Secure Reversible Image Data Hiding over Encrypted Domain via Key Modulation

Jiantao Zhou, *Member, IEEE,* Weiwei Sun, *Student Member, IEEE,* Li Dong, *Student Member, IEEE,* Xianming Liu, *Member, IEEE*, Oscar C. Au, *Fellow, IEEE,* and Yuan Yan Tang, *Fellow, IEEE*

***Abstract*—This work proposes a novel reversible image data hiding (RIDH) scheme over encrypted domain. The data embedding is achieved through a public key modulation mechanism, in which access to the secret encryption key is not needed. At the decoder side, a powerful two-class SVM classifier is designed to distinguish encrypted and non-encrypted image patches, allowing us to jointly decode the embedded message and the original image signal. Compared with the state-of-the-arts, the proposed approach provides higher embedding capacity, and is able to *perfectly* reconstruct the original image as well as the embedded message. Extensive experimental results are provided to validate the superior performance of our scheme.**

***Keywords*—Reversible image data hiding (RIDH), signal processing over encrypted domain, feature extraction, SVM**

I. Introduction

Reversible image data hiding (RIDH) is a special category of data hiding technique, which ensures perfect reconstruction of the cover image upon the extraction of the embedded message. The reversibility makes such image data hiding approach particularly attractive in the critical scenarios, e.g., military and remote sensing, medical images sharing, law forensics and copyright authentication, where high fidelity of the reconstructed cover image is required.

The majority of the existing RIDH algorithms are designed over the plaintext domain, namely, the message bits are embedded into the original, un-encrypted images. The early works mainly utilized the lossless compression algorithm to compress certain image features, in order to vacate room for message embedding [1], [2]. However, the embedding capacity of this type of method is rather limited and the incurred distortion on the watermarked image is severe. Histogram shifting (HS)-based technique, initially designed by Ni *et al.* [3], is another class of approach achieving better embedding performance through shifting the histogram of some image features [4], [5]. The latest difference expansion (DE)-based schemes and the improved prediction error expansion (PEE)-based strategies were shown to be able to offer the state-of-the-art capacity-distortion performance [6]–[10].

Recently, the research on signal processing over encrypted domain has gained increasing attention, primarily driven by the needs from Cloud computing platforms and various privacy-preserving applications [11]–[14]. This has triggered the investigation of embedding additional data in the encrypted images in a reversible fashion. In many practical scenarios, e.g., secure remote sensing and Cloud computing, the parties who process the image data are un-trusted. To protect the privacy and security, all images will be encrypted before being forwarded to a un-trusted third party for further processing. For instance, in secure remote sensing, the satellite images, upon being captured by on-board cameras, are encrypted and then sent to the base station(s), as illustrated in Fig. 1. After receiving the encrypted images, the base station embeds a confidential message, e.g., base station ID, location information, time of arrival (TOA), local temperature, wind speed, etc., into the encrypted images. Eventually, the encrypted image carrying the additional message is transmitted over a public network to a data center for further investigation and storage. For security reasons, any base station has no privilege of accessing the secret encryption key K pre-negotiated between the satellite and the data center. This implies that the message embedding operations have to be conducted entirely over the encrypted domain. In addition, similar to the case of Cloud computing, it is practically very costly to implement a reliable key management system (KMS) in such multi-party environment over insecure public networks, due to the differences in ownership and control of underlying infrastructures on which the KMS and the protected resources are located [15] [1]. It is therefore much desired if secure data hiding could be achieved *without* an additional secret data hiding key shared between the base station and the data center. Also, we appreciate simple embedding algorithm as the base station usually is constrained by limited computing capabilities and/or power. Finally, the data center, which has abundant computing resources, extracts the embedded message and recovers the original image by using the encryption key K.

This work was supported in part by the Macau Science and Technology Development Fund under grants FDCT/009/2013/A1, FDCT/046/2014/A1, FDCT/100/2012/A3, in part by the Research Committee at University of Macau under grants MRG007/ZJT/2015/FST, MRG021/ZJT/2013/FST, MYRG2014-00031-FST, MYRG205(Y1-L4)-FST11-TYY and MYRG187 (Y1-L3)-FST11-TYY, and in part by the National Science Foundation of China under grants 61402547, 61300110 and 61273244.

Jiantao Zhou, Weiwei Sun, Li Dong, and Yuan Yan Tang are with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau. e-mails: jtzhou@umac.mo, mb25429@umac.mo, yb47452@umac.mo, yytang@umac.mo

Xianming Liu is with the School of Computer Science and Technology, Harbin Institute of Technology. e-mail: xmliu.hit@gmail.com

Oscar C. Au is with the Department of Electronics and Computer Engineering, Hong Kong University of Science and Technology. e-mail: eeau@ust.hk

[1] Key management challenges in the Cloud have been thoroughly studied in [15]

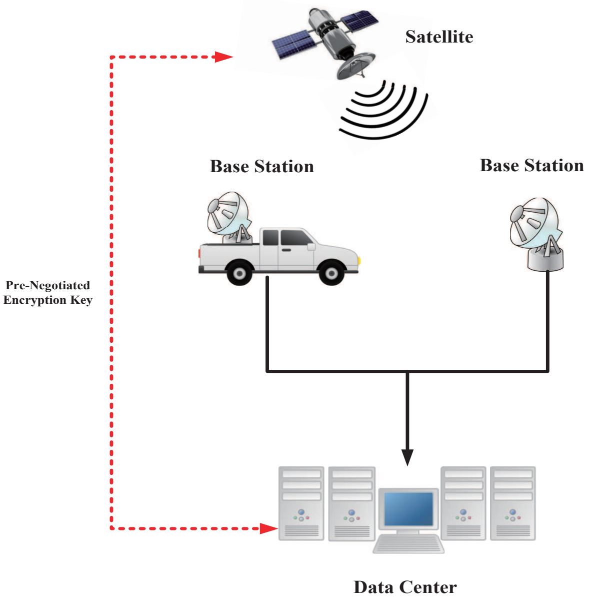


Fig. 1. Image data hiding in the scenario of secure remote sensing.

In this work, we propose an encrypted-domain RIDH scheme by specifically taking the above-mentioned design preferences into consideration. The proposed technique embeds message through a public key modulation mechanism, and performs data extraction by exploiting the statistical distinguishability of encrypted and non-encrypted image blocks. Since the decoding of the message bits and the original image is tied together, our proposed technique belongs to the category of *non-separable* RIDH solutions [16] [2]. Compared with the state-of-the-arts, the proposed approach provides higher embedding capacity, and is able to achieve perfect reconstruction of the original image as well as the embedded message bits. Extensive experimental results on 100 test images validate the superior performance of our scheme.

The rest of this paper is organized as follows. Section II briefly overviews the related work on RIDH over the encrypted domain. Section III presents the proposed data hiding technique in encrypted images. In Section IV and V, we describe the approach for data extraction by exploiting the statistical distinguishability of encrypted and non-encrypted image blocks. Section VI analyzes the security of our embedding strategy, and Section VII gives the experimental results. Finally, we conclude in Section VIII.

II. Related Work

Some recent attempts were made on embedding message bits into the encrypted images. In [17], Puech *et. al.* used a simple substitution method to insert additional bits into AES encrypted images. Local standard deviation (SD) was then exploited at the decoder side to reconstruct the original image. Zhang designed a method to embed additional message bits into stream cipher encrypted images by flipping 3 LSBs of half of the pixels in a block [18]. The data extraction can be performed by utilizing the local smoothness inherent to natural images. This method was later improved by Hong *et. al* through a side match technique [19]. As local smoothness does not always hold for natural images, data extraction errors can be observed in the high-activity regions. Further, Zhang proposed a separable RIDH method such that the protection scopes of data hiding key and encryption key are gracefully separated [16]. In [20], Zhang *et. al.* extended the lossless compression based RIDH approach to the encrypted domain, namely, losslessly compress half of the 4th LSBs of the encrypted image via LDPC code to create space for data hiding. As the source coding with side information at the decoder requires a feedback channel, this scheme would face severe challenges in many practical scenarios, e.g., secure remote sensing, where the feedback channel could be very costly. Ma *et. al* suggested a new embedding method by reserving room before encryption with a traditional reversible image watermarking algorithm [21]. Significant improvements on embedding performance can be achieved by shifting partial embedding operations to the encryption phase. More recently, Qian *et. al.* proposed a RIDH framework that is capable of hiding data into an encrypted JPEG bitstream [22]. Other relevant approaches were reported in [23]–[25].

