector; $a[k] \in \mathbb{R}^m$ is the measurement vector (data collected by the sensors); $w[k]$ denotes the state noise (i.e., Gaussian with known statistics); and $v[k]$ denotes measurement errors. Here the matrix B is a constant matrix, $\delta : \mathbb{R}^n \to \mathbb{R}^n$ denotes the state transition function and $h : \mathbb{R}^n \to \mathbb{R}^m$ denotes the topology of the system, which are the nonlinear functions with respect to the states. The process of estimating system states from the measurements is called *state estimation*.

In traditional weighted least squares (WLS) state estimation, the system states are valid only if the measurement residual vector $r[k]$ is less than a threshold [14],

$$r[k] = \|z[k] - h(\hat{x}[k])\|_{\ell_2}, \tag{2}$$

where $\hat{x}[k]$ is the estimated system state after the process of state estimation. Specifically, the presence of bad measurements is inferred if $Jr[k] > \tau$, where τ is a chosen identification threshold. Upon detection of bad data, two kinds of methods, named the largest normalized residual test ($r_N^{\max}$) and hypothesis testing identification (HTI) method, are widely used to identify whether the measurements contain bad data.

2.2. Data Injection Attack. Data injection attacks are commonly known as *false data injection attacks* [2], *data framing attacks* [3, 15], in the sense of the following definition.

Definition 1. A vector $a[k]$ is called a (κ, m)-data injection attack if there exists an index set $i \in \mathcal{A}$, where $\mathcal{A}$ is the set of manipulated measurements and $\mathcal{A} \subset \mathcal{P} \triangleq \{1, \ldots, m\}$, such that

(i) $\|a[k]\|_{\ell_0} \leq \kappa$;

(ii) $a_i[k] = 0, \ \forall i \in \mathcal{P} \setminus \mathcal{A}$;

(iii) $a_i[k] \neq 0, \ \forall i \in \mathcal{A}$.

To implement this class of attack, it requires the attacker to have the knowledge of either the measurements information (z) or the topology configuration ($h(\cdot)$). Specifically, data injection attack can be written in the form of

$$\overline{z}[k] = z[k] + a[k] = h(x[k]) + a[k], \tag{3}$$

where $a[k]$ is the injected false measurement data. There are many ways to generate this type of attacks. For example, if $h(\cdot)$ is available to the attacker, the attack a can be constructed in the following form (namely, false data injection attack in a linear system):

$$a = Hc, \tag{4}$$

where c is the error injected on the system state and $H = \partial h(x)/\partial x$ is the Jacobian matrix. However, to implement this attack, the attacker needs to take control of at least κ sensors, where $\kappa \leq m$.

2.3. Measurement Dynamics. We can use the polynomial regression approach to fit the measurement dynamics,

$$z[k+1] = \delta(x[k]) + Bu[k] + w[k] = \delta'(z[k]), \tag{5}$$

where $\delta' : \mathbb{R}^m \to \mathbb{R}^m$ denotes the dynamics of the measurements. Furthermore, we define $z_i[k]$ as the ith corrupted measurement at time k. That is, a polynomial regression model, which expresses the dynamics of the ith measurement can be given as follows:

$$\delta_i'(z_i[k]) = \gamma_{i,1}(z_i[k])^l + \cdots + \gamma_{i,l}(z_i[k]) + \gamma_{i,l+1}, \tag{6}$$

where l is called the degree of the polynomial and $i \in \mathscr{P}$. We denote $\gamma_i = (\gamma_{i,1}, \ldots, \gamma_{i,l+1}) \in \mathbb{R}^{l+1}$. As $\delta_i'(z_i[k])$ can be expressed in matrix form in terms of a response vector $z_i[k]$ and a parameter vector $\gamma_{i,j}$, where $1 \le j \le l+1$, we can rewrite $z_i[k+1]$ as a system of linear equations:

$$z_i[k+1] = X \begin{pmatrix} \gamma_{i,1} \\ \vdots \\ \gamma_{i,l+1} \end{pmatrix}, \tag{7}$$

where $X = ((z_i[k])^l \quad \cdots \quad z_i[k] \quad 1) \in \mathbb{R}^{l+1}$. Thus, the dynamical matrix γ can be estimated as

$$\widehat{\gamma}_i = \left(X^T X\right)^{-1} X^T z_i[k+1] \quad (i \in \mathscr{P}). \tag{8}$$

3. Our Methodologies

In this section, we formulate the detection problem as an error correction problem. We will further describe and explain why we can use ℓ_1-norm minimization technique (including Douglas-Rachford) to solve the detection problem.

3.1. Sparse Optimization Problem Formulation. In this paper, we consider the scenario that an attacker is limited to the resources of κ sensors and possesses the knowledge of system topology h, as well as the historical measurements $\overline{Z} = (\overline{z}[0]; \ldots; \overline{z}[T-1]) \in \mathbb{R}^{mT}$. Denote $Z = (z[0]; \ldots; z[T-1]) \in \mathbb{R}^{mT}$ as the initial measurements (without attacks) in time base. The obtained temporal observations $\overline{Z}$ can be expressed as

$$\overline{Z} = Z + \mathbb{A}, \tag{9}$$

where $\mathbb{A} = (a[0]; \ldots; a[T-1]) \in \mathbb{R}^{mT}$. Remark that, due to the property of data injection attacks, only a small fraction of the observations are supposed to be attacked at a given time instance. Hence, noticing the sparsity of vector $\mathbb{A}$, the detection problem can be converted to

$$\begin{aligned} \underset{\mathbb{A}}{\text{minimize}} \quad & \|\mathbb{A}\|_{\ell_0} \\ \text{subject to} \quad & \overline{Z} = Z + \mathbb{A}, \\ & \|a[k]\|_{\ell_0} \le \kappa, \ k \in \mathscr{I}, \end{aligned} \tag{10}$$

where κ is the maximum number of the meters that can be compromised. Under certain conditions which are explained above, we will focus on the problem of recovering the sparse vector $\mathbb{A}$ from $\overline{Z}$. And we denote the optimal solution of problem (10) as $\mathbb{A}^*$.

3.2. Subproblem Formulation. In the rest of this paper, we define the matrices $\mathbb{A} = [\mathbb{A}_1, \ldots, \mathbb{A}_{Tm+m}]$, $Z = [Z_1, \ldots, Z_{Tm+m}]$, and $\overline{Z} = [\overline{Z}_1, \ldots, \overline{Z}_{Tm+m}]$. We further define the matrices E, W, and $\overline{W}$ in the following forms:

$$\begin{aligned} E &= \begin{bmatrix} E_1^T \\ \vdots \\ E_m^T \end{bmatrix} = \begin{bmatrix} \mathbb{A}_1 & \mathbb{A}_{m+1} & \cdots & \mathbb{A}_{Tm+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{A}_m & \mathbb{A}_{2m} & \cdots & \mathbb{A}_{Tm+m} \end{bmatrix} \in \mathbb{R}^{m \times T}; \\ W &= \begin{bmatrix} W_1^T \\ \vdots \\ W_m^T \end{bmatrix} = \begin{bmatrix} Z_1 & Z_{m+1} & \cdots & Z_{Tm+1} \\ \vdots & \vdots & \ddots & \vdots \\ Z_m & Z_{2m} & \cdots & Z_{Tm+m} \end{bmatrix} \in \mathbb{R}^{m \times T}; \\ \overline{W} &= \begin{bmatrix} \overline{W}_1^T \\ \vdots \\ \overline{W}_m^T \end{bmatrix} = \begin{bmatrix} \overline{Z}_1 & \overline{Z}_{m+1} & \cdots & \overline{Z}_{Tm+1} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{Z}_m & \overline{Z}_{2m} & \cdots & \overline{Z}_{Tm+m} \end{bmatrix} \in \mathbb{R}^{m \times T}. \end{aligned} \tag{11}$$

We can further obtain the following formulation among $E_i \in \mathbb{R}^T$, $W_i \in \mathbb{R}^T$, and $\overline{W}_i \in \mathbb{R}^T$:

$$\begin{aligned} \overline{W}_i &= W_i + E_i \quad (i \in \mathscr{P}), \\ \|\mathbb{A}\|_{\ell_0} &= \sum_{j=1}^{mT} \left\|\mathbb{A}_j\right\|_{\ell_0} = \sum_{i=1}^{m} \left\|E_i\right\|_{\ell_0} = \|E\|_{\ell_0}. \end{aligned} \tag{12}$$

We denote by $\mathrm{col}_{k \in \mathscr{I}}(E) \in \mathbb{R}^m$ the columns of the matrix E. Hence, problem (10) is equivalent to

$$\begin{aligned} \underset{E}{\text{minimize}} \quad & \|E\|_{\ell_0} \\ \text{subject to} \quad & \overline{W} = W + E, \\ & \left\|\mathrm{col}_k(E)\right\|_{\ell_0} \le \kappa, \ k \in \mathscr{I}. \end{aligned} \tag{13}$$

Note that $\|E\|_{\ell_0} = \sum_{i=1}^{m} \|E_i\|_{\ell_0}$; we can further solve problem (13) by seeking for the locally optimal choice for each E_i^* with the hope of finding a globally optimal solution (E^*):

$$\begin{aligned} \underset{E_i}{\text{minimize}} \quad & \left\|E_i\right\|_{\ell_0} \\ \text{subject to} \quad & \overline{W}_i = W_i + E_i, \\ & (i \in \mathscr{P}). \end{aligned} \tag{14}$$

The solution of this subproblem (14) will be given in Section 3.4. After solving m above optimization problems, the optimal solution E^* will be checked by the following constraints:

$$f\left(\mathrm{col}_k\left(E^*\right)\right) = \mathrm{sgn}\left(\kappa - \left\|\mathrm{col}_k\left(E^*\right)\right\|_{\ell_0}\right). \tag{15}$$

For any $k \in \mathscr{I}$, if $f(\mathrm{col}_k(E^*)) = 1$, there exists the attack; otherwise, there does not exist any data injection attack.

3.3. Solving Subproblem by ℓ_1-Minimization. Recall that the dynamical coefficients $(\gamma_1, \ldots, \gamma_m)$ are obtained (by polynomially fitting in Section 2.3). In view of adversary, $\overline{W}_i$ can be

rewritten as

$$\begin{aligned}\overline{W}_i &= \begin{pmatrix} \overline{z}_i[0] \\ \overline{z}_i[1] \\ \overline{z}_i[2] \\ \vdots \\ \overline{z}_i[T-1] \end{pmatrix} \\ &= \begin{pmatrix} z_i[0] + a_i[0] \\ z_i[1] + a_i[1] \\ z_i[2] + a_i[2] \\ \vdots \\ z_i[T-1] + a_i[T-1] \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ \gamma_{i,l} \\ (\gamma_{i,l})^2 \\ \vdots \\ (\gamma_{i,l})^{T-1} \end{pmatrix} z_i[0] \\ &\quad + \begin{bmatrix} 1 & & & \\ \gamma_{i,l} & 1 & & \\ (\gamma_{i,l})^2 & \gamma_{i,l} & 1 & \\ \vdots & \vdots & & \ddots \\ (\gamma_{i,l})^{T-1} & \cdots & \gamma_{i,l} & 1 \end{bmatrix} E_i \\ &\quad + \begin{pmatrix} 0 \\ g_i(\overline{z}[0]) \\ g_i(\overline{z}[1]) + \gamma_{i,l} g_i(z[0]) \\ \vdots \\ g_i(\overline{z}[T-1]) + \gamma_{i,l} g_i(z[T-2]) + \cdots \end{pmatrix}. \end{aligned} \tag{16}$$

Then we use the notation $\widetilde{W}_i$ as follows:

$$\begin{aligned}\widetilde{W}_i &= \overline{W}_i \\ &\quad - \begin{pmatrix} 0 \\ g_i(\overline{z}[0]) \\ g_i(\overline{z}[1]) + \gamma_{i,l} g_i(z[0]) \\ \vdots \\ g_i(\overline{z}[T-1]) + \gamma_{i,l} g_i(z[T-2]) + \cdots \end{pmatrix} \\ &= \Gamma_i z_i[0] + \Psi_i E_i, \end{aligned} \tag{17}$$

where the matrices $\Gamma_i \in \mathbb{R}^T$ and $\Psi_i \in \mathbb{R}^{T \times T}$ are

$$\begin{aligned}\Gamma_i &= \begin{pmatrix} 1 \\ \gamma_{i,l} \\ (\gamma_{i,l})^2 \\ \vdots \\ (\gamma_{i,l})^{T-1} \end{pmatrix}, \\ \Psi_i &= \begin{bmatrix} 1 & & & \\ \gamma_{i,l} & 1 & & \\ (\gamma_{i,l})^2 & \gamma_{i,l} & & \\ \vdots & \vdots & \ddots & \\ (\gamma_{i,l})^{T-1} & \cdots & & 1 \end{bmatrix}. \end{aligned} \tag{18}$$

In this paper, We have an approximation $g_i(z[k]) \doteq g_i(\overline{z}[k])$. The reason we take this approximation is that the difference of $z[k]$ and $\overline{z}[k]$ is

$$\begin{aligned} g_i(\overline{z}[k]) - g_i(z[k]) &= \gamma_{i,1} \overline{z}_i[k]^l + \cdots + \gamma_{i,l-1} \overline{z}_i[k]^2 \\ &\quad - \gamma_{i,1} z_i[k]^l - \cdots \\ &\quad - \gamma_{i,l-1} z_i[k]^2. \end{aligned} \tag{19}$$

For example, $g_i(\overline{z}[k]) - g_i(z[k]) = \gamma_{i,1} a_i[k](1 + a_i[k])$ when $l = 2$. Since the values of $\gamma_{i,1}$ are small ($i = 1, \ldots, m$), $g_i(z[k]) \doteq g_i(\overline{z}[k])$. We have done experiments about this fact, and the experimental result supports our approximation claim. Then, $\widetilde{W}_i$ in (17) can be updated as

$$\begin{aligned}\widetilde{W}_i &\doteq \overline{W}_i \\ &\quad - \begin{pmatrix} 0 \\ g_i(\overline{z}[0]) \\ g_i(\overline{z}[1]) + \gamma_{i,l} g_i(\overline{z}[0]) \\ \vdots \\ g_i(\overline{y}[T-1]) + \gamma_{i,l} g_i(\overline{z}[T-2]) + \cdots \end{pmatrix}. \end{aligned} \tag{20}$$

We can further take the QR decomposition of $\Gamma_i \in \mathbb{R}^T$ [16]:

$$\Gamma_i = S \begin{pmatrix} R_1 \\ 0 \end{pmatrix} = \begin{bmatrix} S_{i1} & S_{i2} \end{bmatrix} \begin{pmatrix} R_1 \\ 0 \end{pmatrix}, \tag{21}$$

where $S \in \mathbb{R}^{T \times T}$, $S_{i1} \in \mathbb{R}^T$, $S_{i2} \in \mathbb{R}^{T \times (T-1)}$, $R_{i1} \in \mathbb{R}^1$, and $\begin{bmatrix} S_{i1} & S_{i2} \end{bmatrix}$ is orthogonal. Before multiplying (17) by $\begin{bmatrix} S_{i1} & S_{i2} \end{bmatrix}^T$, we can have

$$\begin{bmatrix} S_{i1}^T \\ S_{i2}^T \end{bmatrix} \widetilde{W}_i = \begin{pmatrix} R_{i1} \\ 0 \end{pmatrix} z_i[0] + \begin{bmatrix} S_{i1}^T \\ S_{i2}^T \end{bmatrix} \Psi_i E_i. \tag{22}$$

By using the second block row, we can solve the following problem to obtain the sparse solution E, instead of $\mathbb{A}$:

$$S_{i2}^T \widetilde{W}_i = S_{i2}^T \Psi_i E_i \quad (i \in \mathscr{P}). \tag{23}$$

Hence, the problem is reduced to reconstruct a sparse vector E_i from the observations $S_{i2}^T \widetilde{W}_i$. Problem (14) is equivalent to the following problem:

$$\begin{aligned} \underset{E_i}{\text{minimize}} \quad & \|E_i\|_{\ell_0} \\ \text{subject to} \quad & S_{i2}^T \Psi_i E_i = S_{i2}^T \widetilde{W}_i, \end{aligned} \tag{24}$$

where $E_i \in \mathbb{R}^T$. As is discussed above, solving problem (24) is in general NP-hard since it requires searches over all subsets of columns of $S_{i2}^T \Psi_i$, a procedure which has exponential complexity. To overcome this problem, a frequently discussed approach considers a similar program in the ℓ_1-norm:

$$\begin{aligned} \underset{E_i}{\text{minimize}} \quad & \|E_i\|_{\ell_1} \\ \text{subject to} \quad & S_{i2}^T \Psi_i E_i = S_{i2}^T \widetilde{W}_i. \end{aligned} \tag{25}$$

This operation is common and can be found in [13, 17, 18]. Throughout this paper, we consider Douglas-Rachford splitting algorithm [13] in the context of above ℓ_1-minimization.

