IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 27, 2019, accepted December 22, 2019, date of publication December 31, 2019,
date of current version January 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2963170

A New MRF-Based Lossy Compression for Encrypted Binary Images

CHUNTAO WANG[1], **(Member, IEEE), TIANZHENG LI**[1], **JIANGQUN NI**[2], **(Member, IEEE), AND QIONG HUANG**[1]

[1]College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China
[2]School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China

Corresponding author: Chuntao Wang (wangct@scau.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Contract 61672242, Contract 61772573, Contract U1736215, Contract 61872152, and Contract 61702199, in part by the Key Realm Research and Development Program of Guangdong Province under Contract 2019B020214002, Contract 2019B020215002, and Contract 2019B020215004, and in part by the Guangdong Program for Special Support of Top-Notch Young Professionals under Grant 2015TQ01X796.

ABSTRACT Although there exist many researches on the compression of original non-encrypted binary images, few approaches focus on the compression of encrypted binary images. As binary images like contract, signature, halftone images are still used widely in practice, how to compress efficiently encrypted binary images in a lossy way deserves further exploration. To this end, this paper develops a lossy compression scheme for encrypted binary images by exploiting the Markov random field (MRF) model. Considering that the third-party in scenarios of cloud or distributed computing cannot access to the encryption key, we develop the concatenated down-sampling and LDPC-based encoding to perform the compression, in which four different down-sampling methods are designed to facilitate improving the quality of reconstructed image. In reconstruction, we first formulate the lossy reconstruction from the encrypted and compressed binary image as an optimization problem, and then build a joint factor graph involving the LDPC-decoding, decryption, and MRF to solve this optimization problem, in which the MRF is exploited to well infer pixels discarded in the down-sampling process. By adapting the sum-product algorithm (SPA) to the constructed joint factor graph for lossy reconstruction (JFG-LR) and running the adapted SPA on the JFG-LR, we thus recover the original binary image in a lossy way. By integrating the stream-cipher-based encryption, the down-sampling and LDPC-based compression, and the JFG-LR-involved reconstruction, we thus propose a new lossy compression scheme for encrypted binary images. Experimental results show that the proposed scheme achieves desirable compression efficiency, which is comparable to or even better than that of the JBIG2 with the original unencrypted binary image as input.

INDEX TERMS Compression of encrypted binary images, lossy compression, Markov random field, factor graph, LDPC.

I. INTRODUCTION

Nowadays, images are generally taken to convey information. As an image contains a large number of pixels, the sender usually compresses the image, then encrypts the compressed image, and finally transmits it to the receiver through the public communication channel, aiming to save the communication bandwidth and ensure the secrecy. This is the conventional compression-then-compression (CTE) system.

In the situation of cloud computing, distributed processing, etc, however, compression and encryption need to be swapped. This is because the sender (e.g., sensors) may have limited computing resources or lack of interest motivation, and thus it would only encrypt the cover image but not conduct the compression before encryption. As a result, to save the communication bandwidth and storage space, the cloud server would have to compress the encrypted image without accessing to the encryption key. At the receiver side, joint decompression and decryption is performed to recover the original image. This then gives rise to the encryption-then-compression (ETC) case.

Intuitively, as the encryption masks the cover image, it would be impossible to compress the encrypted image. By formulating the ETC system as the problem of distributed

The associate editor coordinating the review of this manuscript and approving it for publication was Eduardo Rosa-Molinar.

source coding with side information at the receiver, however, Johnson *et al.* [1] demonstrated via the information theory that the ETC system neither sacrifices the compression efficiency nor degrades the security in comparison to the conventional CTE system. They also developed in [1] both the lossless and lossy compression schemes to illustrate the practical feasibility of the demonstration. By further exploiting the Markov model to characterize the cover image, Schonberg *et al.* significantly improved the compression efficiency of encrypted binary images and videos [2], [3]. By sufficiently leveraging the statistical correlation among spatial pixels, bitplanes, and color bands before encryption, Lazzeretti and Barni [4] later further enhanced the compression performance of encrypted gray and color images. Later, Kumar and Makur [5] succeeded to better improve the compression efficiency of encrypted gray images by generating prediction errors and conducting the encryption on prediction errors. By using the rate-compatible punctured turbo code for compression and the resolution-progressive way for reconstruction, Liu *et al.* [6] proposed a lossless compression method for encrypted gray images, which can well compress the four most significant bitplanes. Via the clustering on prediction errors and the permutation-based encryption, Zhou *et al.* [7] achieved the compression efficiency close to the conventional state-of-the-art compression approaches taking the original, unencrypted grey images as input. Recently, Wang *et al.* [8] exploited the Markov random field (MRF) to characterize the binary image and constructed the joint factor graph involving the LDPC decoding, decryption, and MRF for binary image reconstruction, which significantly improves the compression efficiency against the state-of-the-art counterpart using the 2-D Markov source model. Later on, they extended it to gray images [9] by deploying the MRF to characterize statistical characteristics for each bitplane and that between successive bitplanes, achieving remarkable improvement in terms of compression efficiency over the method adopting the 2-D Markov source model [2] and obtaining the performance slightly comparable or inferior to the resolution-progressive approach of [6].