It should be noted that, for all the existing RIDH schemes including both non-separable as well as separable solutions, an extra data hiding key is introduced to ensure embedding security. Certainly, the data hiding key needs to be shared and managed between the date hider and the recipient. As mentioned earlier, the key management functions, e.g., the key generation, activation, de-activation, suspension, expiration,

[2] Opposite to the non-separable schemes, there is another type called separable RIDH approaches, in which the data extraction and image decryption can be separately carried out.

destruction, archival, and revocation, are difficult to be reliably implemented within such distributed infrastructure [15]. A natural question arising now is whether we can design an encrypted-domain RIDH scheme, which does not require a secret data hiding key, while still ensuring that only the party with the secret encryption key K can disclose the embedded message. This could be very valuable in practice, as the cost and the potential risk of building up the KMS can be significantly reduced. Intuitively, this is achievable because the security offered by the encryption key may be appropriately extended to protect the data embedding. In the following sections, we propose an encrypted-domain secure RIDH scheme without data hiding key. As will be clear shortly, the possibility of eliminating the data hiding key is not unique to our proposed method, but rather applicable for all non-separable RIDH schemes. Here, some design goals are slightly different from those of the existing solutions, due to the elimination of the data hiding key. In [18], [20], [21], the images after direct decryption (i.e., decryption without data extraction) are required to be of high quality. However, such requirement becomes invalid in our framework since we only have one single encryption key, making the decryption and data extraction naturally tie together.

III. Proposed RIDH Scheme over Encrypted Domain

Instead of considering dedicated encryption algorithms tailored to the scenario of encrypted-domain data hiding, we here stick to the conventional stream cipher applied in the standard format. That is, the ciphertext is generated by bitwise XORing the plaintext with the key stream. If not otherwise specified, the widely used stream cipher AES in the CTR mode (AES-CTR) is assumed. The resulting data hiding paradigm over encrypted domain could be more practically useful because of two reasons: 1) stream cipher used in the standard format (e.g., AES-CTR), is still one of the most popular and reliable encryption tools, due to its provable security and high software/hardware implementation efficiency [26]. It may not be easy, or even infeasible, to persuade customers to adopt new encryption algorithms that have not been thoroughly evaluated; 2) large number of data have already been encrypted using stream cipher in a standard way.

When stream cipher is employed, the encrypted image is generated by

$$[[\mathbf{f}]] = Enc(\mathbf{f}, K) = \mathbf{f} \oplus \mathbf{K} \tag{1}$$

where $\mathbf{f}$ and $[[\mathbf{f}]]$ denote the original and the encrypted images, respectively. Here, $\mathbf{K}$ denotes the key stream generated by using the secret encryption key K. In this work, without loss of generality, all the images are assumed to be 8-bit. Throughout the paper, we use $[[\mathbf{x}]]$ to represent the encrypted version of $\mathbf{x}$. Clearly, the original image can be obtained by performing the following decryption function

$$\mathbf{f} = Dec([[\mathbf{f}]], K) = [[\mathbf{f}]] \oplus \mathbf{K} \tag{2}$$

As mentioned earlier, the encrypted image $[[\mathbf{f}]]$ now serves as the cover to accommodate message to be hidden. We first divide $[[\mathbf{f}]]$ into a series of non-overlapping blocks $[[\mathbf{f}]]_i$'s of size $M \times N$, where i is the block index. Each block is designed to carry n bits of message. Letting the number of blocks within the image be B, the embedding capacity of our proposed scheme becomes $n \cdot B$ bits. To enable efficient embedding, we propose to use $S = 2^n$ binary *public* keys $\mathbf{Q}_0, \mathbf{Q}_1, \cdots, \mathbf{Q}_{S-1}$, each of which is of length $L = M \times N \times 8$ bits. All $\mathbf{Q}_j$'s, for $0 \le j \le S-1$, are made publicly accessible, which implies that even the attacker knows them. These public keys are pre-selected prior to the message embedding, according to a criterion of maximizing the minimum Hamming distance among all keys. The algorithm developed by MacDonald [27] can be used to this end. Note that all the public keys are built into the data hider and the recipient when the whole system is set up, and hence, it is not necessary to transmit them during the data embedding stage. Also, for fixed S and L, Hamming showed that an upper bound on the minimum Hamming distance can be given as follows [28]. First determine two integers m_1 and m_2 by

$$\sum_{i=0}^{m_1} \binom{L}{i} \le \frac{2^L}{S} < \sum_{i=0}^{m_1+1} \binom{L}{i} \tag{3}$$

$$\sum_{i=0}^{m_2} \binom{L-1}{i} \le \frac{2^{L-1}}{S} < \sum_{i=0}^{m_2+1} \binom{L-1}{i} \tag{4}$$

where $\binom{L}{i} = \frac{L!}{i!(L-i)!}$. It can be shown that both m_1 and m_2 are unique. Then the minimum Hamming distance among all $\mathbf{Q}_j$'s satisfies

$$d_{\min} \le \max\left\{2m_1 + 1, 2m_2 + 2\right\} \tag{5}$$

The schematic diagram of the proposed message embedding algorithm over encrypted domain is depicted in Fig. 2. In this work, we do not consider the case of embedding multiple watermarks for one single block, meaning that each block is processed once at most. For simplicity, we assume that the number of message bits to be embedded is $n \cdot A$, where $A \le B$ and B is the number of blocks within the image. The steps of performing the message embedding are summarized as follows:

Step 1: Initialize block index $i = 1$.

Step 2: Extract n bits of message to be embedded, denoted by $\mathbf{W}_i$.

Step 3: Find the public key $\mathbf{Q}_{[\mathbf{W}_i]_d}$ associated with $\mathbf{W}_i$, where the index $[\mathbf{W}_i]_d$ is the decimal representation of $\mathbf{W}_i$.

For instance, when $n = 3$ and $\mathbf{W}_i = 010$, the corresponding public key is $\mathbf{Q}_2$.

Step 4: Embed the length-n message bits $\mathbf{W}_i$ into the ith block via

$$[[\mathbf{f}]]_i^w = [[\mathbf{f}]]_i \oplus \mathbf{Q}_{[\mathbf{W}_i]_d} \tag{6}$$

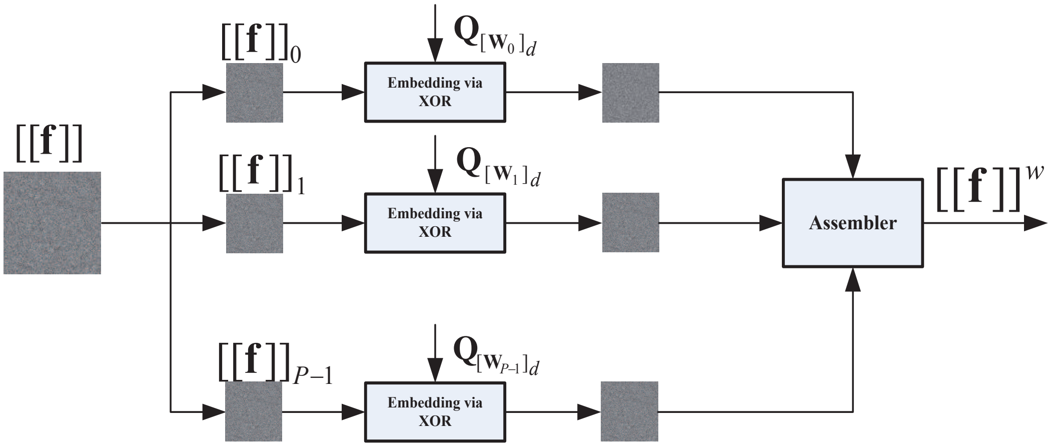


Fig. 2. Schematic diagram of data hiding over encrypted domain.

Step 5: Increment $i = i + 1$ and repeat ***Steps 2-4*** until all the message bits are inserted.

The watermark length parameter A needs to be transmitted alone with the embedded message bits. There are many ways to solve this problem. For instance, we can reserve some blocks to embed A. Or, we can append an end-of-file symbol to the message to be embedded, such that the decoder can implicitly determine A. Both strategies can be readily implemented in practice with negligible affect to the actual embedding rate. For the sake of simpler presentation, we exclude the discussion of embedding A in the sequel.

From the above steps, it can be seen that the message embedding is performed without the aid of a secret data hiding key. As will be proved in the Section VI, high level of embedding security can still be guaranteed, thanks to the protection offered by the encryption key K. In addition, the computations involved in message embedding are rather small (simple XOR operations), and all the block-by-block processing can be readily made parallel, achieving high-throughput.