3.4. Theoretical Guarantee. In this paper, we are also interested in studying the theoretical conditions under which obtaining the solution of the problem is guaranteed. It is well known that an inverse problem of finding the solution to the compressive sensing problem involves mathematical questions on the existence, uniqueness, and stability of the solution. On the other hand, the equivalence of the solution between (13) and (25) is not very clear and proof may be needed. We therefore consider two questions for a given $S_{i2}^T \Psi_i$ and signal $S_{i2}^T \widetilde{W}_i$ ($i \in \mathscr{P}$): (i) *uniqueness*: under which conditions a possible sparsest solution is necessarily unique to problem (13)/(25)? and (ii) *equivalence*: under which conditions a sparse solution to problem (13) is also equivalent to the solution of problem (25)?

3.4.1. Uniqueness. As is described in Section 3.3, solving problem (24) requires exhaustive searches over all subsets of columns of $S_{i2}^T \Psi_i$. Actually, it is a combinatorial procedure in nature and has exponential complexity. Inspired by [7, 17], Theorem 3 provides a sufficient condition for a unique solution to problem (24). It guarantees obtaining a unique sparse vector (i.e., E) from the corrupted observations (i.e., $\overline{Z}$) for the ℓ_0 minimization. We denote by $\text{row}_{i \in \mathscr{P}}(E) \in \mathbb{R}^T$ the rows of the matrix E. Before giving the theorem, we need to first introduce the following definition [17].

Definition 2 (see [17, Definition 1.1]). Let $S_{i2}^T \Psi_i$ be the matrix with the finite collection of vectors $\text{col}(S_{i2}^T \Psi_i)_{k \in \mathscr{J}} \in \mathbb{R}^m$ as columns. For every integer $1 \le \nu \le |\mathscr{J}|$, we define the ν-restricted isometry constants ρ_ν to be the smallest quantity such that $S_{i2}^T \Psi_i$ obeys

$$(1 - \rho_\nu) \|E_i\|^2 \le \left\| S_{i2}^T \Psi_i E_i \right\|^2 \le (1 + \rho_\nu) \|E_i\|^2, \tag{26}$$

for all real coefficients $E_{i \in \mathscr{P}}$.

The number ρ_ν measures how close the vectors $\text{row}_i(S_{i2}^T \Psi_i)$ are to behave. In particular, for $\nu = 1$, we can have

$$1 - \rho_1 \le \left\| \text{row}_i \left(S_{i2}^T \Psi_i \right) \right\|^2 \le 1 + \rho_1, \quad \text{for } \forall i \in \mathscr{P}. \tag{27}$$

To see the relevance of ρ_ν to the error recovery problem, we consider the following theorem.

Theorem 3. *In a cyber-physical system, let S_{i2}, $\widetilde{W}_i$, Ψ_i, ν, κ, and $\mathscr{J}$ be specified as above. A sparse solution E can be uniquely recovered from solving the optimization problem (13), if $\rho_{2\nu} < 1$, and $\|\text{col}_{k \in \mathscr{J}}(E)\|_{\ell_0} \le \kappa$.*

Proof. We first prove that if $\rho_{2\nu} < 1$, there exists a unique E_i to problem (24). Suppose for the sake of contradiction that the solution is not unique; then there exist two solutions $E^{\text{opt1}} \ne E^{\text{opt2}}$. Thus, there exists at least one variable i ($1 \le i \le m$) such that

$$\begin{aligned} S_{i2}^T \Psi_i \text{row}_i \left(E^{\text{opt1}} \right) &= S_{i2}^T \widetilde{W}_i, \\ S_{i2}^T \Psi_i \text{row}_i \left(E^{\text{opt2}} \right) &= S_{i2}^T \widetilde{W}_i, \end{aligned} \tag{28}$$

where $\|\text{row}_i(E^{\text{opt1}})\|_{\ell_0} = \|\text{row}_i(E^{\text{opt2}})\|_{\ell_0} = \nu$. Then we can have

$$S_{i2}^T \Psi_i \left(\text{row}_i \left(E^{\text{opt1}} \right) - \text{row}_i \left(E^{\text{opt2}} \right) \right) = 0. \tag{29}$$

By construction $\text{row}_i(E^{\text{opt1}}) - \text{row}_i(E^{\text{opt2}})$ is of size less than or equal to 2ν. Applying (27) and the hypothesis $\rho_{2\nu} < 1$, we conclude that $\|\text{row}_i(E^{\text{opt1}}) - \text{row}_i(E^{\text{opt2}})\|^2 = 0$, contradicting the hypothesis that $\text{row}_i(E^{\text{opt1}})$ and $\text{row}_i(E^{\text{opt2}})$ are distinct.

Then we prove that E is unique to problem (13). Given the proof that E_i, or equivalently $\text{row}_i(E)$, can be uniquely obtained by solving problem (24) and $E = [E_1; \ldots; E_m]$, we conclude that E is unique to the following problem:

$$\begin{aligned} \underset{E}{\text{minimize}} \quad & \|E\|_{\ell_0} \\ \text{subject to} \quad & \overline{W} = W + E. \end{aligned} \tag{30}$$

And given the condition that $\|\text{col}(E)_{k \in \mathscr{J}}\|_{\ell_0} \le \kappa$, we can conclude that E is also unique to problem (13). □

In the literature, a lot of efforts have been made to determine how sparse the desired corrected error must be for equivalence to hold. As we consider to use ℓ_1-minimization instead of ℓ_0 (to obtain the desired error), the conditions in the above lemma may not be guaranteed. Thus, Theorem 4 gives a general condition, which guarantees a unique solution E_i for ℓ_1-minimization problem.

Theorem 4. *In a cyber-physical system, let S_{i2}, $\widetilde{W}_i$, and Ψ_i be specified as above. A sparse solution E can be uniquely recovered from solving the optimization problem*

$$\begin{aligned} \underset{E}{\text{minimize}} \quad & \|E\|_{\ell_1} \\ \text{subject to} \quad & \overline{W} = W + E, \\ & \left\| \text{col}\,(E)_k \right\|_{\ell_1} \le \kappa, \quad k \in \mathscr{J}, \end{aligned} \tag{31}$$

if, for all $E^* \neq E$, we have $\|(E-E^*)_J\|_{\ell_1} - \|(E-E^*)_{\bar{J}}\|_{\ell_1} < 0$ and $\|\mathrm{col}(E)_{k\in\mathcal{J}}\|_{\ell_0} \leq \kappa$, where J and $\bar{J}$ are the support of vectors E and $E^* - E$, respectively.

Proof. We prove that given any $E^{\mathrm{opt1}} \neq E^{\mathrm{opt2}}$ and $\|(E^{\mathrm{opt2}} - E^{\mathrm{opt1}})_J\|_{\ell_1} - \|(E^{\mathrm{opt2}} - E^{\mathrm{opt1}})_{\bar{J}}\|_{\ell_1} < 0$ and $\|\mathrm{col}_k(E)\|_{\ell_0} \leq \kappa$ ($k \in \mathcal{J}$), we can always uniquely recover E^* from (31). Suppose for the sake of contradiction that the solution is not unique; then there exist two instinct solutions that $E^{\mathrm{opt1}} \neq E^{\mathrm{opt2}}$ but $\|E^{\mathrm{opt1}}\|_{\ell_1} = \|E^{\mathrm{opt2}}\|_{\ell_1}$. We use the vectors $\mathbb{A}^{\mathrm{opt1}} = [\mathrm{row}_1(E^{\mathrm{opt1}}); \ldots; \mathrm{row}_m(E^{\mathrm{opt1}})] \in \mathbb{R}^{mT}$ and $\mathbb{A}^{\mathrm{opt2}} = [\mathrm{row}_1(E^{\mathrm{opt2}}); \ldots; \mathrm{row}_m(E^{\mathrm{opt2}})] \in \mathbb{R}^{mT}$ instead of E^{opt1} and E^{opt2}, respectively.

$$\begin{aligned}
\left\|\mathbb{A}^{\mathrm{opt1}}\right\|_{\ell_1} &= \left\|\overline{Z} - Z^{\mathrm{opt1}}\right\|_{\ell_1} = \left\|Z^{\mathrm{opt1}} + \mathbb{A}^{\mathrm{opt1}} - Z^{\mathrm{opt2}}\right\|_{\ell_1} \\
&= \left\|\left(\mathbb{A}^{\mathrm{opt2}} - Z^{\mathrm{opt1}} + Z^{\mathrm{opt2}}\right)_J\right\|_{\ell_1} \\
&\quad + \left\|\left(Z^{\mathrm{opt1}} - Z^{\mathrm{opt2}}\right)_{\bar{J}}\right\|_{\ell_1} \\
&\geq \left\|\mathbb{A}^{\mathrm{opt2}}{}_J\right\|_{\ell_1} - \left\|\left(Z^{\mathrm{opt1}} - Z^{\mathrm{opt2}}\right)_J\right\|_{\ell_1} \\
&\quad + \left\|\left(Z^{\mathrm{opt1}} - Z^{\mathrm{opt2}}\right)_{\bar{J}}\right\|_{\ell_1} \\
&= \left\|\mathbb{A}^{\mathrm{opt2}}{}_J\right\|_{\ell_1} - \left\|\left(\mathbb{A}^{\mathrm{opt2}} - \mathbb{A}^{\mathrm{opt1}}\right)_J\right\|_{\ell_1} \\
&\quad + \left\|\left(\mathbb{A}^{\mathrm{opt2}} - \mathbb{A}^{\mathrm{opt1}}\right)_{\bar{J}}\right\|_{\ell_1} > \left\|\mathbb{A}^{\mathrm{opt2}}{}_J\right\|_{\ell_1} \\
&= \left\|\mathbb{A}^{\mathrm{opt2}}\right\|_{\ell_1},
\end{aligned} \tag{32}$$

contradicting the hypothesis that $\mathbb{A}^{\mathrm{opt1}} \neq \mathbb{A}^{\mathrm{opt2}}$. Therefore, we conclude that $\mathrm{row}_i(E)$ is unique to problem (25). Equivalently, E is unique to the following problem:

$$\begin{aligned}
&\underset{E}{\text{minimize}} \quad \|E\|_{\ell_1} \\
&\text{subject to} \quad \overline{W} = W + E.
\end{aligned} \tag{33}$$

Furthermore, given the condition that $\|\mathrm{col}(E)_{k\in\mathcal{J}}\|_{\ell_0} \leq \kappa$, we conclude that E is unique to problem (31). □

In conclusion, Theorems 3 and 4 show that the hypothesis of our theorem holds provided that the sparse error can be uniquely corrected. Naturally, if the assumption does not hold, then neither does (13) or (31).

3.4.2. Equivalence. Next, we will discuss the conditions under which it is theoretically possible to use ℓ_1-minimization to obtain the sparse solution E (or $\mathbb{A}$) instead of ℓ_0-minimization. We derive an algorithm for precisely verifying ℓ_0-ℓ_1 equivalence. We can use the following definition and proposition proposed in [19].

Definition 5 (see [19, Definition 2]). We define $\mathcal{SK}_d(B_1)$ as the collection of all d-dimensional faces of the ℓ_1-ball B:

$$\mathcal{SK}_d(B_1) \doteq \left\{\mu \in \mathbb{R}^{mT} : \|\mu\|_{\ell_1} = 1, \|\mu\|_{\ell_0} \leq d+1\right\}, \tag{34}$$

where $B_1 \doteq \{\mu \in \mathbb{R}^{mT} : \|\mu\|_{\ell_1} \leq 1\}$.

Proposition 6 (see [19, Proposition 3]). *In a cyber-physical system, let S_{i2}, $\widetilde{W}_i$, and Ψ_i be specified as above. For every $E_i \in \mathbb{R}^T$ and $S_{i2}^T\widetilde{W} \in \mathbb{R}^{T-1}$, the following implication holds:*

$$\begin{aligned}
\left\|S_{i2}^T\widetilde{W} - S_{i2}^T\Psi_i E_i^*\right\|_{\ell_0} \leq \frac{1}{2}\mathscr{C}_i \Longrightarrow E_i^* \\
= \underset{E_i}{\mathrm{argmin}} \left\|S_{i2}^T\widetilde{W} - S_{i2}^T\Psi_i E_i\right\|_{\ell_1},
\end{aligned} \tag{35}$$

if and only if $\forall \mu \in \mathcal{SK}_{\mathscr{C}_i-1}(B_1), \forall S^T\widetilde{W} \in \mathbb{R}^T \setminus 0$ and $\|\mu + S_{i2}^T\Psi_i S^T\widetilde{W}\|_{\ell_1} > 1$, where $\mathscr{C}_i$ = (number of columns of $S_{i2}^T\Psi_i$ that are linearly independent).

Proof. See Proposition 3 in [19]. □

Note that implication (35) is the condition that we want to verify. As we need to deal with high-dimensional matrices (e.g., $E \in \mathbb{R}^{m\times T}$), we need to give asymptotic guarantees of equivalence, which is described in Proposition 6. In our experiments, it is confirmed that we can benefit from this equivalence, even when the matrices are in high dimensions.

4. Experimental Results

4.1. Case Study: Power Network. We employ a real-world power grid system as the test system we used. A state-space control model in a smart grid consists of buses connected to transmission lines. We use the IEEE 14-bus system as the test system [20]. Moreover, we use the real load data in year 2016 from New York Independent System Operator (NYISO). The NYISO load data include the 11 regions (namely, A-H). Similar to [12], the following procedures are used to estimate 5-minute system state (x) using load pattern from NYISO.

(1) Link each load bus of IEEE 14-bus system to one region of NYISO using the following matrix:

$$\begin{pmatrix} 2 & 3 & 4 & 5 & 6 & 9 & 10 & 11 & 12 & 13 & 14 \\ \mathrm{F} & \mathrm{C} & \mathrm{I} & \mathrm{B} & \mathrm{G} & \mathrm{K} & \mathrm{E} & \mathrm{H} & \mathrm{J} & \mathrm{D} & \mathrm{A} \end{pmatrix}. \tag{36}$$

The first row of the matrix is the bus number of IEEE 14-bus system and the second row represents the corresponding NYISO region index.

(2) Normalize the load data collected from NYISO to the initial real and reactive load of the corresponding IEEE 14-bus system. Due to lack of reactive load information in NYISO database, we use the direct current (DC) power flow model to estimate system states. This condition can be relaxed when the reactive load data is available.

(3) Add the normalized load data on the IEEE 14-bus system.