Attempting to reach higher compression efficiency at the cost of tolerable distortions, researchers turn to lossy compression methods for encrypted images. According to the techniques of lossy compression, the lossy compression methods for encrypted grey images can be roughly categorized into three categories. The first category exploits the technique of compressive sensing [10]–[12]. The methods in [10], [11], and [12] employ the conventional measurement matrix of compressive sensing, the gradient projection matrix, and the learned dictionary to compress the encrypted signal, respectively, and adopt the modified basis pursuit to reconstruct the original signal.

The second category deploys the technique of scalar quantization for compression [13]–[21]. In [13], Zhang compressed the encrypted grey image by imposing the scalar quantization on transformed coefficients of part of permuted image pixels, which essentially discards the excessively rough and fine transformed coefficients, and recovered the original image via an iterative way. By decomposing the stream-ciphered image into several parts and quantizing each part separately, Zhang *et al.* developed a lossy scalable compression system for encrypted gray images [14]. Later, Zhang *et al.* [15] proposed another lossy compression scheme, which decomposes the permuted grey image into multi-layers and seeks an optimum quantization step via the rate-distortion optimization for quantization of each layer. In [16], the auxiliary information is first generated from the original grey image at the content owner side, and it is then exploited at the cloud side to facilitate the optimization of quantization step and further leveraged at the receiver side to improve the reconstruction quality. In [17], Hu *et al.* exploited the spatial correlation due to the specially designed blockwise mod-256 addition and block permutation to generate prediction errors at the cloud side, deployed the quantization to reduce the encrypted data, and reconstructed the image using the content-adaptive interpolation. Recently, Wang *et al.* first decomposed the gray image with the lifting integer wavelet, and then optimized the quantizer on transformed coefficients by means of the heuristic strategy [18], the rate-distortion theory [19], and the weighted rate-distortion optimization [20]. Very recently, Qin *et al.* [21] arranged the selective encrypted blocks of the cover image into four sets according to block complexity, compressed different blocks with different quantizer, and reconstructed the missing pixels via the total-variation-based inpainting, which well improves the compression efficiency for encrypted grey images.

For the third category, the uniform down-sampling is adopted to compress the encrypted image [22], [23]. In [22], the uniformly down-sampled portion of the stream-ciphered grey image is taken as the base layer while other down-sampled portions are selectively sent in a recursive way to the receiver as the enhancement layer. At the receiver side, the base layer and the available enhancement layers are used to reconstruct the original image in a lossy way, in which the content-adaptive interpolation is employed to recover missing portions of the cover image. In [23], Zhou *et al.* produced the base layer by coding a series of non-overlapping patches of uniformly down-sampling version of the stream-ciphered grey image, select image pixels adaptively according to the off-line learned error model to form the enhancement layer, and designed the iterative and multiscale technique to reconstruct the original image form all available pixel samples.

From the above brief introduction, one can observe that there are a number of schemes [1], [2], [7] focusing on the lossless compression of encrypted binary images, but there is few methods concentrating on the lossy compression of encrypted binary images. Actually, binary images are still used widely in practical scenarios like signature, contract, and halftoning. Therefore, it is vital to investigate the lossly compression for encrypted binary images.

To this end, by taking into account the fact that the MRF well characterizes the spatial statistical correlation and really facilities the lossless compression of encrypted binary images, as demonstrated in our previous work [8], in this paper we propose a new MRF-based lossy compression scheme for the encrypted binary image. In more detail, the content owner encrypts the binary image via the stream cipher, the cloud side down-samples the encrypted binary image via a certain manner followed by generating the LDPC syndrome for the down-sampled sequence, and the receiver recovers the original image by constructing a joint factor graph involving the LDPC decoding, decryption, and MRF and executing the sum-product algorithm (SPA) on the constructed joint factor graph. Extensive simulations show that the proposed scheme achieves high compression efficiency for encrypted binary images and is comparable to or even better than the JBIG2 that works in a lossy way and takes the original unencrypted binary image as input. This thus demonstrates the feasibility and effectiveness of the proposed scheme.