It is emphasized that the possibility of eliminating the data hiding key is not unique to our proposed method, but rather arguably applicable for all non-separable RIDH schemes over encrypted domain. For instance, the existing non-separable RIDH schemes [18], [19], upon trivial modifications, can still ensure embedding security even if the data hiding key is eliminated. In [18], if we fix the way of partitioning a block into S_0 and S_1 (namely, do not use data hiding key to randomize the block partitioning), then an attacker still cannot compute the fluctuation function (see Eq. (10) of [18]) so as to decode the embedded message. This is because an attacker does not access to the secret encryption key K. In other words, the protection mechanism in the encrypted domain can be naturally extended to provide security for message embedding, eliminating the necessity of introducing an extra data hiding key. This could lead to significant reduction of the computational cost and potential risk of building up a secure KMS, which has been proved to be very challenging in the multi-party environment [15].

Though the possibility of removing the data hiding key holds for all non-separable RIDH schemes over encrypted domain, it has never been pointed out in the existing work. It can be witnessed by the fact that all the existing RIDH schemes, including separable and non-separable ones, involve a data hiding key that has to be shared and managed between the data hider and the recipient. In addition to identifying this property, we, in the following Section VI, will exploit the message indistinguishability to prove that the removal of data hiding key will not hurt the embedding security.

Before presenting the data extraction and image decryption methods, let us first investigate the features that can be used to discriminate encrypted and non-encrypted image blocks. The classifier designed according to these features will be shown to be crucial in the proposed joint data extraction and image decryption approach.

IV. Feature Selection for Discriminating Encrypted and Non-Encrypted Image Blocks

To differentiate encrypted and original, un-encrypted image blocks, we here design a feature vector $\boldsymbol{\rho} = (H, \sigma, \mathbf{V})'$, integrating the characteristics from multiple perspectives. Here, H is a tailored entropy indicator, σ is the SD of the block, $\mathbf{V}$ represents the directional local complexities in four directions. The formation of the above feature elements will be detailed as follows.

Compared with the original, un-encrypted block, the pixels in the encrypted block tend to have a much more uniform distribution. This motivates us to introduce the local entropy into the feature vector to capture such distinctive characteristics. However, we need to be cautious when calculating the entropy values because the number of available samples in a block would be quite limited, resulting in estimation bias, especially when the block size is small. For instance, in the case that $M = N = 8$, we only have 64 pixel samples, while the range of each sample is from 0 to 255. To reduce the negative effect of insufficient number of samples relative to the large range of each sample, we propose to compute the entropy quantity based on quantized samples, where the quantization step size

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2015.2416591, IEEE Transactions on Circuits and Systems for Video Technology

is designed in accordance with the block size. Specifically, we first apply uniform scalar quantization to each pixel of the block

$$\hat{f} = \left\lfloor \frac{MN \cdot f}{256} \right\rfloor \tag{7}$$

where f and $\hat{f}$ denote the original and the quantized pixel values, respectively. Certainly, $\hat{f}$ falls into the range $[0, MN-1]$. The entropy indicator H based on quantized samples is then given by

$$H = -\sum_{j=0}^{MN-1} p(j) \log p(j) \tag{8}$$

where $p(j)$ is the empirical probability of j in the quantized block.

As a single, first-order entropy quantity may not be sufficient to cover all the underlying characteristics of a block, we suggest to augment the feature vector by introducing another element, i.e., the SD defined by

$$\sigma = \sqrt{\frac{1}{MN} \sum_j (\mathbf{f}(j) - \mu)^2} \tag{9}$$

where $\mathbf{f}(j)$ is the jth pixel in the block and $\mu = \frac{1}{MN}\sum_j \mathbf{f}(j)$ is the sample mean over all the samples in the block. By including this feature element, we can improve the classification performance as the data dispersiveness and denseness can be better reflected.

In addition to the above feature components, we also include directional complexity indicators which encode the local geometric information. To this end, we define a 4-tuple vector $\mathbf{V} = (v_1, v_2, v_3, v_4)'$, where

$$\begin{aligned} v_1 &= \sum_j |\mathbf{f}(j) - \mathbf{f}(j_{ne})| \\ v_2 &= \sum_j |\mathbf{f}(j) - \mathbf{f}(j_e)| \\ v_3 &= \sum_j |\mathbf{f}(j) - \mathbf{f}(j_{se})| \\ v_4 &= \sum_j |\mathbf{f}(j) - \mathbf{f}(j_s)| \end{aligned} \tag{10}$$

where $\mathbf{f}(j_{ne})$, $\mathbf{f}(j_e)$, $\mathbf{f}(j_{se})$, and $\mathbf{f}(j_s)$ represent the neighbors in the 45^o (northeast), 0^o (east), -45^o (southeast) and -90^o (south) directions, relative to $\mathbf{f}(j)$, as illustrated in Fig. 3.

Upon the determination of the feature vector $\boldsymbol{\rho}$, we train a two-class SVM classifier with RBF (Gaussian) kernel [29] taking the form

$$Ker(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|} \tag{11}$$

The 0-class and 1-class correspond to the un-encrypted and encrypted image blocks, respectively.

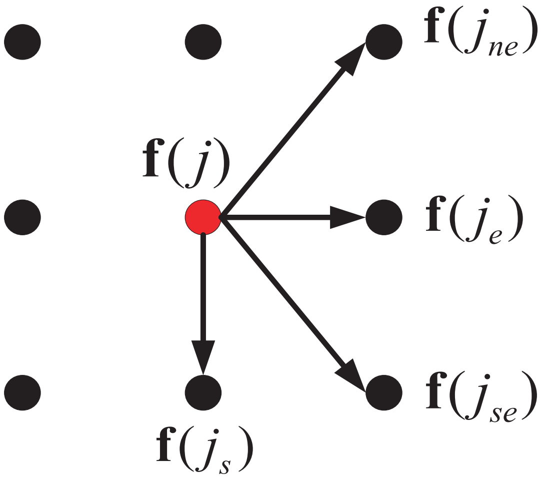


Fig. 3. Illustration of the neighbors of $\mathbf{f}(j)$

Here, the training image set consists of 100 images of size 512×512, with a wide variety of characteristics including natural scenes, artificial images, synthetic images and textual images. The *off-line* trained SVM classifier will be used to discriminate the encrypted and non-encrypted image patches in the process of data extraction and image decryption.

V. Joint Data Extraction and Image Decryption

The decoder in the data center has the decryption key K, and attempts to recover both the embedded message and the original image simultaneously from $[[\mathbf{f}]]^w$, which is assumed to be perfectly received without any distortions. Note that this assumption is made in almost all the existing RIDH methods. Due to the interchangeable property of XOR operations, the decoder first XORs $[[\mathbf{f}]]^w$ with the encryption key stream $\mathbf{K}$ and obtains

$$\mathbf{f}^w = [[\mathbf{f}]]^w \oplus \mathbf{K} \tag{12}$$

The resulting $\mathbf{f}^w$ is then partitioned into a series of non-overlapping blocks $\mathbf{f}_i^w$'s of size $M \times N$, similar to the operation conducted at the embedding stage. From (6), we have

$$\mathbf{f}_i^w = \mathbf{f}_i \oplus \mathbf{Q}_{[\mathbf{W}_i]_d} \tag{13}$$

The joint data extraction and image decryption now becomes a blind signal separation problem as both $\mathbf{W}_i$ and $\mathbf{f}_i$ are unknowns. Our strategy of solving this problem is based on the following observation: $\mathbf{f}_i$, as the original image block, very likely exhibits certain image structure, conveying sematic information. Note that $\mathbf{Q}_{[\mathbf{W}_i]_d}$ must match one of the elements in $\mathcal{Q} = \{\mathbf{Q}_0, \mathbf{Q}_1, \cdots, \mathbf{Q}_{S-1}\}$. Then if we XOR $\mathbf{f}_i^w$ with all $\mathbf{Q}_j$'s, one of the results must be $\mathbf{f}_i$, which would demonstrate structural information. As will become clear shortly, the other results correspond to randomized blocks, which can be distinguished from the original, structured $\mathbf{f}_i$.