(4) Estimate the system state ($\hat{x}$) from the solution of power flow analysis for benchmarking purpose. In this paper, we apply Newton-Raphson algorithm for estimating $\hat{x}$.

Table 1: Regression coefficients for the predicting at 11:55 pm, Jun 30, 2016.

	$\widehat{\gamma}_1$	$\widehat{\gamma}_2$	$\widehat{\gamma}_3$
Zone A	-7.12×10^{-6}	1.02	−14.62
Zone B	-2.66×10^{-6}	1.01	−2.47
Zone C	-1.63×10^{-5}	1.05	−41.82
Zone D	-1.5×10^{-3}	2.39	−314.79
Zone E	-2.14×10^{-5}	1.03	−12.99
Zone F	-8.72×10^{-6}	1.02	−16.77
Zone G	-4.03×10^{-6}	1.01	−3.64
Zone H	-6.22×10^{-5}	1.04	−4.19
Zone I	-2.67×10^{-5}	1.04	−13.75
Zone J	-1.76×10^{-6}	1.02	−474.37
Zone K	-1.27×10^{-6}	1.01	−18.61

Similar to [12], we estimate T operating points of the system state (x) by adding the normalized 5-minute load data on the MATPOWER IEEE 14-bus case file [21]. In this paper, we use one-day NYISO data as the testing set. Thus, on one day, there will be 288 operating points. So, we set $T = 288$ to construct the F-DDIA method. Second, we prepare the *attacked* samples as follows. We let the parameter κ range from 1 to $m = 54$ in the IEEE 14-bus test system. For each κ, we simulate κ-specific meters to attempt the attack construction ($a = Hc$) with a randomly injected error c. Thus, at most, a total of 6564 labeled samples, which includes 6017 attack samples and 547 initial samples (without attacks), are prepared.

4.2. Parameters in Load Fitting. According to Section 2.3, the γ in (6) is the parameter of the measurement (load) dynamical model for power grid system. We estimate γ by polynomial regression using data traces of $\overline{z}[k+1] - \overline{z}[k]$. The historical load data in NYISO and attack samples prepared in previous session are used to construct the matrix $\widehat{\gamma}$ (i.e., polynomial regression in order of l) in (6). The measurement dynamics at each time k are estimated by the data of 24 hours prior to the time. For example, if we want to estimate the load dynamics at 0:05 am Jun 30, Zone F, the load data samples (which may contain attacks) during 0:05 am, Jun 29–0:00 am, Jun 30 are used.

We are concerned about what the regression order l is appropriate for fitting the dynamics of the system. The experimental results show that $l = 2$ is a suitable regression order. As the increase of l will improve the load fitting accuracy at the cost of computation time, we will use $l = 2$ in the rest of our experiments. Table 1 gives the regression results for predicting the dynamical model by using the load data on Jun 30, 2016.

Specifically, we take Zone F for an example; Figure 1 shows a quadratic polynomial fit of load in Zone F with 95% confidence bounds (the 95% interval indicates that we have a 95% chance that a new observation will fall within the bounds.). We collect the hourly data to fit the model, where the blue "+" represents the actual hourly load, and the green curve describes the fitting model.

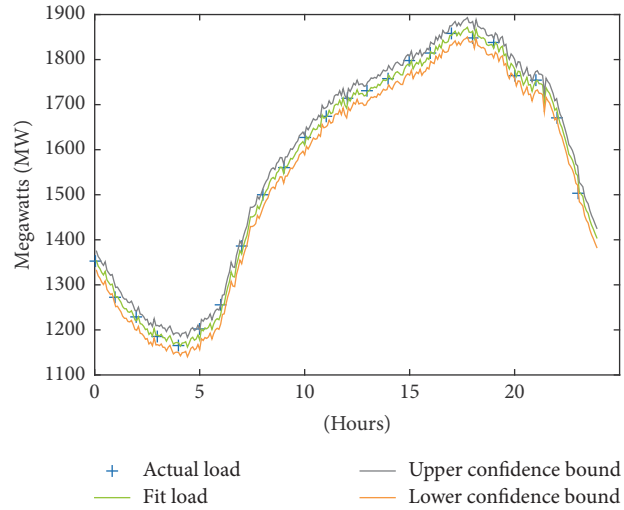


Figure 1: The quadratic polynomial fit of the load data in Zone F with 95% confidence bounds on Jun 30, 2016, when $l = 2$.

4.3. Performance Matrices. When $\mathbb{A}$ is calculated by our detector, we set the following rule to identify whether the system is attacked:

$$\mathscr{D}_i[k] = \begin{cases} 1 & \left|\mathbb{A}_{i+11k}\right| \geq \sigma_{\text{ob}} \times \left|\overline{Z}_{i+11k}\right| \\ 0 & \text{otherwise,} \end{cases} \tag{37}$$

where σ_{ob} is the observation threshold when detecting data injection attacks. The parameter σ_{ob} will be discussed later in this section. We denote the user-defined threshold $\mathscr{D}_i[k] = 1$ when $\overline{z}_i[k]$ is identified as attacked. Then, we identify whether $\overline{z}[k]$ is attacked by aggregating the values of $\mathscr{D}_i[k]$ ($i \in \mathscr{P}$). We predict $\overline{z}[k]$ as *attacked* (denoted as Label$[k] = 1$) if the sum of $\mathscr{D}_i[k]$ is larger than the all-users-defined threshold $\mathscr{N}_a$, and *secure* (denoted as Label$[k] = 0$) otherwise:

$$\text{Label}[k] = \begin{cases} 1 & \sum_{i=1}^{m} \mathscr{D}_i[k] > \mathscr{N}_a \\ 0 & \text{otherwise.} \end{cases} \tag{38}$$

In smart grid networks, the major concern is not only the detection of attack cases but also that of the secure cases. In other words, after following the rule (38), we need to be careful of the samples with high precision and recall performance in order to avoid false alarms. Therefore, we utilize precision and recall metrics, which are commonly used for classification tasks [10]. Specifically, as Table 2 defines, we denote CA as the number of attacked samples, which we identified as *attacked*, WA as the number of secure samples, which we identified as *attacked*, CS as the number of secure samples, which we identified as *secure*, and WS as the number of attacked samples, which we identified as *secure*.

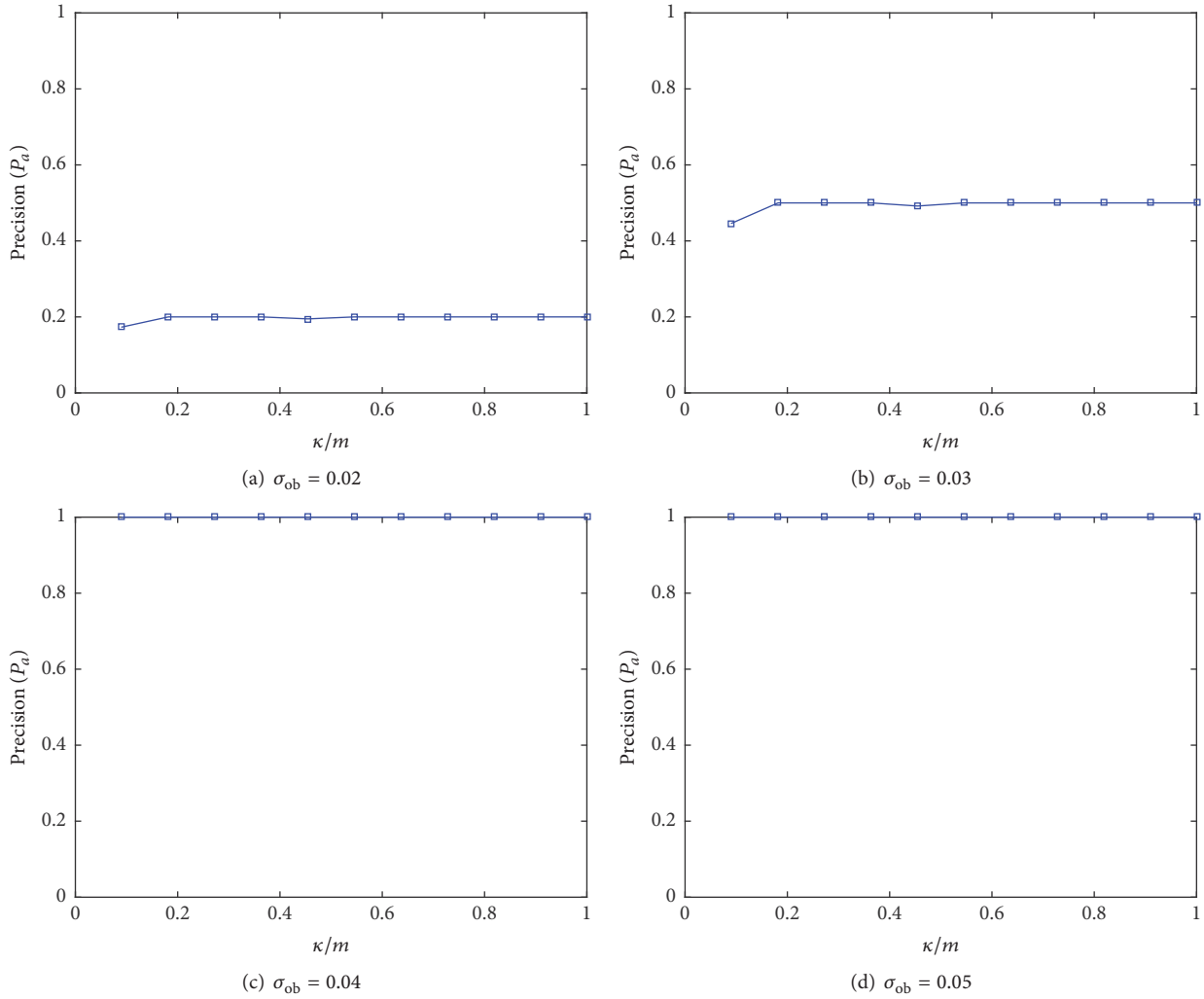


Figure 2: Precision of *attacked* samples for the IEEE 14-bus system.

Table 2: Denotations for defining evaluation metrics.

	Attacked	Secure
Classified as attacked	CA	WA
Classified as secure	WS	CS

In addition, the performance of the proposed detector can be measured by the precision and recall metrics:

$$P_a = \frac{\text{CA}}{\text{CA} + \text{WA}},$$
$$R_a = \frac{\text{CA}}{\text{CA} + \text{WS}},$$
$$P_s = \frac{\text{CS}}{\text{CS} + \text{WS}}, \quad (39)$$
$$R_s = \frac{\text{CS}}{\text{WA} + \text{CS}},$$

where P_a (P_s) and R_a (R_s) indicate the precision and recall values for the class *attacked (secure)*, respectively. Precision values give information about the decision performance of the algorithms among identified class. And recall values measure the degree of attack retrieval.

4.4. Performance on Detecting Attacks. We first analyze the performance of the proposed algorithm against the attacks, which are made from a set of false data injection attacks when $\kappa = 1$. In the experiments, we observe that the selection of threshold parameter σ_{ob} does affect the precision and recall performances. Table 3 shows the comparison for different σ_{ob} values. P_a and R_s increase as σ_{ob} increases and remain 100% when $\sigma_{ob} \geq 0.04$. In addition, R_a and P_s decrease as σ_{ob} increases. Note that the precision value at $\sigma_{ob} = 0.01$ is 7.14% and the recall value at $\sigma_{ob} > 0.06$ is lower than 50% for class *attacked*. Thus, the optimal σ_{ob} value should be in range $[0.02, 0.06]$. Note that the performance at $\sigma_{ob} = 0.05$ is quite similar to that at $\sigma_{ob} = 0.06$; thus we do not draw the performance at $\sigma_{ob} = 0.06$ in Figures 2, 3, 4, and 5 to avoid unreadability.

The performance of different σ_{ob} values for identifying *attacked* samples is compared in Figures 2 and 3, where

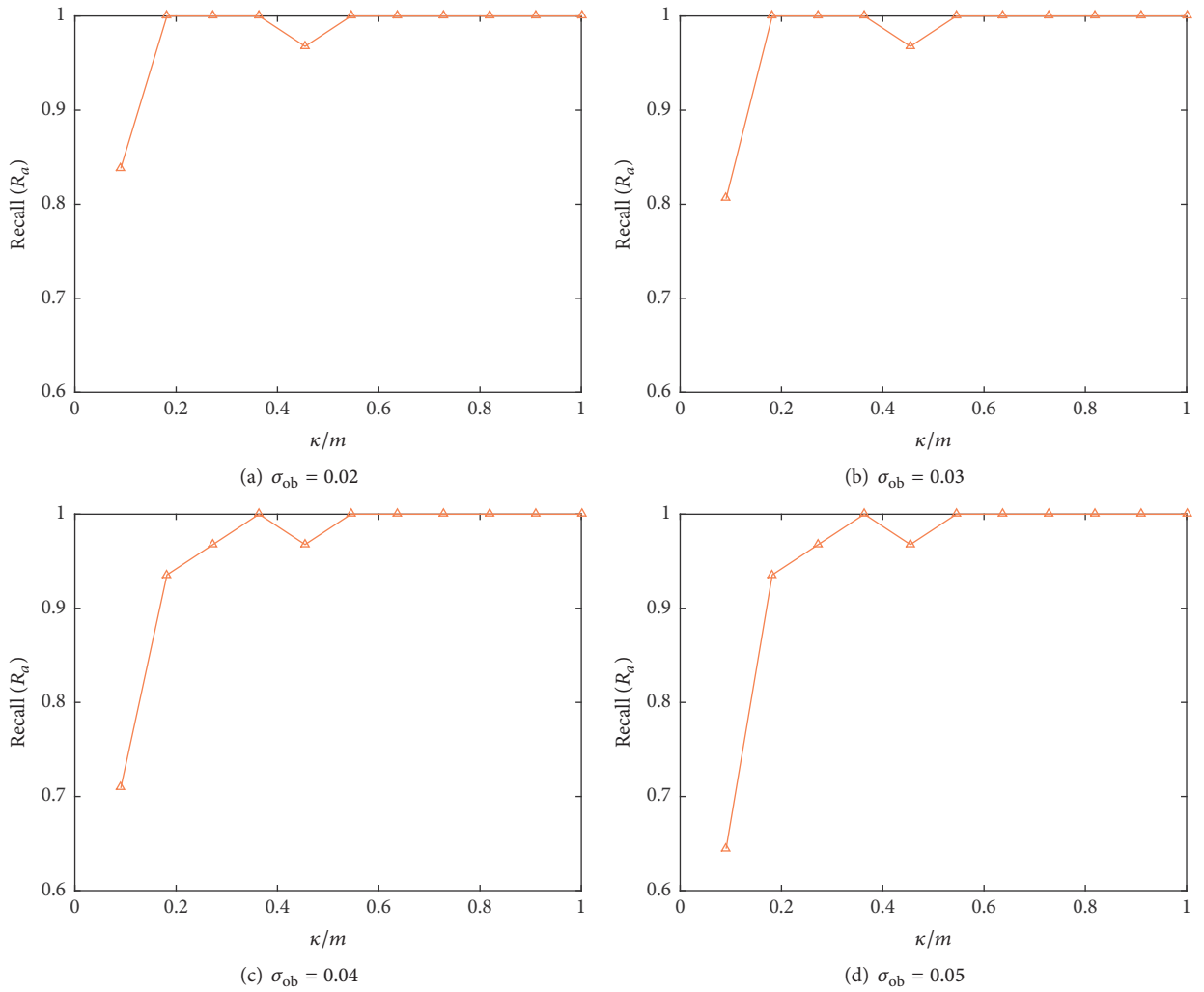


Figure 3: Recall of *attacked* samples for the IEEE 14-bus system.