Contributions of this paper are three-fold: 1) Develop a down-sampling method that both achieves any practical compression rate and obtains desirable trade-off between uniformness and randomness of the down-sampled pixels; 2) Formulate the lossy reconstruction as an optimization problem, and solve it by constructing a joint factor graph that involves the LDPC decoding, decryption, and MRF and deriving the SPA that adapts to the constructed joint factor graph; and 3) Propose a new lossy compression scheme for the encrypted binary image, obtaining the compression efficiency comparable to or even better than the JBIG2.

The remainder of this paper is organized as follows. Section 2 briefly introduces the preliminary knowledge about the factor graph and the MRF. The proposed scheme for the lossy compression of encrypted binary images is presented in Section 3, and experimental results are given in Section 4. Section 5 finally draws the conclusion.

II. PRELIMIARY KNOWLEDGE

A. FACTOR GRAPH AND SUM-PRODUCT ALGORITHM

Suppose that a global function, $f\left(x_1, x_2, \ldots, x_n\right)$, can be factorized as a product of local functions $f_j(X_j)$, i.e.,

$$f\left(x_1,\ x_1,\ \ldots,\ x_n\right) = \prod_{j\in J} f_j(X_j), \tag{1}$$

where $x_i(i = 1, 2, \ldots, n)$ are independent variables, and $X_j(j = 1, 2, \ldots, J)$ denotes a proper subset of the variable set, $\{x_1,\ x_1,\ \ldots,\ x_n\}$. Assume that $f^i(x_i)$ is a marginal function for variable x_i, and then it is calculated as:

$$f^i\left(x_i\right) = \sum_{\sim\{x_i\}} f\left(x_1, x_2, \ldots, x_n\right), \tag{2}$$

where $\sim\{x_i\}$ denotes the set containing all variables excluding the x_i.

According to [24], (1) and (2) can be well represented and efficiently computed with the factor graph, respectively. Specifically, by denoting the x_i and $f_j(X_j)$ with a blank circle and a black square, respectively, and connecting the circle and square in case of $x_i \in X_j$, a factor graph can thus be built, where the circle and square are termed the *variable node* (*VN*) and the *factor node (FN)*, respectively. For instance, consider the case of $g(x_1,\ x_2,\ x_3,\ x_4,\ x_5) = f_A(x_1)f_B(x_2)f_C(x_1, x_2, x_3)f_D(x_3, x_4)f_E(x_3, x_5)$, where $J = \{A,\ B,\ C,\ D,\ E\}, X_A = \{x_1\}, X_B = \{x_2\}, X_C = \{x_1, x_2, x_3\}$, $X_D = \{x_3, x_4\}$, and $X_E = \{x_3, x_5\}$ [24]. Then the factor graph for this case can be constructed as shown in Fig. 1.

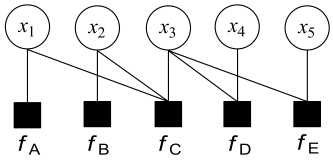


FIGURE 1. The factor graph for $g(x_1,\ x_2,\ x_3,\ x_4,\ x_5)$, where circles and squares denote x_i ($i = 1,\ \ldots,\ 5$) and $f_j(X_j)$, respectively.

By running the SPA (sum-product algorithm) on the constructed factor graph, the marginal function $f^i(x_i)$ can be efficiently computed [24]. To illustrate this, consider the factor graph shown in Fig. 2, which can actually be obtained from a certain factor graph via a proper arrangement and thus would not lose the generality. Let $v_{x\to f}(x)$ and $\mu_{f\to x}(x)$ be the message from VN x to FN f and that from f to x, respectively. Then the $v_{x\to f}(x)$ is calculated via the PRODUCT operation, i.e.,

$$v_{x\to f}\left(x\right) = \prod_{h\in N(x)\backslash f} \mu_{h\to x}\left(x\right), \tag{3}$$

where $N\left(x\right)\backslash f$ denotes the neighborhood of x (i.e., all FNs connecting to x) excluding f. The $\mu_{f\to x}(x)$ can be computed via the SUM operation, i.e.

$$\mu_{f\to x}\left(x\right) = \sum_{\sim\{x\}} \left(f\left(X\right) \prod_{y\in N(f)\backslash x} v_{y\to f}\left(x\right) \right). \tag{4}$$

After completing product and sum operations, summing up all messages from the neighborhood of x results in the marginal function for x, says $f(x)$, i.e.,

$$f(x) = \prod_{h\in N(x)} \mu_{h\to x}\left(x\right). \tag{5}$$

where $N(x)$ stands for the neighborhood of x.

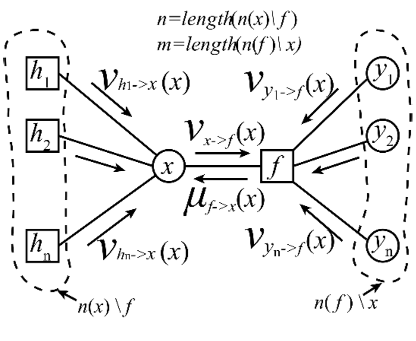


FIGURE 2. Illustration of the SPA.