More specifically, we first create S decoding candidates by XORing $\mathbf{f}_i^w$ with all the S possible public keys $\mathbf{Q}_0, \mathbf{Q}_1, \cdots, \mathbf{Q}_{S-1}$

$$\begin{aligned} \mathbf{f}_i^{(0)} &= \mathbf{f}_i^w \oplus \mathbf{Q}_0 = \mathbf{f}_i \oplus \mathbf{Q}_{[\mathbf{W}_i]_d} \oplus \mathbf{Q}_0 \\ \mathbf{f}_i^{(1)} &= \mathbf{f}_i^w \oplus \mathbf{Q}_1 = \mathbf{f}_i \oplus \mathbf{Q}_{[\mathbf{W}_i]_d} \oplus \mathbf{Q}_1 \\ &\vdots \\ \mathbf{f}_i^{(S-1)} &= \mathbf{f}_i^w \oplus \mathbf{Q}_{S-1} = \mathbf{f}_i \oplus \mathbf{Q}_{[\mathbf{W}_i]_d} \oplus \mathbf{Q}_{S-1} \end{aligned} \tag{14}$$

As mentioned earlier, one of the above S candidates must be $\mathbf{f}_i$, while the others can be written in the form

$$\mathbf{f}_i^{(t)} = \mathbf{f}_i \oplus \mathbf{Q}_{[\mathbf{W}_i]_d} \oplus \mathbf{Q}_t \tag{15}$$

where $t \neq [\mathbf{W}_i]_d$.

The result $\mathbf{f}_i^{(t)} = Enc(\mathbf{f}_i, \mathbf{Q}_{[\mathbf{W}_i]_d} \oplus \mathbf{Q}_t)$ corresponds to an encrypted version of $\mathbf{f}_i$ with equivalent key stream being $\mathbf{Q}_{[\mathbf{W}_i]_d} \oplus \mathbf{Q}_t$. Notice that all the public keys $\mathbf{Q}_j$'s, for $0 \leq j \leq S-1$, are designed to have maximized minimum-Hamming distance, and the upper bound is given in (5). Hence, $\mathbf{f}_i^{(t)}$ tends to loose the image structural information, making it appear random.

To identify which candidate corresponds to $\mathbf{f}_i$, we apply the designed two-class SVM classifier to these S candidates. Let $\mathbf{r} = (r_0, r_1, \cdots, r_{S-1})'$ be the vector recording the classification results, where $r_j = 0$ and $r_j = 1$ correspond to the original (structured) and randomized blocks, respectively. If there exists a unique j such that $r_j = 0$, then we decode the embedded message bits as

$$\mathbf{W}_i = [j]_2 \tag{16}$$

where $[j]_2$ denotes the length-n binary representation of j and $n = \log_2 S$. For example, if $n = 3$ and $j = 7$, then $[j]_2 = 111$.

Upon determining $\mathbf{W}_i$, the original image block can be easily recovered by

$$\mathbf{f}_i = \mathbf{f}_i^w \oplus \mathbf{Q}_{[\mathbf{W}_i]_d} \tag{17}$$

However, we do observe several cases where there exist multiple j's or no j such that $r_j = 0$. When any of these two cases happens, it indicates that some decoding errors appear. To formally analyze these errors and later suggest an effective error correction mechanism, we define two types of classification error

Type I Error: $\mathbf{f}_i^{(j)} = \mathbf{f}_i$, while $r_j = 1$.

Type II Error: $\mathbf{f}_i^{(j)} \neq \mathbf{f}_i$, while $r_j = 0$.

Type I error mainly occurs when the original block $\mathbf{f}_i$ is very complicated, e.g., from highly textured regions, behaving similarly as an encrypted block. Type II error usually arises when the block size is rather small, making an encrypted block mistakenly be classified as an original, un-encrypted one. As verified experimentally from 200 test images of size 512×512, for a specific block, we assume that at most one type of error will occur. Under this assumption, both Type I and Type II errors can be easily detected. When Type I error occurs, the classification result vector becomes $\mathbf{r} = \mathbf{1}'$. While when Type II error appears, the following inequality holds

$$\sum_j r_j < 2^n - 1 \tag{18}$$

where $n = \log_2 S$. In the rare cases that the above assumption does not hold (both types of errors appear simultaneously), these errors cannot be detected and will still be counted when calculating the extraction accuracy.

When classification errors are detected for some blocks, we need a mechanism to correct them. Though the classifier is carefully designed, it is still difficult to distinguish those highly textured original blocks from the encrypted ones, especially when the block size is small. To solve this challenging problem, we propose to exploit the self-similarity property inherent to natural images. Even for those highly textured images, it is observed that similar blocks could be found in a non-local window [30], as also illustrated in Fig. 4.

According to this phenomenon, the proposed error correction approach is based on the following key observation: if a block is correctly decoded, then with very high probability, there are some similar patches around it. Such property of non-local image similarity motivates us to rank all the potential candidate blocks according to the minimum distance with the patches in a non-local search window. To this end, we first define a to-be-corrected (TBC) set $\mathcal{C}$ by

$$\mathcal{C} = \begin{cases} \left\{ \mathbf{f}_i^{(j)} \middle| 0 \leq j \leq S-1 \right\} & \text{Type I error detected} \\ \left\{ \mathbf{f}_i^{(j)} \middle| r_j = 0 \right\} & \text{Type II error detected} \end{cases} \tag{19}$$

For any candidate block $\mathbf{f}_i^{(j)}$ in $\mathcal{C}$, we calculate its ℓ_2 distances from all the other blocks in a search range $\mathcal{D} \backslash \{\mathbf{f}_i^{(j)}\}$, where $\mathcal{D}$ shares the same center as $\mathbf{f}_i^{(j)}$ and its size is experimentally determined as $5M \times 5N$.

We then can compute the minimum patch distance within the search window

$$d_i^{(j)} = \min_{\mathbf{D} \in \mathcal{D} \backslash \{\mathbf{f}_i^{(j)}\}} \|\mathbf{f}_i^{(j)} - \mathbf{D}\|_F^2 \tag{20}$$

where $\mathbf{D}$ is an arbitrary block of size $M \times N$ within $\mathcal{D} \backslash \{\mathbf{f}_i^{(j)}\}$. Here, we employ the simple MSE criterion when ranking the candidate blocks. By including the texture direction and scale into the above minimization framework, we could further improve the error correcting performance; but we find that the additional gain is rather limited and the incurred complexity is large. The candidate $\mathbf{f}_i^{(j)}$ that gives the smallest $d_i^{(j)}$ is then selected as the decoded block. Upon determining the index j of the employed public key, the embedded message bits and the original image block can be straightforwardly recovered as in (16) and (17). This non-local-based error correction strategy will be shown experimentally to be quite effective in Section VII. The above joint data extraction and image decryption procedures can also be summarized in Fig. 5.

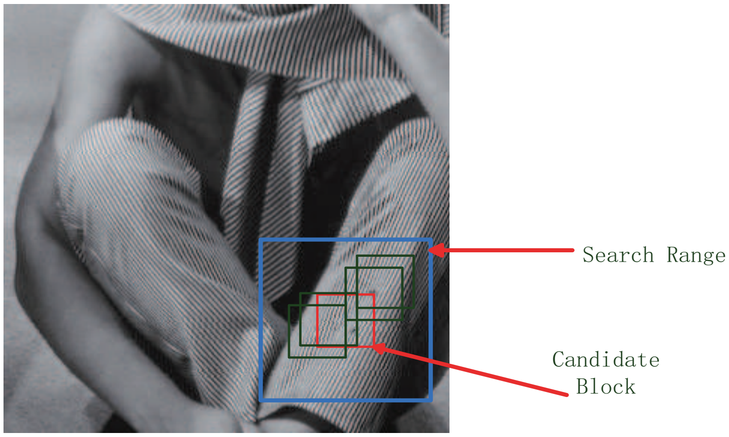


Fig. 4. Illustration of the error correction mechanism based on image self-similarity.

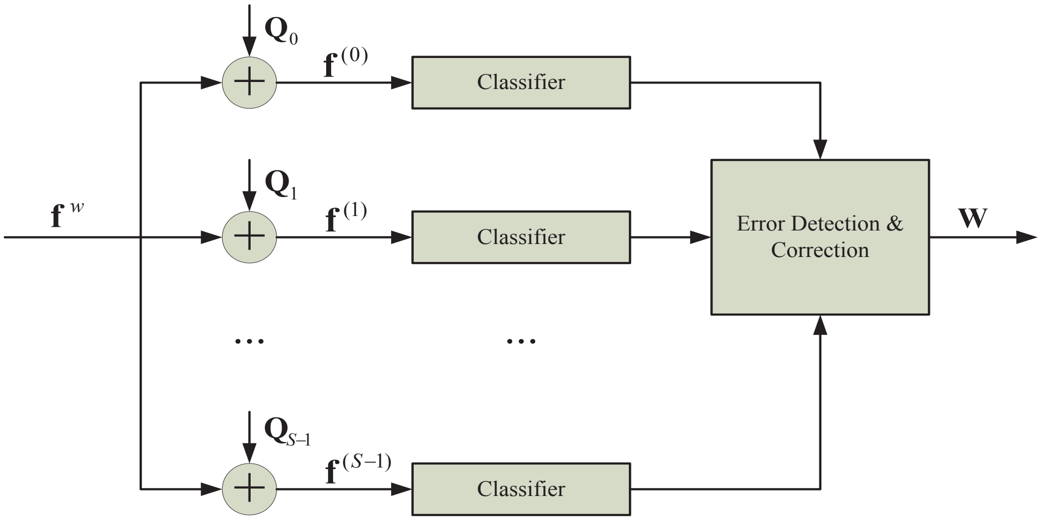


Fig. 5. Schematic diagram of the data extraction.