Table 3: Performance of proposed detector against multiperiod attacks for IEEE 14-bus system, $\kappa = 1$.

σ_{ob}	P_a	R_a	P_s	R_s
0.01	7.14%	100%	100%	95.47%
0.02	**17.33%**	**83.87%**	**99.94%**	**98.61%**
0.03	**44.64%**	**80.65%**	**99.93%**	**99.65%**
0.04	**100%**	**70.97%**	**99.90%**	**100%**
0.05	**100%**	**64.52%**	**99.88%**	**100%**
0.06	100%	63.64%	99.87%	100%
0.07	100%	41.67%	99.80%	100%
0.08	100%	45.45%	99.81%	100%
0.09	100%	41.67%	99.80%	100%
0.10	100%	38.10%	99.78%	100%
0.20	100%	9.10%	99.68%	100%

$\kappa/m \in [0, 1]$. We observe that P_a increases and R_a decreases when σ_{ob} increases. The precision value of *attacked* class is approximately 100% when $\sigma_{ob} = 0.04$ and $\sigma_{ob} = 0.05$. The recall value of the *attacked* class increases with rising κ/m values and is approximately 100% when κ/m is larger than 54.55%. Although the proposed algorithm at $\sigma_{ob} = 0.02$ and $\sigma_{ob} = 0.03$ may correctly detect the *attacked* samples as κ/m increases, the *secure* variables are incorrectly labeled as *attacked* and therefore give more false alarms.

Meanwhile, the performance of identifying *secure* samples is compared in Figures 4 and 5. Both values (precision and recall) of the *secure* class are high (i.e., near 100%). Summing up, the above experimental results show that if we choose the parameter $\sigma_{ob} \in [0.04, 0.06]$, our methodology can efficiently detect the data injection attacks.

4.5. Performance on Recovering System States. In this part, we compare the performances of our detector and the residual-based approach with the performance of recovering the initial systems states. We first introduce how we evaluate the performances of an algorithm. In IEEE 14-bus system, the state vector x will have 14 bus voltage magnitudes and 13 phase angles, where the phase angle of one reference bus is set as the reference. If the system is observable [14], the state vector x can be represented as follows:

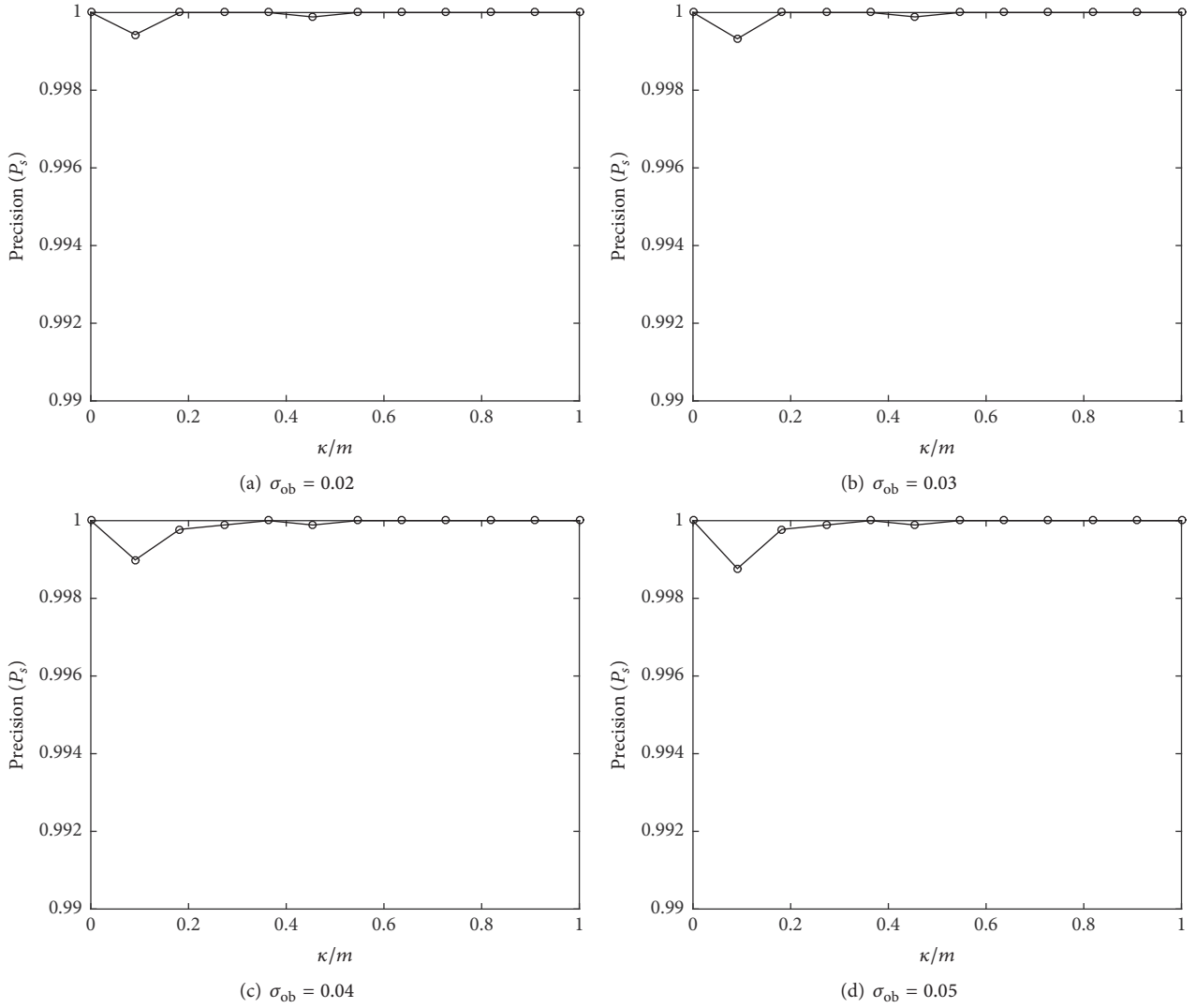


Figure 4: Precision of *secure* samples for the IEEE 14-bus system.

$x = (V_1, V_2, \ldots, V_{14}, \theta_2, \theta_3, \ldots, \theta_{13})^T$, where V_i, θ_i is voltage magnitude and voltage angle at bus i. Therefore, the average absolute phase error for bus i, denoted as $\zeta_{\theta_i[k]}$, can be described as follows:

$$\zeta_{\theta_i[k]} = \frac{1}{\vartheta}\sum_{j=1}^{\vartheta}\zeta_{\theta_i^j[k]} = \frac{1}{\vartheta}\sum_{j=1}^{\vartheta}\left|\frac{\widehat{\theta}_i^j[k] - \theta_i[k]}{\theta_i[k]}\right|, \quad (40)$$

where

ϑ is number of testing samples;

$\zeta_{\theta_i^j[k]}$ ith is bus absolute phase error at time k when under the jth attack in the testing samples;

$\widehat{\theta}_i^j[k]$ ith is bus recovered phase angle at time k when under the jth attack in the testing samples;

$\theta_i[k]$ ith is bus true phase angle at time k.

The proposed algorithm and residual-based algorithm have been tested under various attack scenarios (i.e., $\kappa = 1, 2, \ldots, 11$). Table 4 presents the results when $\kappa = 1$ and $\kappa = 11$, respectively. We can first see the superiority of our methodology, comparing with the residual-based algorithm. For example, when $\kappa = 11$, the average phase error of our proposed algorithm on bus 12 is 0.2886, whereas the error is 4.2793 for the residual-based algorithm, which is 13 times larger than our algorithm. Second, we can see that the average phasor errors for $\kappa = 1$ are in general smaller than those for $\kappa = 11$, which means that the performances of both algorithms work better when κ is small. Third, we see that the F-DDIA result of bus 11 (or bus 13) is quite different from the that of bus 12 (or 14). We think the reason that causes this phenomenon is because of the κ value. When $\kappa = 1$, the bus indexes 11–14 are with little difference. To sum up, the reason that causes this difference of the average absolute phase error is complex, and thus the F-DDIA performance depends on a series of parameters (i.e., κ, σ_{ob}, etc.).

4.6. Comparison on Execution Time. In our experiments, we find that the proposed approach is faster than other

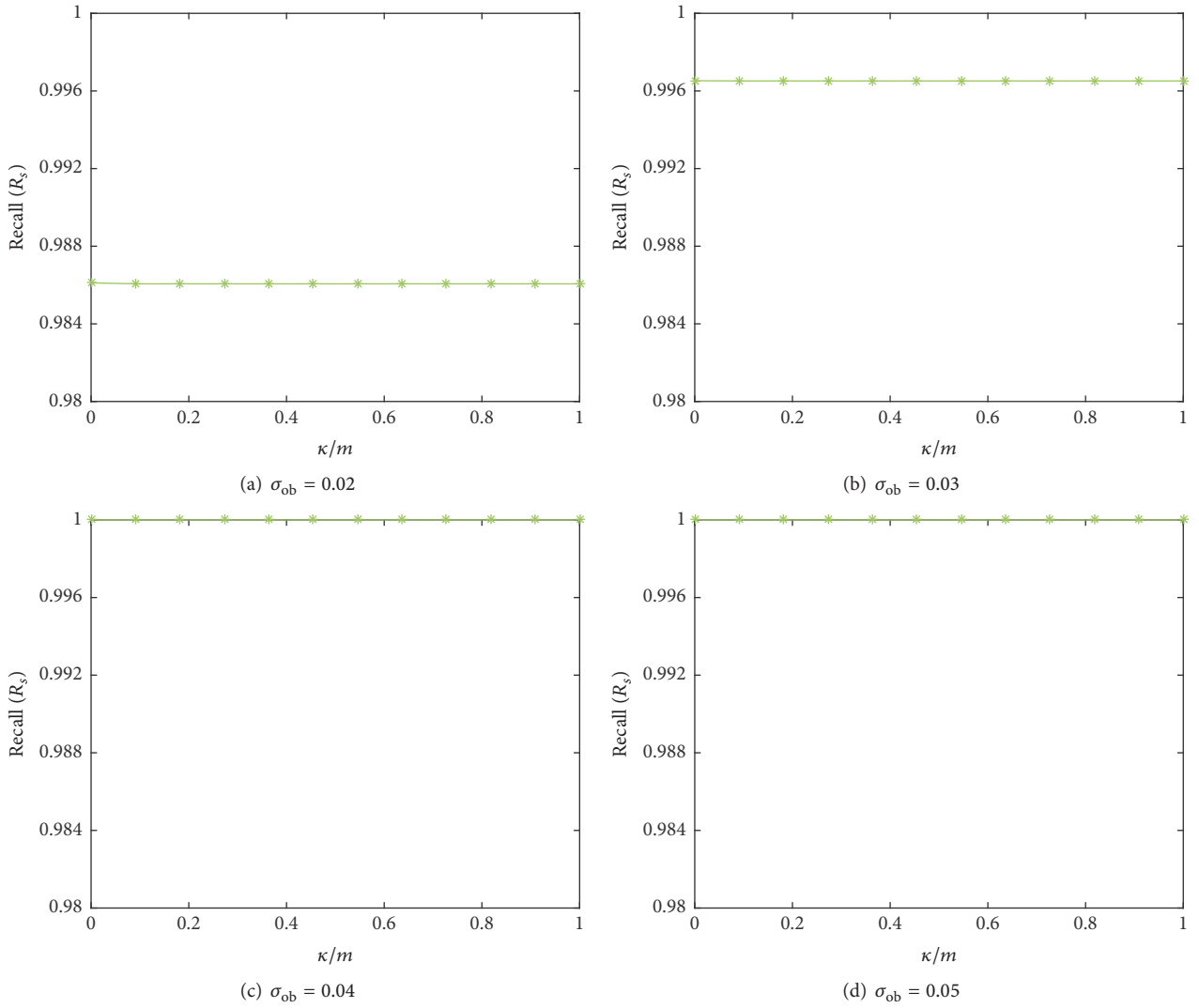


Figure 5: Recall of *secure* samples for the IEEE 14-bus system.

Table 4: $\zeta_{\theta_i[k]}$ for IEEE 14-bus system when $\sigma_{ob} = 0.05$.

Bus index	$\zeta_{\theta_i[k]}$ ($\kappa = 1$)		$\zeta_{\theta_i[k]}$ ($\kappa = 11$)	
	Residual-based	Our detector	Residual-based	Our detector
2	0.2388	0.0770	0.5855	0.1387
3	0.7420	0.3612	1.2131	0.1374
4	0.3274	0.0430	0.9988	0.2117
5	0.4026	0.1076	0.9927	0.2187
6	1.0687	0.3478	2.0091	0.7807
7	0.6875	0.0493	1.6335	0.4186
8	0.6875	0.0493	1.6335	0.4186
9	0.8095	0.0443	1.8286	0.4823
10	0.9589	0.1817	1.7800	0.7220
11	0.9869	0.2925	1.8136	1.6674
12	1.2767	0.4810	4.2793	0.2886
13	1.1997	0.4977	2.1074	1.1148
14	1.0374	0.2894	1.6876	0.3817

works. The residual-based fault detector takes around 50 min (0.043 s per sample), while the proposed approach only takes 12 min (0.011 s per sample). The 12 min of our approach includes load dynamics fitting and Douglas-Rachford iterations process. The main computation burden for our proposed approach is to proceed Douglas-Rachford iterations for basis pursuit process. In general, we do not consider the state estimation process. This is why our proposed approach is faster than the other one.

5. Conclusions

The paper examines the problem of detecting data injection attacks in smart grid networks. We propose a detection framework named F-DDIA, which can recover the initial system state, as well as the real measurement readings. Due to the sparse nature of data injection attacks, ℓ_1 minimization technique (including Douglas-Rachford) can be applied. The validation of the proposed detecting algorithm is validated using load data from NYISO. Our detector works well in both linear and nonlinear systems.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported in part by National High Technology Research and Development Program of China (no. 2015AA016008).

References

[1] R. Langner, "Stuxnet: dissecting a cyberwarfare weapon," *IEEE Security and Privacy*, vol. 9, no. 3, pp. 49–51, 2011.

[2] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security*, vol. 14, no. 1, article 13, 2011.

[3] J. Kim, L. Tong, and R. J. Thomas, "Data framing attack on state estimation," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 7, pp. 1460–1470, 2014.

[4] J. Wang, L. C. K. Hui, S. M. Yiu, X. Cui, E. K. Wang, and J. Fang, "A survey on the cyber attacks against non-linear state estimation in smart grids," in *Proceedings of the Australasian Conference on Information Security and Privacy (ACISP '16)*, Springer, Berlin, Germany, 2016.

[5] J. Wang, L. C. K. Hui, S. M. Yiu, E. K. Wang, and J. Fang, "A survey on cyber attacks against nonlinear state estimation in power systems of ubiquitous cities," *Pervasive and Mobile Computing*, vol. 39, pp. 52–64, 2017.

[6] L. Liu, M. Esmalifalak, Q. Ding, V. A. Emesih, and Z. Han, "Detecting false data injection attacks on power grid by sparse optimization," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 612–621, 2014.

[7] Q. Hu, D. Fooladivanda, Y. H. Chang, and C. J. Tomlin, "Secure state estimation and control for cyber security of the nonlinear power systems," 2016, https://arxiv.org/abs/1603.06894.

[8] D. Han, Y. Mo, and L. Xie, "Robust state estimation against sparse integrity attacks," 2016, https://arxiv.org/abs/1601.04180.

[9] T. T. Kim and H. V. Poor, "Strategic protection against data injection attacks on power grids," *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 326–333, 2011.