Remark: Our proposed RIDH scheme over encrypted domain may also be extended to handle compressed and encrypted images, namely, embed watermark into the compressed and encrypted bit stream. Take the JPEG for example. Assume that the encryption is conducted without destroying the structure of JPEG bit stream. For instance, the encryption scheme proposed in [22] can be used to this end. We can XOR the encrypted parts with one of the designed S binary public keys, according to the message bits to be embedded. At the extraction stage, we try all the S possibilities, and identify the one that generates structured image patches in the pixel domain. The embedded message can then be extracted based on the index of the identified public key.

VI. Security Analysis

According to the context of the attack, the attacker may have access to different amount of information. Clearly, the attacker at least can access to watermarked signal, namely, $[[\mathbf{f}]]^w$. In some occasions, the embedded message or the cover signal can also be available to the attacker [31]. Therefore, the security level of the encrypted-domain RIDH scheme should be assessed for different contexts. Similar to the problem of evaluating the security for encryption primitives, [31] defined three types of attacks:

- the Watermarked Only Attack (WOA), in which the attacker only has access to watermarked images;
- the Known Message Attack (KMA), in which the attacker has access to several pairs of *previously* watermarked images and the associated messages. Certainly, the currently transmitted message bits are not known to the attacker;
- the Known Original Attack (KOA), in which the attacker has access to several pairs of *previously* watermarked images and the corresponding cover image. Certainly, the current cover image is not known to the attacker.

As explained in [31], the purposes of the last two attacks are mainly to recover the data hiding key, so as to extract the future embedded messages or hack different pieces of content watermarked with the same key. In our proposed RIDH scheme, the data hiding key has been eliminated, and hence, these two attack models are not applicable.

Under the WOA, the only attack type relevant to our scheme, the attacker attempts to extract the embedded message and/or recover the original image from the watermarked and encrypted image $[[\mathbf{f}]]^w$. Before evaluating the security under WOA, let us first give the definition of message indistinguishability, which should hold for any secure encryption method.

Definition of Message Indistinguishability — concrete version [32]: We say that an encryption scheme (Enc, Dec) is (c, ϵ) message indistinguishable if for every two messages G and G', and for every boolean function T of complexity no larger than c, we have

$$|\mathbb{P}[T(Enc(K,G)) = 1] - \mathbb{P}[T(Enc(K,G')) = 1]| \leq \epsilon \quad (21)$$

where the probability is taken over the randomness of $Enc()$ and the choice of K.

The message indistinguishability implies that the attacker can do no better than simple random guessing if he only observes the ciphertext. This property is regarded as a basic requirement for any secure encryption scheme.

We then have the following Theorem concerning the security of our RIDH algorithm.

Theorem 1: Assuming that the encryption scheme (Enc, Dec) is secure in terms of message indistinguishability, then our RIDH system is secure under WOA attack.

Sketch of the proof: Upon getting the watermarked and encrypted image $[[\mathbf{f}]]^w$, we can still partition it into non-overlapping blocks of size $M \times N$. For each block, we can generate S decoding candidates, in a similar fashion as (14)

$$\begin{aligned}
\mathbf{f}_i^{(0)} &= [[\mathbf{f}]]_i^w \oplus \mathbf{Q}_0 = \mathbf{f}_i^w \oplus \mathbf{Q}_0 \oplus \mathbf{K}_i \\
&= Enc(\mathbf{f}_i^w \oplus \mathbf{Q}_0, \mathbf{K}_i) \\
\mathbf{f}_i^{(1)} &= [[\mathbf{f}]]_i^w \oplus \mathbf{Q}_1 = \mathbf{f}_i^w \oplus \mathbf{Q}_1 \oplus \mathbf{K}_i \\
&= Enc(\mathbf{f}_i^w \oplus \mathbf{Q}_1, \mathbf{K}_i) \\
&\vdots \\
\mathbf{f}_i^{(S-1)} &= [[\mathbf{f}]]_i^w \oplus \mathbf{Q}_{S-1} = \mathbf{f}_i^w \oplus \mathbf{Q}_{S-1} \oplus \mathbf{K}_i \\
&= Enc(\mathbf{f}_i^w \oplus \mathbf{Q}_{S-1}, \mathbf{K}_i) \qquad (22)
\end{aligned}$$

where $\mathbf{K}_i$ denotes the sub-keystream for the ith block.

With any observed $\mathbf{f}_i^{(j)}$, it is computationally infeasible to figure out, with probability significantly larger than $1/S$, which one among $\{\mathbf{f}_i^w \oplus \mathbf{Q}_0, \mathbf{f}_i^w \oplus \mathbf{Q}_1, \cdots, \mathbf{f}_i^w \oplus \mathbf{Q}_{S-1}\}$ is the message encrypted by $\mathbf{K}_i$, due to the property of message indistinguishability described in (21). Therefore, the attacker attempting to extract the embedded message bits from $[[\mathbf{f}]]^w$ should be able to do no better than random guessing. This proves the security of our proposed encrypted-domain RIDH strategy against WOA attack.■

VII. Experimental results

In this section, we experimentally evaluate the embedding performance of our proposed encrypted-domain RIDH scheme. The test set is composed of 100 images of size 512×512 with various characteristics, including natural images, synthetic images, and highly textured images. All the test images can be downloaded from https://dl.dropboxusercontent.com/u/103270026/TestImage.zip. Obviously, the test set is different from the training set used to derive the two-class SVM classifier.

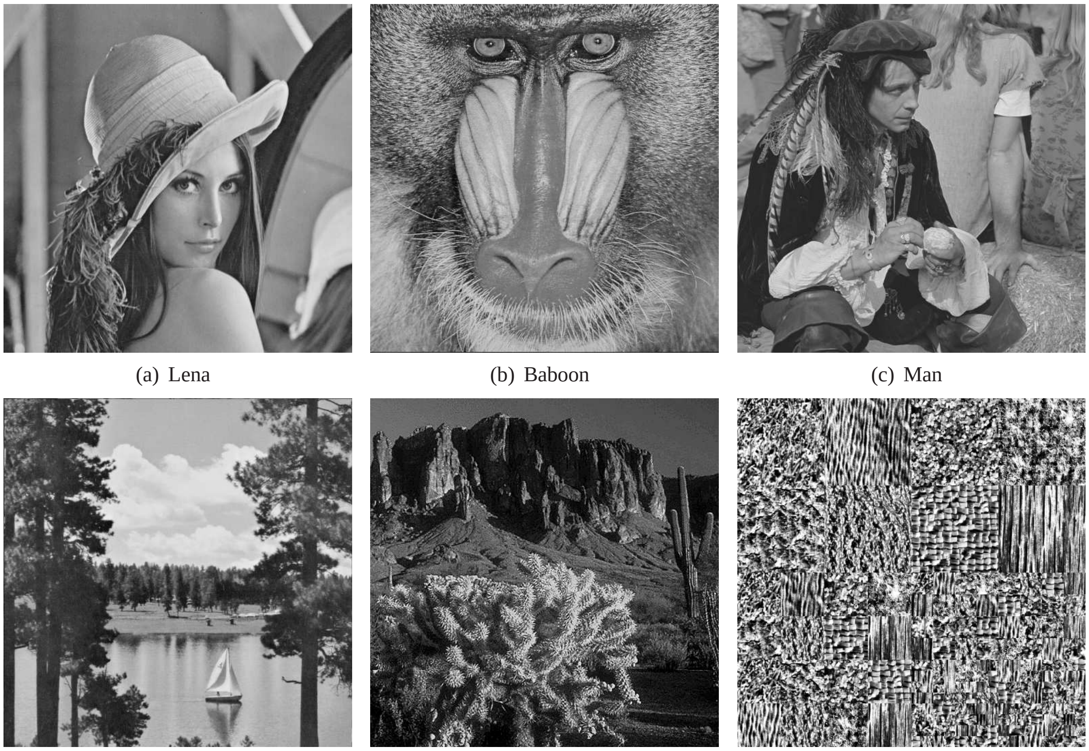
As mentioned in Section III, we stick to standardized encryption method, and all the images are encrypted using the stream cipher AES-CTR [26]. We would like to compare our scheme with three state-of-the-art algorithms [18], [19] and [17], where standardized encryption methods were also used.