[10] J. Wang, W. Tu, L. C. K. Hui, S. M. Yiu, and E. K. Wang, "Detecting time synchronization attacks in cyber-physical systems with machine learning techniques," in *Proceedings of the In 37th IEEE International Conference on Distributed Computing Systems (ICDCS '17)*, 2017.

[11] G. Hug and J. A. Giampapa, "Vulnerability assessment of AC state estimation with respect to false data injection cyber-attacks," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1362–1370, 2012.

[12] G. Chaojun, P. Jirutitijaroen, and M. Motani, "Detecting False Data Injection Attacks in AC State Estimation," *IEEE Transactions on Smart Grid*, 2015.

[13] L. Demanet and X. Zhang, "Eventual linear convergence of the Douglas-Rachford iteration for basis pursuit," *Mathematics of Computation*, vol. 85, no. 297, pp. 209–238, 2016.

[14] A. Abur and A. G. Exposito, *Power System State Estimation: Theory and Implementation*, CRC Press, Boca Raton, Fla, USA, 2004.

[15] J. Wang, L. C. K. Hui, and S. M. Yiu, "Data framing attacks against nonlinear state estimation in smart grid," in *Proceedings of the IEEE Global Communications Conference Workshop (GLOBECOM '15)*, 2015.

[16] C. R. Goodall, *13 Computation Using The QR Decomposition*, Handbook of Statistics, 1993.

[17] E. J. Candes and T. Tao, "Decoding by linear programming," *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[18] J. Zhang, C. Zhao, D. Zhao, and W. Gao, "Image compressive sensing recovery using adaptively learned sparsifying basis via L0 minimization," *Signal Processing*, vol. 103, pp. 114–126, 2014.

[19] Y. Sharon, J. Wright, and Y. Ma, "Computation and relaxation of conditions for equivalence between l_1 and l_0 minimization," Tech. Rep. UILU-ENG-07-2208, University of Illinois, Urbana-Champaign, Illinois, Ill, USA, 2007.

[20] S. K. M. Kodsi and C. A. Canizares, "Modeling and simulation of ieee 14 bus system with facts controllers," Tech. Rep., University of Waterloo, Waterloo, Ontario, Canada, 2003.

[21] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.

Hindawi
Security and Communication Networks
Volume 2017, Article ID 1790268, 13 pages
https://doi.org/10.1155/2017/1790268

Research Article

Vague Sets Security Measure for Steganographic System Based on High-Order Markov Model

Chun-Juan Ouyang,[1,2] Ming Leng,[1,2] Jie-Wu Xia,[1,2] and Huan Liu[1,2]

[1]*Key Laboratory of Watershed Ecology and Geographical Environment Monitoring of NASG, Jinggangshan University, Ji'an 343009, China*
[2]*School of Electronics and Information Engineering, Jinggangshan University, Ji'an 343009, China*

Correspondence should be addressed to Chun-Juan Ouyang; oycj001@163.com

Received 26 April 2017; Accepted 12 June 2017; Published 6 August 2017

Academic Editor: Yushu Zhang

Security measure is of great importance in both steganography and steganalysis. Considering that statistical feature perturbations caused by steganography in an image are always nondeterministic and that an image is considered nonstationary, in this paper, the steganography is regarded as a fuzzy process. Here a steganographic security measure is proposed. This security measure evaluates the similarity between two vague sets of cover images and stego images in terms of *n-order* Markov chain to capture the interpixel correlation. The new security measure has proven to have the properties of boundedness, commutativity, and unity. Furthermore, the security measures of zero order, first order, second order, third order, and so forth are obtained by adjusting the order value of *n-order* Markov chain. Experimental results indicate that the larger n is, the better the measuring ability of the proposed security measure will be. The proposed security measure is more sensitive than other security measures defined under a deterministic distribution model, when the embedding is low. It is expected to provide a helpful guidance for designing secure steganographic algorithms or reliable steganalytic methods.

1. Introduction

Security of the steganographic system is the fundamental issue in the field of the information hiding. Image steganography is the technique of hiding information in digital image and trying to conceal the existence of the secret information. The image with and the image without hidden information are called stego image and cover image, respectively [1]. Steganography and steganalysis are in a hide-and-seek game [2]. They try to defeat each other and also develop with each other. In recent years, steganalysis researches have made much head-way [3, 4], and many attempts have been made to build up secure steganographic algorithms [5–8]. Up until now, there is no standard security measure for steganographic system. The security of the steganography always depends on the encryption of the steganography, which contradicts Kerckhoffs' principle [9]. Hence, it is very necessary to study the security measure which can provide guidance for designing the high-secure steganography and steganalytic algorithms with high performance.

Now, the study of the security measure becomes one of the hotspots in the steganography research field. Researchers have put forward their views from different viewing angels. From the point of view of information theory, Cachin [10] proposed a security measure in terms of the relative entropy between the probability mass functions (PMF) of the cover images and the stego images. Sullivan et al. [11] employed the divergence distance of the empirical matrices to define the security measure. They modeled the sequence of image pixels as first-order Markov chain which could capture one adjacent pixel dependency. Furthermore, Zhang et al. [12] models the images pixels as n-order Markov chain to provide the security measure. Based on game theory, Liu et al. [13] presented that the counterwork relationship is modeled between steganography side and attack side. In [14], Schöttle and Böhme studied adaptive steganography while taking the

knowledge of the steganalyst into account. Liu and Tang [15] also provided the security for the adaptive steganography. In [16], Chandramouli et al. proposed an alternative security measure based on steganalyzer's ROC (Receiver Operating Characteristic) performance. From the point of feature space, Pevný and Fridrich [17] provided the MMD (Maximum Mean Discrepancy) by employing a high-dimensional feature space set as the covers models.

The security measures mentioned above all assume that accurate statistical estimations can be obtained from the finite data samples. However, an image is a nonstationary process; its local statistical correlation will change when image is changed slightly. So the statistical features change is nondeterministic after steganography processing. Meanwhile, for a steganographic system, the warden is lack of the knowledge of the cover distribution. Thus, the distribution estimates of the cover and stego image are not stable. So the security measures defined under the deterministic statistical model are hard to apply due to the lack of the accurate distribution.

To address this problem, we regard the steganography as a fuzzy and indeterministic process. The goal of this paper is to provide a practical security measure in terms of the vague sets similarity measure between cover images and the stego images. Particularly, the sequence of image pixels is modeled as an *n-order* Markov chain to capture the interpixel correlations. The main contributions of this work are as follows:

(1) We derive a security measure for a steganographic system which is different from the deterministic ones. The existing security measures are defined by evaluating the difference between cover images and stego images. In contrast, the new security measure is defined by evaluating the similarity between cover images and their stego version.

(2) The *n-order* security measure based on vague sets similarity measure is proven to have the properties of the boundedness, commutativity, and unity. The properties guarantee the security measure is indeed a real distance which indeed satisfies the symmetry and triangle inequality. The boundedness guarantees the new benchmark can measure the steganographic security.

(3) Simulation results verify the effectiveness of the new security measure by benchmarking several popular steganographic schemes. When embedding rate is low, the new security measure is more sensitive to reveal the statistical features change than other security measures. Thus, the proposed security measure can provide a better guidance for the design of steganography and steganalysis.

The rest of the paper is organized as follows. Section 2 gives a review of the two security measures with the deterministic statistical distribution model and introduces the *n-order* Markov chain model. The *n-order* secure measure based on vague sets similarity measure is presented in detail in Section 3. Experimental results are provided in Section 4 to demonstrate the effectiveness and the superiority of the proposed security measure. We draw our conclusions in Section 5.

2. Steganographic Security and Cover Model

2.1. Security Measure Based on Kullback-Leibler (K-L) Divergence. Suppose C is the set of all the covers, and it is an assumption that the selections of the covers and stegos from the set C can be described by the random variables c and s on C with the probability mass functions (PMF) P_c and P_s, respectively. Cachin [10] quantified the security of a steganographic system in terms of the Kullback-Leibler (K-L) divergence (sometimes called relative entropy); that is,

$$D\left(P_c \parallel P_s\right) = \sum_{x \in X} P_c\left(x\right) \log \frac{P_c\left(x\right)}{P_s\left(x\right)}, \tag{1}$$

where X is the set of possible pixel values. A steganographic system is called perfectly secure if (1) is zero or ε-secure if $0 \le D(P_c \parallel P_s) \le \varepsilon$ is satisfied. The K-L divergence provides a simple yet convenient method for measuring the difference between cover images and stego images.

In fact, we have little information about the PMF involved due to the large dimensionality of the set C. So the security measure is usually defined with simplified cover models, such as independent and identically distributed (i.i.d.) ones. The security measure of K-L divergence calculates the difference from the view of the first-order statistical features (such as one-dimensional histogram feature).

2.2. Security Measure Based on Divergence Distance. To account for the dependence of the pixels, Sullivan et al. [11] employed the first-order Markov chain model to capture the interpixel correlation. The divergence distance was used to quantify the statistical feature perturbations introduced by a steganography between the two empirical matrices of cover images and stego images. Suppose C and S are two random sequences of the cover image pixels and the stego image pixels, respectively, obtained by a given scanning method. Let M^c and M^s be the empirical matrixes of C and S, respectively. The divergence distance is given by

$$D\left(M^c, M^s\right) = \sum_{i,j \in R} M_{ij}^c \log\left(\frac{M_{ij}^c}{\sum_j M_{ij}^c} \frac{\sum_j M_{ij}^s}{M_{ij}^s}\right), \tag{2}$$

where $M_{ij}^c / \sum_j M_{ij}^c$ and $M_{ij}^s / \sum_j M_{ij}^s$ are the transition probabilities of cover images and stego images, respectively. The transition probability is commonly calculated by the ratio of the total number to the pixel changes from value i to value j over the total number of possible pixel changes (e.g., for an 8-bit image, the total possible pixel changes number is 256×256). The constant R is the range of all possible pixel values. Thus, the divergence distance provides the difference between cover images and their stego version from the view of the second-order statistical features (such as two-dimensional histogram feature and difference histogram feature).

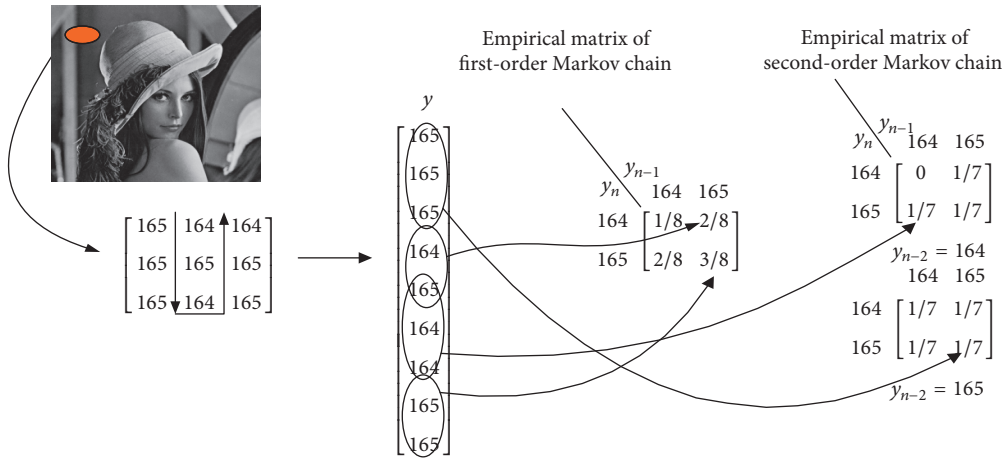


Figure 1: The generating process of the empirical matrixes of first-order and second-order Markov chain.

The two security measures mentioned above are defined based on the Shannon information theory under the assumption that the image data statistical distribution is deterministic. Most of the security measures proposed later are also defined under the same assumption. However, the image data shows the sceneries in the aspects of gray, texture, shape, and so forth. There are many a kind of indeterministic factors (such as noise) in a steganography process. Therefore, the security measures with the deterministic statistical distribution model cannot measure the security accurately.

2.3. n-Order Markov Chain Model. The weakness of the above two security measure lies in the fact that the image model such as i.i.d and first-order Markov are too simple to capture interpixel dependency. Therefore, here we model the sequence of image pixels as an n-order Markov chain. The *n-order* Markov chain is a random sequence indexing the image pixels scanned by a given mode. For instance, when $n = 2$, the second-order Markov chain accounting for two adjacent pixels' correlation meets the following condition:

$$P\left(Y_m \mid Y_{m-1}, Y_{m-2}, \ldots, Y_1\right) = P\left(Y_m \mid Y_{m-1}, Y_{m-2}\right). \quad (3)$$

There are at least two reasons for us to select *n-order* Markov chain model. First, the model is flexible. When $n = 0$, it turns out to be the i.i.d model, in which the image pixels are assumed to be unrelated. When $n = 1$, the first-order Markov chain can capture only one adjacent pixel dependence. Furthermore, the n-order Markov chain can capture more interpixel relationships among the pixels when $n \geq 2$. Second, compared with the Markov random field model [9], the Markov chain model, though simple, is able to calculate the statistical estimation of the image samples. For *n-order* Markov chain, it is easy to calculate the realistic statistical estimates using the empirical matrixes. In the following, we construct the empirical matrixes of the first-order and second-order Markov chain.

Let $\{Y_n,\ n = 1, 2, \ldots, L\}$ be an *n-order* Markov chain on the finite set ω, where Y_n is the n-indexed set of pixels obtained by a row, column, zigzag, or Hilbert scanning method. ω is the possible gray scale values. When $n = 1$, the first-order Markov chain source is defined by the transition matrixes $T_{i_1,i_2} = P(Y_n = i_1 \mid Y_{n-1} = i_2)$ and marginal probabilities $p_{i_1} = P(Y_n = i_1)$. For a realization, $y = (y_1, y_2, \ldots, y_L)^T$. Let η_{i_1,i_2} be the number of transitions from values i_1 to i_2 in y. The empirical matrixes are $M_1(y) = \eta_{i_1,i_2}(y)/(L-1)$. That is, the i_1, i_2 element represent the proportion of spatially adjacent pixel pairs with the grayscale value of i_1 followed by i_2. Thus the empirical matrixes provide an estimation of the transition matrixes and marginal probabilities. The empirical matrixes are similar to the concurrence matrixes of the image. It can be recognized as a matrix form of the two-dimensional normalized histogram for estimating the joint probability mass function (PMF) of a source image. Similarly, when $n = 2$, we can get the empirical matrixes of the second-order Markov chain, denoted by $M_2(y) = \eta_{i_1,i_2,i_3}(y)/(L-1)$. $\eta_{i_1,i_2,i_3}(y)$ is the number of transitions from values i_1 to i_3 via i_2 in y. For an 8-bit image, the size of the empirical matrixes $M_2(y)$ is 256×256×256. The element of the empirical matrixes represents the proportion of spatially adjacent pixel group with a grayscale value of i_1 followed by i_2 and i_3. A simple example of generating the empirical matrixes of first-order and second-order Markov chain is shown in Figure 1.

In Figure 1, the small block is derived from the standard image "Lena." Its size is 3 × 3, including pixels 164 and 165. The example image pixels are scanned vertically. The size of the empirical matrixes of first-order Markov chain in Figure 1 is 2 × 2. The element represents the proportion of spatially adjacent pixel pairs with (164, 164), (164, 165), (165, 164), and (165, 165). The right-hand side of Figure 1 demonstrates the procedure of the empirical matrixes of second-order Markov chain. Its size is 2 × 2 × 2, in which the element represents the proportion of spatially adjacent pixel groups with (164, 164, 164), (164, 165, 164), (165, 164, 164), (165, 165, 164), and so forth.