In Table I, we tabulate the embedding capacity and data extraction accuracy τ of our method, [18] and [19] for different settings of block size. Here, τ is defined by

$$\tau = \frac{\#\text{ of correctly extracted bits}}{\#\text{ of embedded bits}} \quad (23)$$

and the values given are averaged over all the blocks in the 100 test images. In this table, we fix $n = 3$ in our method, i.e., each block accommodates 3 bits. As the scheme of [18] only works on blocks no less than 3×3, the results for smaller block configurations are marked with '–'. For fair comparison with [18] and [19], we try different numbers of flipped LSBs, instead of fixing to flip 3 LSBs, and only record the best extraction accuracy in Table I. This is equivalent to remove the constraint on direct decryption. It can be seen that, for all the three methods, the embedding capacity increases as the block size drops. Our method can embed 21675 message bits for each 512×512 image when the block size is 6×6, while ensuring 100% accuracy of data extraction. As the block size decreases further, small number of extraction errors appear. Even when the block size shrinks to 2×2, the accuracy is still as high as 99.2356%. In contrast, the values of τ in [18] and its improved version [19] are consistently lower than 100%, even when the block size is as big as 8×8. Also, for the same block size, the extraction accuracy of our method is significantly higher than those of [18] and [19], while the embedding capacity is 3 times higher.

In addition to the comparison of the averaged extraction accuracy, we also show the results of these three methods for six representative images illustrated in Fig. 6. As can be seen from Fig. 7, for images with large portion of textural regions, e.g., `Texture mosaic 1` and `Cactus`, [18] and [19] give much degraded results, especially when the block size is small. For instance, the extraction accuracy is only 72.1252%, for the image `Cactus` when the block size is 4×4. In contrast, our method offers much better extraction accuracy for all settings of block size. In fact, extraction errors are only detected in three images `Texture mosaic 1`, `Cactus`, and `Baboon` in the case that the block size is 4×4, while for all the other cases with bigger block sizes, 100% extraction accuracy is retained.



(a) Lena (b) Baboon (c) Man

(d) Lake (e) Cactus (f) Texture mosaic 1

Fig. 6. Six test images for fine-grained comparison.

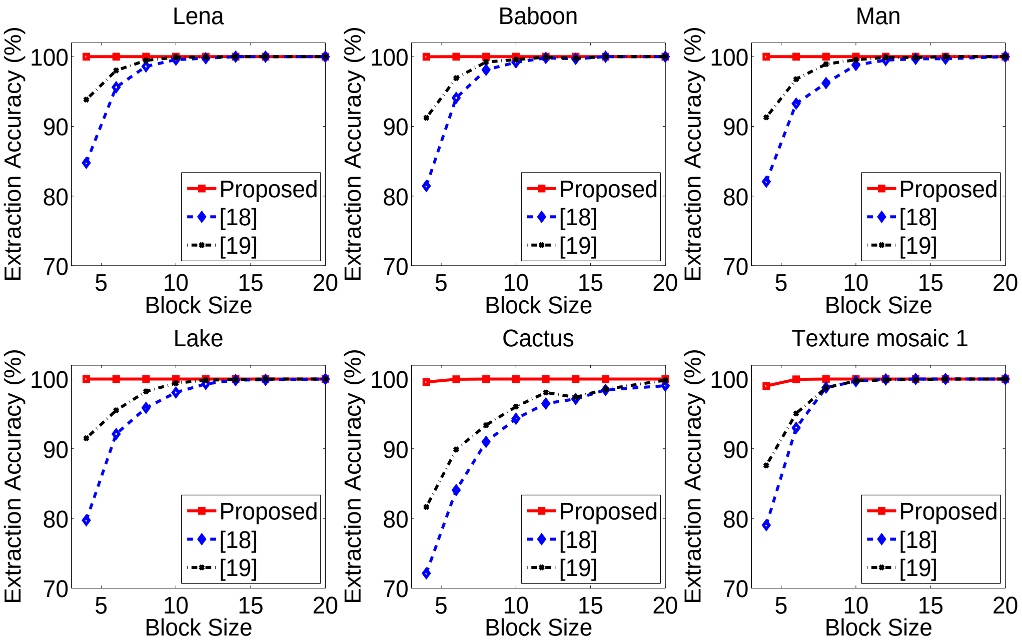


Fig. 7. Comparison of the extraction accuracy for six representative test images.

TABLE I. EMBEDDING PERFORMANCE COMPARISON WITH [18] AND [19].

	proposed		[18]		[19]	
Block Size	Capacity	Accuracy	Capacity	Accuracy	Capacity	Accuracy
8×8	12288 bits	100%	4096 bits	89.4468%	4096 bits	92.0461%
8×7	14016 bits	100%	4672 bits	88.4133%	4672 bits	91.4372%
7×7	15987 bits	100%	5329 bits	87.2088%	5329 bits	90.6550%
7×6	18615 bits	100%	6205 bits	85.7437%	6205 bits	89.6938%
6×6	21675 bits	100%	7225 bits	84.1943%	7225 bits	88.8833%
6×5	26010 bits	99.9973%	8670 bits	82.1644%	8670 bits	87.6347%
5×5	31212 bits	99.9930%	10404 bits	79.9319%	10404 bits	86.1932%
5×4	39168 bits	99.9903%	13056 bits	77.1022%	13056 bits	84.3227%
4×4	49152 bits	99.9761%	16384 bits	73.9654%	16384 bits	82.3897%
3×3	86700 bits	99.8224%	28900 bits	64.0132%	28900 bits	76.8219%
2×2	196608 bits	99.2356%	- -	- -	65536 bits	69.1936%

When comparing with [17], our method also achieves better embedding performance. For a 512×512 image, the embedding capacity of [17] is 16384 bits, as it can only work with 4×4 blocks, and each block accommodates one message bit. As a comparison, our scheme can embed 49152 message bits with the same block size, assuming $n = 3$. Under the above settings, the averaged accuracy of recovering the original image block in our method is 99.9761%, which outperforms the result 97.3062% given by [17]. The performance gap becomes even more significant if we focus on the texture-rich images. For `Texture mosaic 1`, our method leads to the extraction accuracy 99.02%, while the counterpart of [17] is dramatically reduced to 74.83%.

Furthermore, we investigate the effect brought by increasing n, i.e., embed more bits into one single block. Obviously, the number of public keys $\mathbf{Q}_j$'s exponentially increases as we make n larger. This will enlarge the complexity of data extraction as we need to examine all the $S = 2^n$ decoding candidates. Also, the maximized minimum Hamming distance among all the public keys $\mathbf{Q}_j$'s decreases for bigger n, which in turn could result in more extraction errors. Thanks to the powerful error correction mechanism based on image self-similarities, these increased errors can still be corrected to a large extent. As illustrated in Table II, when $n \leq 5$, we still can ensure 100% success rate of data extraction for all 100 test images. As we further increase n from 6 to 10, some extraction errors gradually appear only in two test images `Texture mosaic 1` and `Cactus`, which contain highly textured areas. The data extraction in the remaining 98 images can still be perfectly performed. In Fig. 8, we highlight the blocks in which extraction errors occur in the two problematic images when $n = 8$. It can be observed that the incorrectly decoded blocks are untypically homogenous in textural characteristics to their context, which explains the difficulty in discretion by the proposed error correction mechanism. To tackle this challenge, an error-correcting code (ECC) such as Hamming code can be used to further correct those unsolvable errors, at the cost of significantly reduced embedding rate. Here, we do not discuss the employment of ECC in details because 1) the ECC is a relatively independent component, and 2) the performance of ECC highly depends on the decoding error rate, on which we focus in this work. Upon knowing the characteristics and behavior of the decoding error, the task of designing and implementing an ECC becomes a trivial issue.