Since the cover sources are strongly correlated, the probabilities of two adjacency samples are equal or nearly equal. As a result, in the empirical matrixes, the masses are more

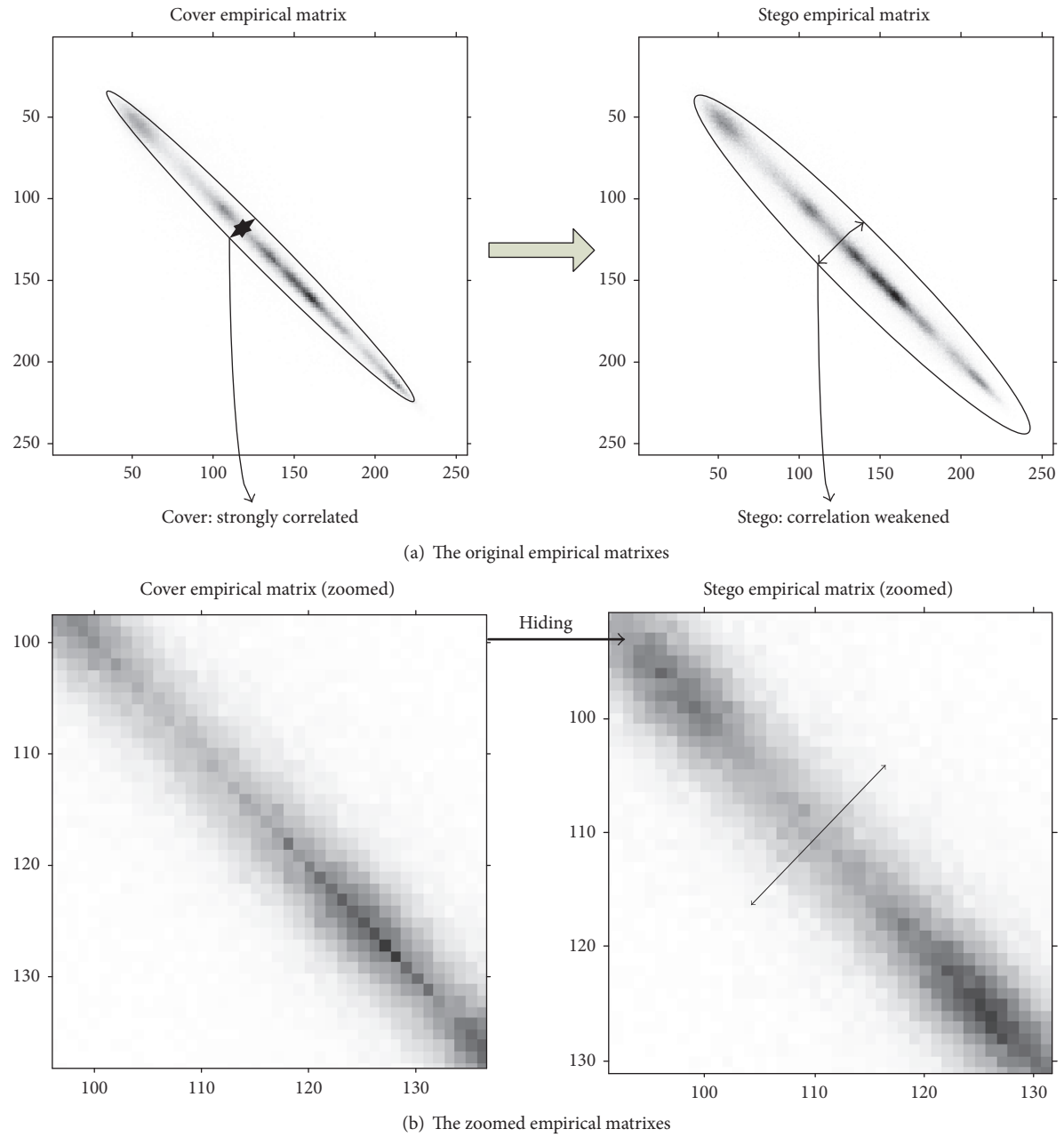


Figure 2: Empirical matrixes of a cover image and its stego image.

concentrated near the main diagonal in a correlated source. In [18], Harmsen and Pearlman considered that information hiding can be viewed as adding the additive noise to the cover image. The secret information (additive noise) is uncorrelated after hiding, and its empirical matrixes spread evenly over the main diagonal. Thus we see that hiding weakens the dependencies among the cover samples, which is illustrated in Figure 2(a). Figure 2(b) is part of the zoomed empirical matrixes. According to the above analysis, the steganography tends to spread the density of the pixels pairs away from the main diagonal of the empirical matrixes. This property may shed some light on designing of the security measure for a steganographic system. Thus, in Section 3, we will propose an *n-order* security measure in terms of the vague sets similarity measure by modeling the sequence of images pixels as an *n-order* Markov chain.

3. Security Measure Based on Vague Sets Similarity Measure

The vague sets similarity measure [19, 20] describes the matching degree of two vague sets. In a practical steganographic system, there are many indeterministic factors introduced by steganography. In this work, we regard the responding probability distribution sets of the cover samples and the stego samples as two discrete vague sets. Then a new security measure is proposed below in terms of vague sets similarity

measure to measure the similarity between cover images and stego images.

3.1. Vague Sets. Roughly speaking, a fuzzy set is a class with fuzzy boundaries. The fuzzy set A is a class of objects X along with a grade of membership function $\mu_A(x)$, $x \in X$. It assigns a single value to each object. This single value combines the evidence for x ∈ X and the evidence against $x \in X$. And it is only a measure of the pros/cons evidence. However, in many practical applications we often require pros and cons evidence simultaneously. Gau and Buehrer [21] advanced the concept of vague sets. The vague sets theory adopts a true membership function t_A and a false membership function f_A to record the lower bounds on μ_A. These lower bounds are used to create a subinterval on [0, 1], namely, $[t_A(x_i), 1-f_A(x_i)]$, to generalize $\mu_A(x_i)$ of fuzzy sets, where $t_A(x_i) \le \mu_A(x_i) \le 1 - f_A(x_i)$. Vague sets expand the value of the membership function to a subinterval of [0, 1] instead of a single value; thus it has stronger ability to reveal the indeterminacy than the fuzzy set theory. The related definitions of vague sets are as follows.

Definition 1 (vague sets). Let X be the universe of discourse, $X = \{x_1, x_2, \ldots, x_n\}$. $V(x)$ denotes all the vague sets of X, $\forall A \in V(x)$. The vague set A is characterized by a true membership function t_A and a false membership function f_A:

$$\begin{aligned} t_A &: X \longrightarrow [0,1], \\ f_A &: X \longrightarrow [0,1], \end{aligned} \tag{4}$$

where $t_A(x_i)$ is the lower bound on the grade of membership of x_i derived from the evidence for x_i. $f_A(x_i)$ is a lower bound on the negation of x_i derived from the evidence against x_i, satisfying $t_A(x_i) + f_A(x_i) \le 1$. The grade of membership of x_i is bounded to a subinterval $[t_A(x_i), 1-f_A(x_i)]$ of [0, 1]. When X is discrete, a vague set A can be written as

$$A = \sum_{i=1}^{n} \frac{\left[t_A(x_i), 1 - f_A(x_i)\right]}{x_i}, \quad x_i \in X. \tag{5}$$

Definition 2. Let X be the universe of discourse, $X = \{x_1, x_2, \ldots, x_n\}$. A and B are two vague sets of X. The entropy of the vague set A, $E(A)$, is defined as

$$E(A) = -\frac{1}{n \ln 2} \sum_{i=1}^{n} \left[t_A(x_i) \ln t_A(x_i) + f_A(x_i) \ln f_A(x_i)\right]. \tag{6}$$

Definition 3. Let X be the universe of discourse, $X = \{x_1, x_2, \ldots, x_n\}$. A and B are two vague sets of X. The partial entropy of vague set A against vague set B, $E_B(A)$, is defined as

$$E_B(A) = -\sum_{i=1}^{n} \left[t_B(x_i) \ln t_A(x_i) + f_B(x_i) \ln f_A(x_i)\right]. \tag{7}$$

3.2. The n-Order Security Measure Based on Vague Sets Similarity Measure. As discussed in Section 2.3, the n-order Markov chain model can capture sufficient inherent correlations. Additionally, the changes in image statistical features, introduced by steganography, are indeterministic. Therefore, in the new security measure, we model the sequence of the image pixels as an *n-order* Markov chain. Simultaneously, the empirical matrixes of the *n-order* Markov chain of cover images and stego images are regarded as two vague sets. Then the n-order security measure based on the vague sets similarity measure is defined as follows.

Suppose C and S are n-order Markov chain sequence of cover images and stego images, respectively, and then scan them by a given mode (such as horizontal, vertical, zigzag, and Hilbert mode). MC and MS represent the corresponding empirical matrixes. $m_{i_1,i_2,\ldots,i_{n+1}}$, the element of empirical matrixes, denotes the joint probability distribution from pixels i_1 to i_{n+1} via the states of $i_2, i_3, \ldots$ and i_n. The $i_1, i_2, \ldots, i_{n+1}$ is the image pixel value, $i \in [0, 255]$. G denotes the set of all possible values of $m_{i_1,i_2,\ldots,i_{n+1}}$. Let $M_{i_1,i_2,\ldots,i_{n+1}}$ be the universe of discourse composed of $m_{i_1,i_2,\ldots,i_{n+1}}$. Then MC and MS are two vague sets on $M_{i_1,i_2,\ldots,i_{n+1}}$. That is,

$$\begin{aligned} MC &= \frac{\sum_{i=0}^{255}\left[t_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right), 1 - f_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\right]}{m_{i_1,i_2\cdots i_{n+1}}}, \\ &\qquad m_{i_1,i_2\cdots i_{n+1}} \in M_{i_1,i_2\cdots i_{n+1}}, \\ MS &= \frac{\sum_{i=0}^{255}\left[t_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right), 1 - f_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\right]}{m_{i_1,i_2\cdots i_{n+1}}}, \\ &\qquad m_{i_1,i_2\cdots i_{n+1}} \in M_{i_1,i_2\cdots i_{n+1}}. \end{aligned} \tag{8}$$

Definition 4. Let $M_{i_1,i_2,\ldots,i_{n+1}}$ be the universe of discourse. MC and MS are two vague sets of $M_{i_1,i_2,\ldots,i_{n+1}}$. The similarity measure $T_n(MC, MS)$ between the vague sets MC and MS is defined as the *n-order* secure measure for a steganographic system; that is,

$$T_n(MC, MS) = \frac{m \ln 2 \left(E(MC) + E(MS)\right)}{E_{MC}(MS) + E_{MS}(MC)}, \tag{9}$$

where $E(MC)$ and $E(MS)$ denote the entropy of the vague set MC and MS, respectively; $E_{MC}(MS)$ stands for the partial entropy of vague set MS against vague set MC; $E_{MS}(MC)$ is the partial entropy of vague set MC against vague set MS. $E(MC)$ and $E_{MC}(MS)$ can be written as

$$\begin{aligned} E(MC) &= -\frac{1}{m \ln 2} \\ &\quad \cdot \sum_{i_1,i_2\cdots i_{n+1} \in G} \left[t_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right) \ln t_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\right. \\ &\quad \left. + f_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right) \ln f_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\right], \\ E_{MS}(MC) &= -\sum_{i_1,i_2\cdots i_{n+1} \in G} \left[t_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right) \ln t_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\right. \\ &\quad \left. + f_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right) \ln f_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\right]. \end{aligned} \tag{10}$$

Similarly, $E(MS)$ and $E_{MC}(MS)$ can be written as

$$\begin{aligned} E(MS) &= -\frac{1}{m \ln 2} \\ &\cdot \sum_{i_1, i_2 \cdots i_{n+1} \in G} \left[t_{MS}\left(m_{i_1, i_2 \cdots i_{n+1}}\right) \ln t_{MS}\left(m_{i_1, i_2 \cdots i_{n+1}}\right) \right. \\ &\left. + f_{MS}\left(m_{i_1, i_2 \cdots i_{n+1}}\right) \ln f_{MS}\left(m_{i_1, i_2 \cdots i_{n+1}}\right) \right], \\ E_{MC}(MS) &= -\sum_{i_1, i_2 \cdots i_{n+1} \in G} \left[t_{MC}\left(m_{i_1, i_2 \cdots i_{n+1}}\right) \ln t_{MS}\left(m_{i_1, i_2 \cdots i_{n+1}}\right) \right. \\ &\left. + f_{MC}\left(m_{i_1, i_2 \cdots i_{n+1}}\right) \ln f_{MS}\left(m_{i_1, i_2 \cdots i_{n+1}}\right) \right]. \end{aligned} \tag{11}$$

Moreover, a steganographic system is called perfectly secure if $T_n(MC, MS) = 1$ or ε-secure if $T_n(MC, MS) = \varepsilon$, $\varepsilon \in (0, 1)$. $T_n(MC, MS) = 0$.

Theorem 5. *Let $T_n(MC, MS)$ be the n-order secure measure of a steganographic system based on vague set similarity measure. Then $T_n(MC, MS)$ satisfies the following.*

(1) Boundedness is

$$0 \le T_n(MC, MS) \le 1. \tag{12}$$

(2) Commutativity is

$$T_n(MC, MS) = T_n(MS, MC). \tag{13}$$

(3) Unity is

$$\begin{aligned} T_n(MC, MS) = 1 &\Longleftrightarrow \\ MC &= MS. \end{aligned} \tag{14}$$

$T_n(MC, MS)$ provides a security measure for a steganographic system in terms of the similarity between cover images and stego images. $T_n(MC, MS)$ is limited in a finite interval of $[0, 1]$, where 1 denotes "perfectly secure," while 0 denotes "definitely unsecure." However, other security measures under the deterministic statistical model calculate the difference between cover images and stego images. The values range in an infinite interval $[0, \infty)$. The property of the boundedness guarantees the proposed security measure can measure a steganographic algorithm quantitatively. Hence, it has stronger ability to reveal the statistical changes of the cover images. When $n = 0$, the image pixels distribution is said to be i.i.d., and $T_0(MC, MS)$ is called the zero-order security measure. When $n = 1$, the sequence of image pixels is considered to be a first-order Markov chain, and $T_1(MC, MS)$ is defined as the first-order security measure. Thus, a different order security measure can be obtained by adjusting the value of n.

4. Experimental Results and Discussion

In this section, we report experimental results that demonstrate the capability of the new security measure. First of all in Section 4.1 the image databases used for the experiment are described. Afterwards, in Section 4.2, we benchmark several different steganographic methods with *n-order* security measure based on vague sets, with particular attention to the effectiveness of low embedding rate. Finally, we compare the proposed security measure with previously used benchmarks designed under the deterministic statistical model.

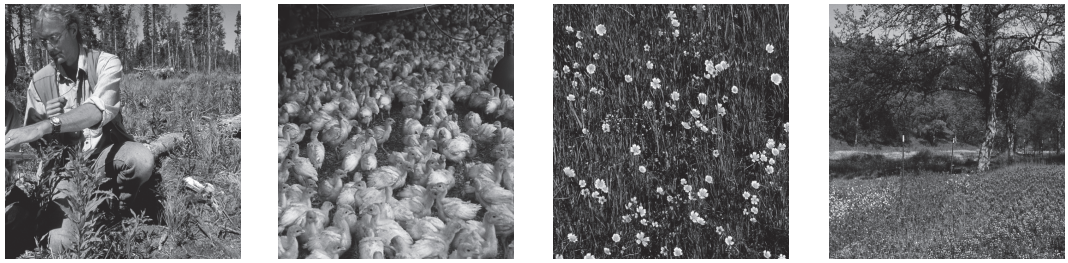
4.1. Image Database. For the experimental validation we used two image databases. The first one is BOWS2 [22] image database including 10000 grayscale images with fixed size 512×512. The other one is NRCS Photo Gallery [23]. We selected 1500 images from NRCS Photo Gallery. All images were converted into grayscale and central cropped to a size of 512×512 for experimental purposes. The images in our experiments show a wide range of scenarios including house, manmade objects, and animal. Some images are shown in Figure 3.

4.2. Verification of the Effectiveness of the Proposed Security Measure. To evaluate the performance of the proposed method for measuring the security of the steganographic algorithms, the new security measure with different orders is used to measure the security of different steganographic algorithms with different embedding rates. First, we select some spatial-domain steganographic algorithms, including LSBM (least significant bit matching) [24], LSB ± 2, HUGO [25] (highly undetectable steganography). We use 2000 images from BOWS2 image database; all the images are grayscale with the fixed size 512×512. As discussed in Section 2.3, first-order and second-order Markov chain models have captured sufficient interpixel correlations. Additionally, considering the computation complexity, we use the zero-order, first-order, and second-order security measure based on vague sets to measure the LSBM, LSBM2, and HUGO steganographic methods with the embedding rate ranging from 0.1 bpp (bits per pixel) to 1 bpp in a step size of 0.1 bpp. The average measure results for zero-order, first-order, and second-order security measure of 2000 images with different embedding rates are depicted in Figure 4.

In Figure 4, all curves indicate that the value of security measure gradually decreases with an increase in the embedding rate for the same steganographic algorithm. It is consistent with the definition of the security measure based on vague sets. Its value is limited in an interval of $[0, 1]$, where 1 denotes "perfectly secure" for the steganographic system. Hence the value of the *n*-order security measure satisfies monotonic decreasing property; that is, the higher the security of the stego images, the larger the value of the security measure. Furthermore, as is evident in Figure 4, the values of the same order security measure are different for different stego schemes with the same embedding rate. Note that LSB ± 2 obtains the lowest value in Figure 4, implying that it is most unsecure among the three hiding methods under the same condition. On the contrary, HUGO gains the highest value. All the measure results are coincident with the theoretical analysis of the three embedding schemes.

Furthermore, in order to evaluate the measuring ability of different order security measures, we compare the security

(a) Some images of BOWS2



(b) Some images of NRCS

Figure 3: Some images of image database.

for the same steganographic algorithm using different order security measures. Figure 6 shows the average measure results of zero-order, first-order, and second-order security measure for LSBM, LSB ± 2, and HUGO, respectively. In fact, all the data in Figure 5 is derived from Figure 4. As demonstrated in Figure 5, for the same steganographic method, the values of the zero-order, first-order, and second-order security measure are different at the same embedding rate. It is demonstrated that the value of the first-order security measure is smaller than that of the zero-order measure but larger than that of the second-order measure for the same steganographic method with the same embedding rate. The experiments show that the second-order security measure provides the largest measure interval to reveal the security change of the cover images with the embedding rate ranging from 0.1 bpp to 1 bpp. So we can conclude that second-order security measure can provide more obvious statistical distributed changes caused by steganography.

To further verify the effectiveness of the proposed security measure. We used it to benchmark JPEG steganographic algorithms schemes on different database. And we focus on low payloads to see if any of the test steganographic schemes becomes distinguishable by using the vague sets security measure with finite image sample.

We selected 1500 images from NRCS Photo Gallery. All images were converted into grayscale and central cropped to a size of 512 × 512 for experimental purposes. The images were embedded with pseudorandom payloads with 5%, 10%, 15%, and 20% bpac (bits per nonzero AC coefficient). The tested stego schemes include F3, F5 without shrinkage (nsF5) [26], Model Based Steganography without deblocking (MB1) [27], and Model Based Steganography with deblocking (MB2) [28]. The cover images were single-compressed JPEGs with quality factor 70. The measure results using zero-order, first-order, and second-order security measure based on vague sets are showed in Table 1. The data in Table 1 indicates that, for the same steganography, the larger the embedding rate, the lower the value of the same security measure. It also exhibited that, for the same steganography, the higher the order of the security measure, the smaller the value of the security measure, suggesting that second-order security measure can get a value lower than the other two security measures under the same condition.

The data in Table 1 also shows, according to the same order security measure, the MB2 is the least statistically detectable, followed by MB1 and nsF5, while F3 is the most detectable. All the measure results are coincident with the theoretical security among adopted stego algorithms. In a word, the experimental results indicate that the proposed security measure is effective for measuring the security for different steganographic methods on different image database. Meanwhile, the greater the order, the stronger the measure ability of the security measure.

4.3. Comparison with Security Measure under Deterministic Statistical Model. To show the superiority of the proposed security measure $T_n(MC, MS)$, we compare it with two security measures under the deterministic statistical model. One is the Kullback-Leibler (K-L) divergence between the probability mass functions (PMF) proposed by Anderson [9], denoted by $D(P_c \parallel P_s)$. The other, denoted as $D(M_c, M_s)$, is the divergence distance between the two empirical matrices proposed by Cachin [10]. To be unbiased, the zero-order measure $T_0(MC, MS)$ is compared with $D(P_c \parallel P_s)$ when

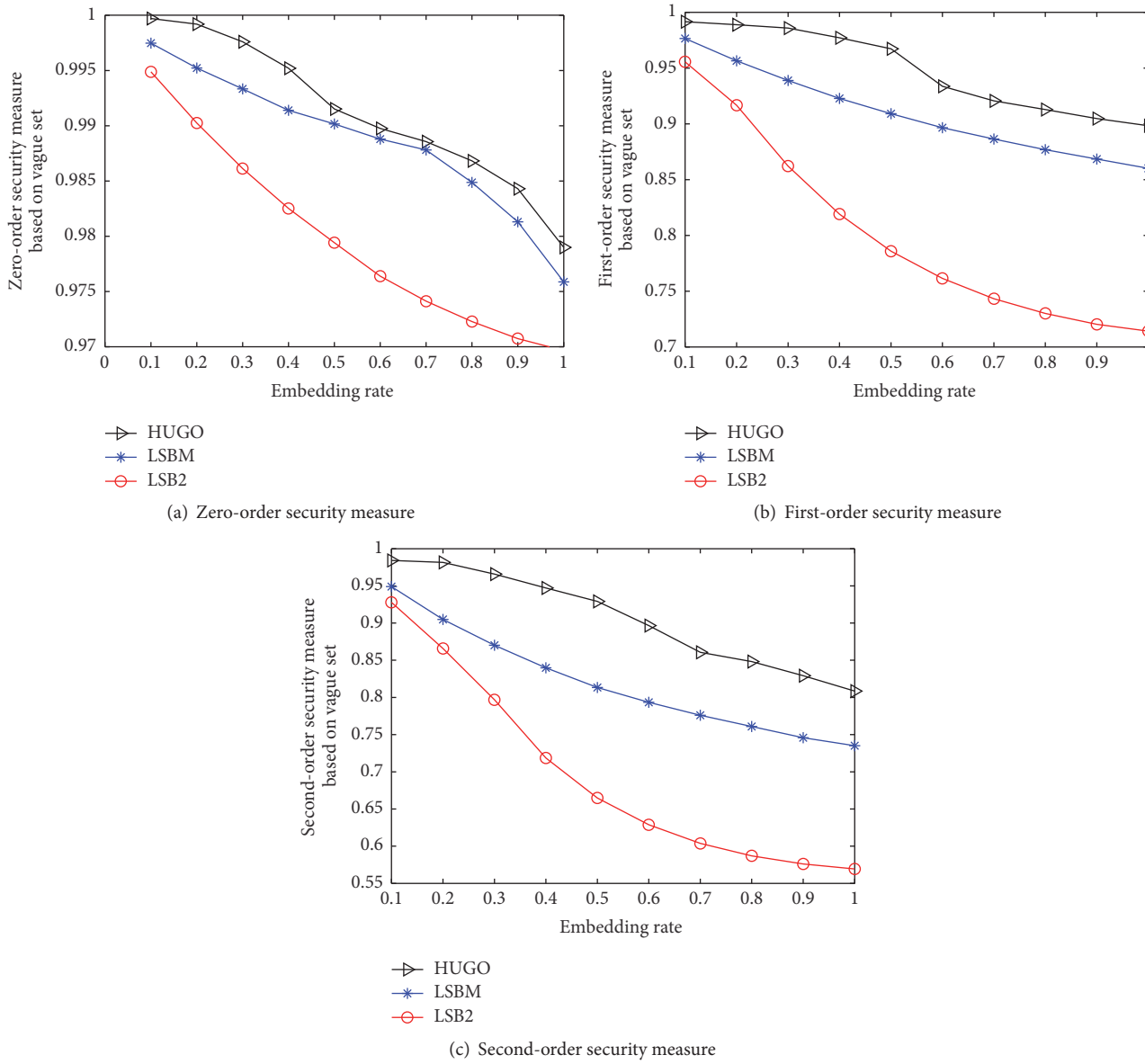


Figure 4: The same order security measure for different steganographic methods with different embedding rate.

$D(P_c \parallel P_s)$ is used under the assumption that the image model is i.i.d. Similarly, the first-order measure $T_1(MC, MS)$ is compared with $D(M_c, M_s)$ since their image pixel sequences are all modeled as the first-order Markov chain. In the experiments, the same 2000 images from BOWS2 are adopted. $T_0(MC, MS)$, $D(P_c \parallel P_s)$, $T_1(MC, MS)$, and $D(M_c, M_s)$ are used to measure the security of the HUGO with the embedding rate ranging from 0.05 bpp to 1 bpp in a step size of 0.05 bpp. Figures 6(a) and 6(b) show the average measure of $T_0(MC, MS)$ and $D(P_c \parallel P_s)$ with different embedding rates, respectively. The average measure values of $T_1(MC, MS)$ and $D(M_c, M_s)$ are also illustrated in Figures 7(a) and 7(b), respectively.

Looking at Figures 6 and 7, we see that the value of security measure based on vague sets decreases as the embedding rate increases, whereas the value of security measure under the deterministic distribution model increases as the embedding rate increases. All the curves in Figures 6 and 7 indicate that both the security measure models are effective in measuring the security of the steganography. In order to show the superiority of the proposed security measure, we define $\delta = \Delta y/y$ as the sensitivity of, where Δy is the security measure variation of a given embedding rate change range, andyis the total security measure variation of the embedding rate change. Obviously, Figures 6(b) and 7(b) demonstrate that δ of security measure is very small when embedding rate is lower than 0.5 bpp. So its corresponding

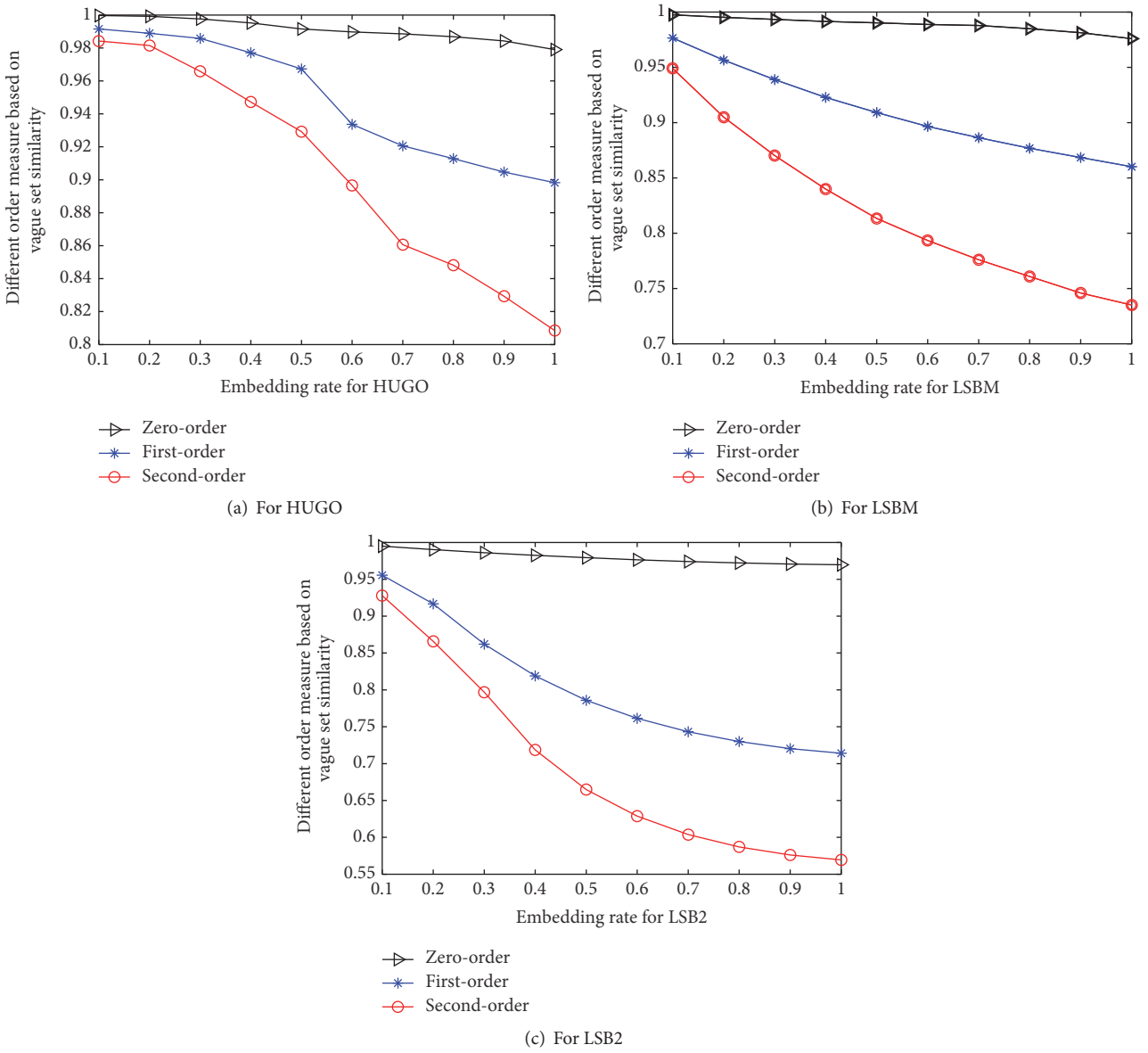
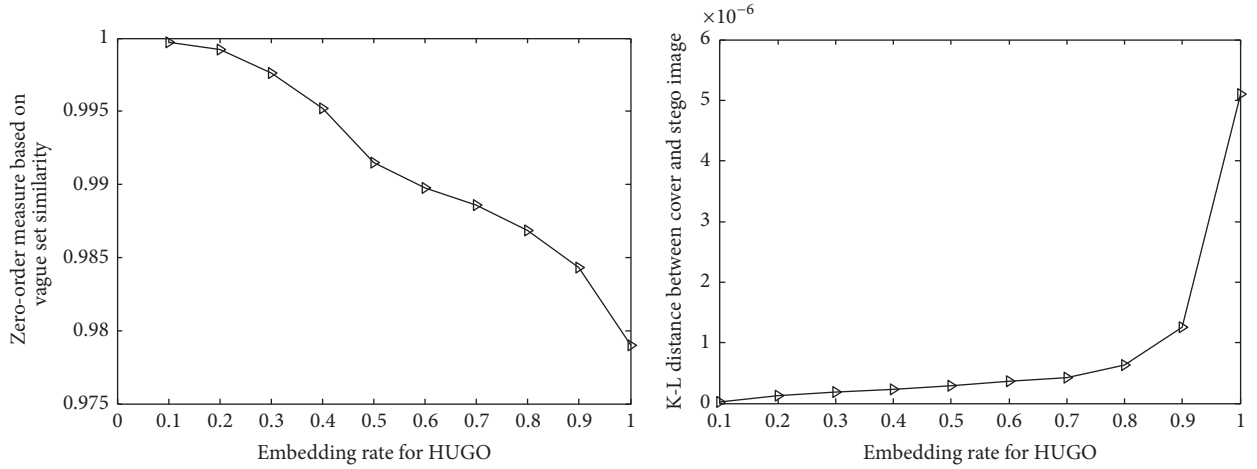


Figure 5: The different order security measures for the same steganography with different embedding rate.

security measure is not sensitive to the statistical distribution change. Hence, the new security measure can reveal more obvious statistical change than the security measures under deterministic statistical distribution model when embedding rate is low.