Finally, we evaluate the time complexity of performing the joint decryption and data extraction, with respect to different settings of n, where n is the number of bits embedded into one single block. As can be seen from Section V, the computational complexity mainly comes from applying SVM classifier to the $S = 2^n$ decoding candidates. Since the SVM training is conducted off-line, the associated complexity will not be counted into the evaluation of joint decryption and data extraction. In Fig. 9, the results are averaged over all the 100 test images of size 512×512. The measurement of the time complexity is carried out over an un-optimized, unparalleled Matlab implementation by using the built-in tic and toc functions in a personal PC with Intel i7@3.40 GHz CPU and 32 GB RAM. When $n = 1$, namely, each block carries 1 bit message, it takes around 0.66 seconds on average to process one 512×512 sized image. As n becomes larger, the time complexity increases, because there are $S = 2^n$ public keys that need to be examined. Noticing that the joint decryption and data extraction of different blocks are largely independent, except the error correction stage where image self-similarity is exploited, significant time saving can be retained by using a parallel computing platform. We also would like to point out that the complexity of performing the joint decryption and data extraction may not be crucial in many applications, e.g. secure remote sensing, where the recipient has abundant computing resources.

VIII. CONCLUSIONS

In this paper, we design a secure reversible image data hiding (RIDH) scheme operated over the encrypted domain. We suggest a public key modulation mechanism, which allows us to embed the data via simple XOR operations, without the need of accessing the secret encryption key. At the decoder side, we propose to use a powerful two-class SVM classifier to discriminate encrypted and non-encrypted image patches, enabling us to jointly decode the embedded message and the original image signal perfectly. We also have performed extensive experiments to validate the superior embedding performance of our proposed RIDH method over encrypted domain.

REFERENCES

[1] M. U. Celik, G. Sharma, A. Tekalp, and E. Saber,"Lossless generalized-lsb data embedding," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp.253-266, 2005.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2015.2416591, IEEE Transactions on Circuits and Systems for Video Technology

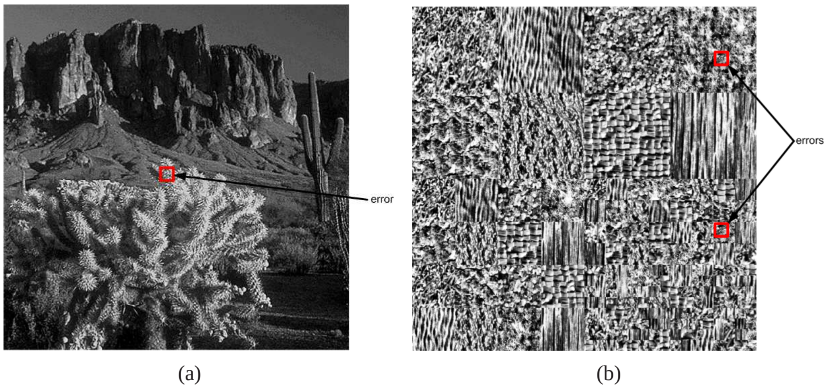


Fig. 8. Erroneous blocks in two problematic images.

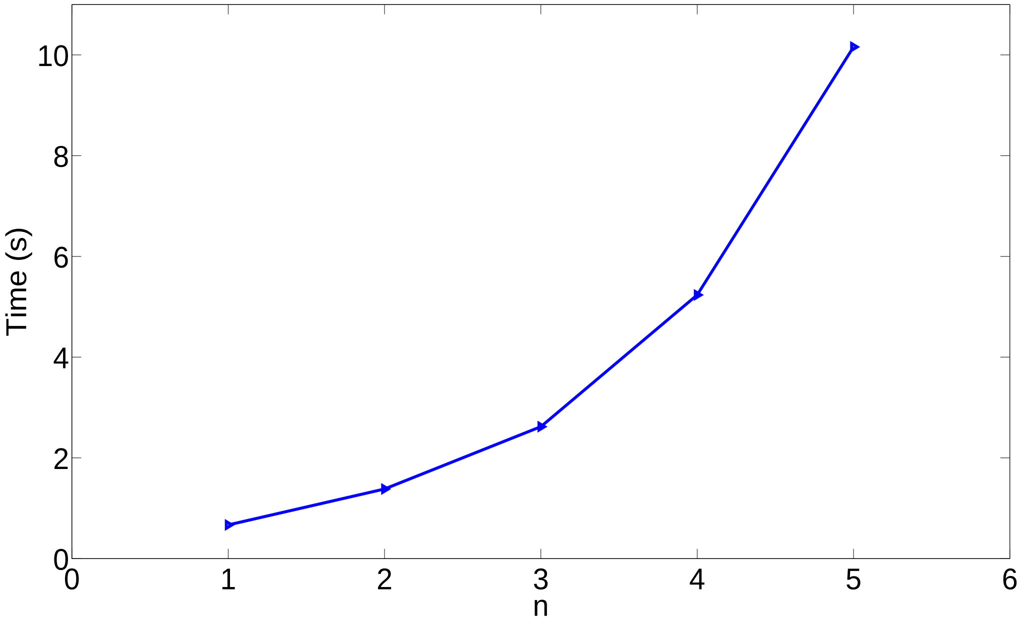


Fig. 9. Time complexity of performing the joint decryption and data extraction over an un-optimized, un-paralleled Matlab implementation.

TABLE II. THE NUMBER OF ERRONEOUS BLOCKS WITH THE INCREASE OF n. HERE, THE BLOCK SIZE IS 8×8.

	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$	$n = 10$
Cactus	0	0	0	0	0	1	1	3
Texture mosaic 1	0	0	0	1	2	2	2	6
Others	0	0	0	0	0	0	0	0
Capacity (bits)	12288	16384	20480	24576	28672	32768	36864	40960
Extraction Error Rate	0	0	0	0.00012%	0.00025%	0.00037%	0.00037%	0.00110%

[2] M. U. Celik, G. Sharma, and A. M. Tekalp,“Lossless watermarking for image authentication: a new framework and an implementation,” *IEEE Trans. Image Process.*, vol. 15, no. 4, pp. 1042-1049, 2006.

[3] Z. Ni, Y. Shi, N. Ansari, and W. Su,“Reversible data hiding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354-362, 2006.

[4] X. Li, W. Zhang, X. Gui, and B. Yang,“A novel reversible data hiding scheme based on two-dimensional difference-histogram modification,” *IEEE Trans. Inf. Forensics Secur*, vol. 8, no. 7, pp. 1091-1100, 2013.

[5] C. Qin, C.-C. Chang, Y.-H. Huang, and L.-T. Liao,“An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 7, pp. 1109-1118, 2013.

[6] W. L. Tai, C. M. Yeh, and C. C. Chang,“Reversible data hiding based on histogram modification of pixel differences,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 6, pp. 906-910, 2009.

[7] J. Tian,“Reversible data embedding using a difference expansion,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890-896, 2003.

[8] Y. Hu, H. K. Lee, and J. Li,“De-based reversible data hiding with improved overflow location map,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, pp. 250-260, 2009.

[9] X. Li, B. Yang, and T. Zeng,“Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection,” *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3524-3533, 2011.

[10] X. Zhang,“Reversible data hiding with optimal value transfer,” *IEEE Trans. Multimedia*, vol. 15, no. 2, pp. 316-325, 2013.

[11] T. Bianchi, A. Piva, and M. Barni,“On the implementation of the discrete fourier transform in the encrypted domain,” *IEEE Trans. Inf. Forensics Secur.*, vol. 4, no. 1, pp. 86-97, 2009.

[12] T. Bianchi, A. Piva, and M. Barni,“Composite signal representation for fast and storage-efficient processing of encrypted signals,” *IEEE Trans. Inf. Forensics Secur.*, vol. 5, no. 1, pp. 180-187, 2010.

[13] M. Barni, F. P., R. Lazzeretti, A.-R. Sadeghi, and T. Schneider,“Privacy-preserving ecg classification with branching programs and neural networks,” *IEEE Trans. Inf. Forensics Secur.*, vol. 6, no. 2, pp. 452-468, 2011.

[14] Z. Erkin, T. Veugen, T. Toft, and R. Lagendijk,“Generating private recommendations efficiently using homomorphic encryption and data packing,” *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 3, pp. 1053-1066, 2012.

[15] M. Chandramouli, R. Iorga and S. Chokhani,“Cryptographic key management issues and challenges in cloud services,” *NIST Report 7956*, pp. 1-31, 2013.

[16] X. Zhang,“Separable reversible data hiding in encrypted image,” *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 2, pp. 826-832, 2012.

[17] W. Puech, M. Chaumont, and O. Strauss,“A reversible data hiding method for encrypted images,” in *Proc. of SPIE 6819*, 2008, pp. 1-9.

[18] X. Zhang,“Reversible data hiding in encrypted image,” *IEEE Signal Processing Lett.*, vol. 18, no. 4, pp. 255-258, 2011.

[19] T. Hong, W. Chen and H. Wu,“An improved reversible data hiding in encrypted images using side match,” *IEEE Signal Processing Lett.*, vol. 19, no. 4, pp. 199-202, 2012.