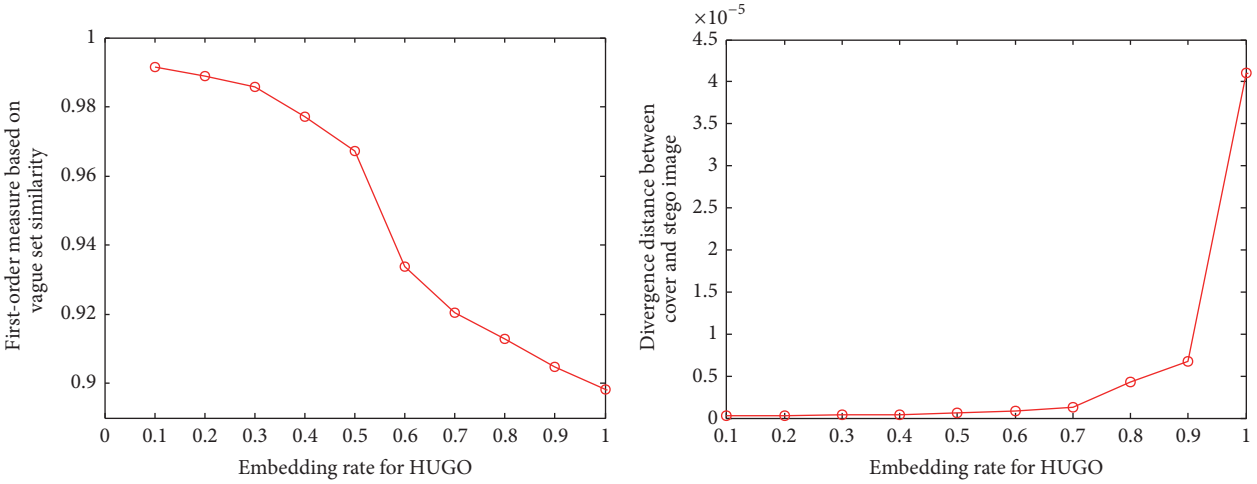
5. Conclusions

Vague sets similarity measure is a simple yet effective tool for measuring the similarity between two vague sets. In this work, a novel security measure for a steganographic system in terms of the vague sets similarity measure is proposed to measure the similarity between cover images and stego images. Particularly, in the new security measure, the sequence of image pixels is modeled as an *n-order* Markov chain to capture sufficient interpixel dependencies. The proposed security measure is proven to have such properties as boundedness, commutativity, and unity. Various order security measures can be obtained by adjusting the value of n. Experimental results confirm the effectiveness of the proposed security measure for evaluating different steganographic algorithms. Meanwhile, the security measure with a higher order always has a better measure ability. Additionally, when the embedding rate is low, the *n-order* security measure based on vague sets is more sensitive than other security measures under the deterministic distribution model. Considering the computational complexity and steganalytic ability, two issues should be tackled in our further research. One is how to use the *n-order* security measure to design reliable steganalytic methods by extracting the statistical feature from the empirical matrixes. The other is



(a) $T_0(MC, MS)$ (b) $D(P_c \parallel P_s)$

Figure 6: $T_0(MC, MS)$ and $D(P_c \parallel P_s)$ for HUGO with different embedding rates.



(a) $T_1(MC, MS)$ (b) $D(M_c, M_s)$

Figure 7: $T_1(MC, MS)$ and $D(M_c, M_s)$ for HUGO with different embedding rates.

how to use the new security measure to design highly secure steganographic algorithms.

Appendix

Proof of Theorem 5. (1)

$$
\begin{aligned}
&E_{MC}(MS) + E_{MS}(MC) - m\ln 2\,(E(MC) + E(MS)) \\
&\quad = -\sum_{i_1,i_2\cdots i_{n+1}\in G}\Big[t_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\ln t_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right) \\
&\quad + f_{MC}\left(m_{ij}\right)\ln f_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\Big] \\
&\quad - \sum_{i_1,i_2\cdots i_{n+1}\in G}\Big[t_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\ln t_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right) \\
&\quad + f_{MS}\left(m_{ij}\right)\ln f_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\Big] \\
&\quad + \sum_{i_1,i_2\cdots i_{n+1}\in G}\Big[t_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\ln t_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right) \\
&\quad + f_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\ln f_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\Big] \\
&\quad + \sum_{i_1,i_2\cdots i_{n+1}\in G}\Big[t_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\ln t_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right) \\
&\quad + f_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\ln f_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\Big] \\
&\quad = \sum_{i_1,i_2\cdots i_{n+1}\in G}\left[t_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\ln\frac{t_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}{t_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}\right. \\
&\quad \left. + f_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\ln\frac{f_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}{f_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}\right]
\end{aligned}
$$

Table 1: Different order vague sets security measure for different steganography methods.

Steganography method	Embedding rate (bpac)	Zero-order	First-order	Second-order
F3	5%	0.9768	0.9662	0.9569
	10%	0.9755	0.9647	0.9569
	15%	0.9714	0.9608	0.9498
	20%	0.9683	0.9584	0.9477
nsF5	5%	0.9865	0.9736	0.9593
	10%	0.9847	0.9711	0.9542
	15%	0.9840	0.9687	0.9531
	20%	0.9818	0.9656	0.9515
MB1	5%	0.9994	0.9879	0.9785
	10%	0.9991	0.9866	0.9699
	15%	0.9965	0.9849	0.9673
	20%	0.9959	0.9837	0.9656
MB2	5%	0.9999	0.9868	0.9687
	10%	0.9996	0.9842	0.9624
	15%	0.9987	0.9922	0.9617
	20%	0.9982	0.9818	0.9609

$$+\sum_{i_1,i_2\cdots i_{n+1}\in G}\left[t_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\ln\frac{t_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}{t_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}+f_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\ln\frac{f_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}{f_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}\right]. \tag{A.1}$$

Since the inequality satisfies $\ln x \geq (1-1/x)$, we have

$$\begin{aligned}
&E_{MC}(MS)+E_{MS}(MC)-m\ln 2\left(E(MC)+E(MS)\right)\\
&\geq\sum_{i_1,i_2\cdots i_{n+1}\in G}\left[t_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\cdot\left(1-\frac{t_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}{t_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}\right)+f_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\cdot\left(1-\frac{f_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}{f_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}\right)\right]\\
&+\sum_{i_1,i_2\cdots i_{n+1}\in G}\left[t_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\cdot\left(1-\frac{t_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}{t_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}\right)+f_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\cdot\left(1-\frac{f_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}{f_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}\right)\right]\\
&\geq\sum_{i_1,i_2\cdots i_{n+1}\in G}\left[t_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)-t_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)+f_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)-f_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\right]\\
&+\sum_{i_1,i_2\cdots i_{n+1}\in G}\left[t_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)-t_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)+f_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)-f_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\right]=0.
\end{aligned} \tag{A.2}$$

Hence $E_{MC}(MS)+E_{MS}(MC)-m\ln 2(E(MC)+E(MS))\geq 0$, such that $E_{MC}(MS)+E_{MS}(MC)\geq m\ln 2(E(MC)+E(MS))$.

Since $t_{MC}(m_{i_1,i_2\cdots i_{n+1}})$, $t_{MS}(m_{i_1,i_2\cdots i_{n+1}})$, $f_{MC}(m_{i_1,i_2\cdots i_{n+1}})$, and $f_{MS}(m_{i_1,i_2\cdots i_{n+1}})$ are all in the range of $[0,1]$ and $0\times\ln 0=0$, $E(MC)$, $E(MS)$, $E_{MC}(MS)$, and $E_{MS}(MC)$ are all positive.

Hence $0\leq T_n(MC,MS)\leq 1$.

(2) According to the definition of the *n-order* security measure, $T_n(MC,MS)$ is described as

$$T_n(MC,MS)=\frac{m\ln 2\left(E(MC)+E(MS)\right)}{E_{MC}(MS)+E_{MS}(MC)}. \tag{A.3}$$

And it can also be described as

$$T_n(MS,MC)=\frac{m\ln 2\left(E(MS)+E(MC)\right)}{E_{MS}(MC)+E_{MC}(MS)}. \tag{A.4}$$

Hence $T_n(MC,MS)=T_n(MS,MC)$.

(3) From the proving procedure of property (1), we have

$$E_{MC}(MS) + E_{MS}(MC) - m\ln 2\,(E(MC) + E(MS)) \geq 0, \tag{A.5}$$

$$\begin{aligned} &E_{MC}(MS) + E_{MS}(MC) - m\ln 2\,(E(MC) + E(MS)) \\ &= \sum_{i_1,i_2\cdots i_{n+1}\in G}\left[t_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\ln\frac{t_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}{t_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}\right. \\ &\left. + f_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\ln\frac{f_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}{f_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}\right] \\ &+ \sum_{i_1,i_2\cdots i_{n+1}\in G}\left[t_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\ln\frac{t_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}{t_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}\right. \\ &\left. + f_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)\ln\frac{f_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}{f_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right)}\right]. \end{aligned} \tag{A.6}$$

If and only if

$$\begin{aligned} t_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right) &= t_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right), \\ f_{MC}\left(m_{i_1,i_2\cdots i_{n+1}}\right) &= f_{MS}\left(m_{i_1,i_2\cdots i_{n+1}}\right). \end{aligned} \tag{A.7}$$

Namely, when $MC = MS$ and $MC = MS$, such that $E_{MC}(MS) + E_{MS}(MC) - m\ln 2(E(MC) + E(MS)) = 0$.

Hence $T_n(MC, MS) = 1 \Leftrightarrow MC = MS$. □

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Foundation of China (nos. 61462046, 61363014), the Science and Technology Research Projects of Jiangxi Province Education Department (nos. GJJ16079, GJJ160750), the Natural Science Foundation of Jiangxi Province (nos. 20151BAB207026, 20161BAB202050, and 20161BAB202049), Jinggangshan University Doctoral Scientific Research Foundation (nos. JZB1311, JZB15016), and Key Laboratory of Watershed Ecology and Geographical Environment Monitoring of NASG (nos. WE2015012, WE2016013).

References

[1] B. Li, J. He, J. Huang, and Y. Q. Shi, "A survey on image steganography and steganalysis," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 2, no. 2, pp. 142–172, 2011.

[2] X.-P. Zang, Z.-X. Qian, and S. Li, "Prospect of digital steganography research," *Journal of Applied Sciences-Electronics and Information Engineering*, vol. 34, no. 5, pp. 475–489, 2016.

[3] J. Fridrich and J. Kodovský, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[4] V. Holub and J. Fridrich, "Low-complexity features for JPEG steganalysis using undecimated DCT," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, article A1, pp. 219–228, 2015.

[5] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-adaptive steganography by minimizing statistical detectability," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 221–234, 2016.

[6] N. Zhou, H. Li, D. Wang, S. Pan, and Z. Zhou, "Image compression and encryption scheme based on 2D compressive sensing and fractional Mellin transform," *Optics Communications*, vol. 343, pp. 10–21, 2015.

[7] L. Guo, J. Ni, and Y. Q. Shi, "Uniform embedding for efficient JPEG steganography," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 5, pp. 814–825, 2014.

[8] N. Zhou, S. Pan, S. Cheng, and Z. Zhou, "Image compression-encryption scheme based on hyper-chaotic system and 2D compressive sensing," *Optics and Laser Technology*, vol. 82, pp. 121–133, 2016.

[9] R. Anderson, "Why information security is hard - An economic perspective," in *Proceedings of the 17th Annual Computer Security Applications Conference, ACSAC 2001*, pp. 358–365, usa, December 2001.

[10] C. Cachin, "An information-theoretic model for steganography," *Information and Computation*, vol. 192, no. 1, pp. 41–56, 2004.

[11] K. Sullivan, U. Madhow, S. Chandrasekaran, and B. S. Manjunath, "Steganalysis for Markov cover data with applications to images," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 275–287, 2006.

[12] Z. Zhang, G. J. Wang, W. Jun et al., "Steganalysis of spread spectrum image steganography based on high-order markov chain mode," *ACTA Electronica Sinica*, vol. 38, no. 11, pp. 2578–2584, 2010.

[13] G.-J. Liu, Y.-W. Dai, Y.-X. Zhao, and Z.-Q. Wang, "Modeling steganographic counterwork by game theory," *Journal of Nanjing University of Science and Technology*, vol. 32, no. 2, pp. 199–204, 2008.

[14] P. Schöttle and R. Böhme, "Game theory and adaptive steganography," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 4, pp. 760–773, 2016.

[15] J. Liu and G.-M. Tang, "Game research on large-payload and adaptive steganographic counterwork," *Acta Electronica Sinica*, vol. 42, no. 10, pp. 1963–1969, 2014.

[16] R. Chandramouli, M. Kharrazi, and N. Memon, "Image steganography and steganalysis: concepts and practice," in *Proceedings of the IWDW'03*, vol. 2939, pp. 35–49, 2003.

[17] T. Pevný and J. Fridrich, "Benchmarking for steganography," in *Information Hiding. 10th International Workshop*, pp. 251–267, Santa Barbara, Calif, USA, 2008.

[18] J. J. Harmsen and W. A. Pearlman, "Steganalysis of additive noise modelable information hiding," in *Proceedings of the IST/SPIE 15th Annu. Symp. Electronic Imaging Science Technology*, pp. 21–24, San Jose, Calif, USA, January 2003.

[19] F. Li and Z.-Y. Xu, "Measures of similarity between vague sets," *Journal of Software*, vol. 12, no. 6, pp. 922–927, 2001.

[20] S. Y. Quan, "The vague set similarity measure based on Meaning of Information," *Computer Engineering and Applications*, vol. 43, no. 25, pp. 87–89, 2007.

[21] W. L. Gau and D. J. Buehrer, "Vague sets," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 2, pp. 610–614, 1993.

[22] P. Bas, T. Filler, and T. Pevny, "Break our steganographic system—the ins and outs of organizing BOSS," in *Proceedings of the 13th International Workshop on Information Hiding*, pp. 59–70, Berlin, Germany, 2011.

[23] United States Department of Agriculture, Natural resources conservation service photo gallery, [DB/OL] http://photogallery.nrcs.usda.gov, 2002.

[24] T. Sharp, "An implementation of key-based digital signal steganography," in *Proceedings of the Information Hiding Workshop*, vol. 2137, pp. 13–26, 2001.

[25] T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6387, pp. 161–177, 2010.

[26] J. Fridrich, D. Soukal, and M. Goljan, "Maximum likelihood estimation of length of secret message embedded using ±K steganography in spatial domain," in *Proceedings of SPIE-IS and T Electronic Imaging - Security, Steganography, and Watermarking of Multimedia Contents VII*, vol. 5681, pp. 595–606, January 2005.

[27] P. Sallee, "Model-Based Steganography," in *Digital Watermarking*, T. Kalker, Ed., vol. 2939 of *Lecture Notes in Computer Science*, pp. 154–167, Springer, Berlin, Heidelberg, 2004.

[28] P. Sallee, "Model-based methods for steganography and steganalysis," *International Journal of Image and Graphics*, vol. 5, no. 1, pp. 167–189, 2005.