[20] X. Zhang, Z. Qian, G. Feng, and Y. Ren,“Efficient reversible data hiding in encrypted images,” *J. Vis. Commun. Image R.*, vol. 25, no. 2, pp. 322-328, 2014.

[21] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li,“Reversible data hiding in encrypted images by reserving room before encryption,” *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 3, pp. 553-562, 2013.

[22] Z. Qian, X. Zhang, and S. Wang,“Reversible data hiding in encrypted jpeg bitstream,” *IEEE Trans. Multimedia*, vol. 15, no. 2, pp. 316-325, 2013.

[23] X. Zhang and J. Wang,“Spread spectrum image data hiding in the encrypted discrete cosine transform coefficients,” *J. Electron. Imaging*, vol. 22, no. 4, p. 043029, 2013.

[24] W. Zhang, K. Ma, and N. Yu,“Reversibility improved data hiding in encrypted images,” *Sig. Proc.*, vol. 94, no. 1, pp. 118-127, 2014.

[25] B. Yang, C. Busch, and X. Niu,“Joint reversible data hiding and image encryption,” in *Proc. of SPIE 7541*, 2010, pp. 1-10.

[26] H. Lipmaa, P. Rogaway, and D. Wagner,“Ctr-mode encryption,” *[Online] Available: http://csrc.nist.gov/encryption/modes/workshop1/papers/lipmaa-ctr.pdf.*

[27] J. MacDonald,“Design methods for maximum minimum-distance error-correcting codes,” *IBM J.*, pp. 43-57, 1960.

[28] R. Hamming,“Error detecting and error correcting codes,” *Bell Sys. Tech. J.*, vol. 29, pp. 147-160, 1950.

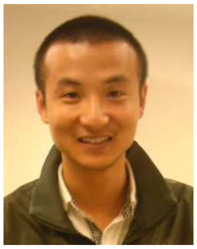
[29] C.-C. Chang and C.-J. Lin,“Libsvm: A library for support vector machines,” *ACM Trans. Intelligent Syst. and Technol.*, vol. 2, no. 3, pp. 27-53, 2011.

[30] A. Buades, B. Coll, and J. Morel,“A non-local algorithm for image denoising,” in *Proc. of CVPR*, 2005, pp. 60-65.

[31] F. Cayre, C. Fontaine, and T. Furon,“Watermarking security: theory and practice,” *IEEE Trans. Sig. Proc.*, vol. 53, no. 10, pp. 3976-3987, 2005.

[32] L. Trevisan, *Cryptography: Lecture Notes from CS276. Stanford University, Online available:* `http://theory.stanford.edu/˜trevisan/books/crypto.pdf`, 2011.

Jiantao Zhou (M’11) is currently an Assistant Professor with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau. He received the B. Eng. Degree from the Department of Electronic Engineering, Dalian University of Technology, Dalian, China, in 2002, the M. Phil degree from the Department of Radio Engineering, Southeast University, Nanjing, China, in 2005, and the Ph.D. degree from the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, in 2009. He held various research positions at the University of Illinois at Urbana-Champaign, the Hong Kong University of Science and Technology, and the McMaster University. His research interests include multimedia security and forensics, and high-fidelity image compression. He was a co-author of a paper that received the Best Paper award in the IEEE Pacific-Rim Conference on Multimedia (PCM) in 2007. He holds 3 granted US patents and 2 granted Chinese patents.



Weiwei Sun (S’14) received the B.S. degree from Shenyang University, Shenyang, China, in 2012. He is currently pursuing the M.S. degree with the Faculty of Science and Technology, University of Macau. His research interests include reversible image data hiding and digital multimedia forensics.

Li Dong (S'14) received the B.S. in software engineering from Chongqing University, Chongqing, China, in 2012, and the M.S. degree also in software engineering from University of Macau, Macau, China, in 2014, where he is currently Ph.D. pursuing degree. His research interests include multimedia security and forensic, signal processing in encrypted domain.

Xianming liu is an Associate Professor with the School of Computer Science and Technology, Harbin Institute of Technology (HIT), Harbin, China. He received the B.S., M.S., and Ph.D degrees in computer science from Harbin Institute of Technology, China, in 2006, 2008 and 2012, respectively. In 2007, he joined the Joint Research and Development Lab (JDL), Chinese Academy of Sciences, Beijing, as a research assistant. From 2009 to 2012, he was with National Engineering Lab for Video Technology, Peking University, Beijing, as a research assistant. In 2011, he spent half a year at the Department of Electrical and Computer Engineering, McMaster University, Canada, as a visiting student. From December 2012 to December 2013, he worked as a post-doctoral fellow at McMaster University, Hamilton, Canada. In 2014, he worked at National Institute of Informatics (NII), Tokyo, Japan, as a project researcher. His research interests include image/video coding, image/video processing.

Oscar C. Au (S'87-M'90-SM'01-F'11) received his B.A.Sc. from Univ. of Toronto in 1986, his M.A. and Ph.D. from Princeton Univ. in 1988 and 1991 respectively. After being a postdoc in Princeton for 1 year, he joined Hong Kong Univ. of Science and Technology (HKUST) as an Assistant Professor in 1992. He is/was a Professor of Dept. of Electronic and Computer Engineering, Director of Multimedia Technology Research Center, and Director of Computer Engineering in HKUST.

His main research contributions are on video/image coding and processing, watermarking/light weight encryption, speech/audio processing. Research topics include fast motion estimation for H.261/3/4/5, MPEG-1/2/4, and AVS, optimal and fast sub-optimal rate control, mode decision, transcoding, denoising, deinterlacing, post-processing, multi-view coding, view interpolation, depth estimation, 3DTV, scalable video coding, distributed video coding, subpixel rendering, JPEG/JPEG2000, HDR imaging, compressive sensing, halftone image data hiding, GPU-processing, software-hardware co-design, etc. He has published 60+ technical journal papers, 350+ conference papers, and 70+ contributions to international standards. His fast motion estimation algorithms were accepted into the ISO/IEC 14496-7 MPEG-4 international video coding standard and the China AVS-M standard. His light-weight encryption and error resilience algorithms are accepted into the China AVS standard. He was Chair of Screen Content Coding AdHoc Group in JCTVC for HEVC. He has 20+ granted US patents and is applying for 70+ more on his signal processing techniques. He has performed forensic investigation and stood as an expert witness in Hong Kong courts many times.

Dr. Au is a Fellow of IEEE, a Fellow of HKIE, and a BoG member of APSIPA. He is/was Associate Editors of 8 journals: TCSVT, TIP, TCAS1, JVCIR, JSPS, TSIP, JMM, and JFI. He is/was Chair of 3 technical committees: IEEE CAS MSA TC, IEEE SPS MMSP TC, and APSIPA IVM TC. He is a member of 6 other TCs: IEEE CAS VSPC TC, DSP TC, IEEE SPS IVMSP TC, IFS TC, and IEEE ComSoc MMC TC. He served on 2 steering committees: IEEE TMM, and IEEE ICME. He also served on organizing committee of many conferences including ISCAS 1997, ICASSP 2003, ISO/IEC 71st MPEG in Jan 2005, ICIP 2010, etc. He was/will be General Chair of several conferences: PCM 2007, ICME 2010, PV 2010, APSIPA ASC 2015, and ICME 2017. He won 5 best paper awards: SiPS 2007, PCM 2007, MMSP 2012, ICIP 2013, and MMSP 2013. He was IEEE Distinguished Lecturer (DL) in 2009 and 2010, APSIPA DL in 2013 and 2014, and has been keynote speaker multiple times.

Yuan Yan Tang (S'88-M'88-SM'96-F'04) is a Chair Professor in Faculty of Science and Technology at University of Macau and Professor/Adjunct Professor/Honorary Professor at several institutes including Chongqing University in China, Concordia University in Canada, and Hong Kong Baptist University in Hong Kong. His current interests include wavelets, pattern recognition, image processing, artificial intelligence. He has published more than 400 academic papers and is the author/coauthor of over 25 monographs/books/bookchapters. He is the Founder and Editor-in-Chief of International Journal on Wavelets, Multiresolution, and Information Processing (IJWMIP), and Associate Editors of several international journals. He is the Founder and Chair of pattern recognition committee in IEEE SMC. He has serviced as general chair, program chair, and committee member for many international conferences. Dr. Tang is the Founder and General Chair of the series International Conferences on Wavelets Analysis and Pattern Recognition (ICWAPRs). He is the Founder and Chair of the Macau Branch of International Associate of Pattern Recognition (IAPR). Dr. Y. Y. Tang is a Fellow of IEEE, and Fellow of IAPR.