5	4	3	4	5
4	2	1	2	4
3	1	(x,y)	1	3
4	2	1	2	4
5	4	3	4	5

FIGURE 3. Illustration of neighborhood systems with different orders.

For a factor graph with cycles, the SPA can be iteratively conducted until a certain convergent condition is reached. After convergence, all marginal functions can be obtained in a way similar to (5).

B. MARKOV RANDOM FIELD

The Markov random field is a kind of statistical model that well characterizes the spatial statistics of natural images and thus has been widely used in many research fields like image denoising, segmentation, computer stereo vision [25], [26]. It is briefly introduced as follows.

Suppose that $I(x, y)$ denotes a Q-bit image of size $W \times B$ and $L = \{(x, y)|x \in [1, W], \ y \in [1, B]\}$ stands for its coordinate set, where Q is the bit depth. If each pixel is represented with a random variable, $F_s(s \in L)$, that takes on values in the state space, $\Phi = \{0, 1, \ldots, 2^Q - 1\}$, then all F_s s form a random filed, $\mathrm{F} = \{F_s|F_s \in \Phi, s \in \mathrm{L}\}$. Clearly, each image of size $W \times B$, says $F = (F_1 = f_1, \ F_2 = f_2, \ \ldots, \ F_{WH} = f_{WH})$ becomes an instance of the F.

If the F satisfies the following characteristics of positivity and Markovian, i.e.,

$$p\left(\mathrm{F} = F\right) \geq 0, \quad \forall F \in \mathrm{F} \tag{6}$$

$$p\left(F_s|F_{L-s}\right) = p\left(F_s|F_{\delta(s)}\right) \tag{7}$$

then the F turns to be a Markov random field. In (6) and (7), the $p(\mathrm{F} = F)$, $L-s$, and $\delta(s)$ denote the probability of instance F, the coordinate set excluding s, and the neighborhood of s, respectively. The two equations imply that the probability $p(F_s)$ for any pixel is non-negative and only depends on its neighbors.

The $\delta(s)$ is also called the neighborhood system defined on L. It is defined as:

$$\delta\left(s\right) = \left\{s'|\ \left\|ss'\right\| \leq d, s \neq s', \left\{s, s'\right\} \subseteq L\right\} \tag{8}$$

where $\|\cdot\|$ denotes the distance between s and s', and d is a positive number. Fig. 3 illustrates 5 neighborhood systems. The number in Fig. 3 represents the distance with respect to the center location $s = (x, y)$, and elements with numbers equal to or less than $k(k \geq 1)$ form a k-th neighborhood system.

According to the Hammersley-Clifford theorem [27], [28], the MRF is equivalent to the Gibbs random field. Thus, the MRF can be equivalently calculated as:

$$p\left(\mathrm{F} = F\right) = \frac{1}{Z} \exp\left(-\frac{U\left(F\right)}{T}\right), \tag{9}$$

where Z and T are normalizing and temperature constants, respectively, and $U(F)$ is an energy function

$$U\left(F\right) = \sum_{c \in C} V_c\left(F\right), \tag{10}$$

where C and $V_c(\cdot)$ denote a set of cliques formed by the neighborhood system $\delta(s)$ and a potential function defined on a given clique c $(c \in C)$, respectively. Clique's structure depends on the order of $\delta(s)$, which is omitted here for compactness and recommended to refer to [26], [27], [7].

In case of one clique, i.e., $(f_s, \ f_c)$, $f_s, f_c \in F, \ s, \ c \in L$, the probability of f_s conditioned on f_c is calculated as:

$$p(f_s|f_c) = \frac{1}{Z}\exp\left(-\frac{V_c(f_s|f_c)}{T}\right). \tag{11}$$

III. JOINT FACTOR GRAPH FOR LOSSY RECONSTRUCTION AND THE ADAPTED SPA

In our scheme, the content owner encrypts the binary image with stream cipher, the cloud side performs the lossy compression by down-sampling the encrypted image and generating the LDPC (low-density parity check) syndrome for the down-sampled encrypted pixels, and the receiver recovers the down-sampled portion through LDPC decoding and decryption followed by reconstructing the missing pixels via the MRF.

As the encryption by the content owner and the compression at the cloud side are relatively simple, in this section we mainly focus on the lossy reconstruction of the original binary image. To achieve desirable reconstruction performance, we first formulate the reconstruction problem as an optimization one, and then employ the factor graph to solve it. Details for the design of joint factor graph for lossy reconstruction (JFG-LR) and the derivation of the SPA adapted to the JFG-LR are presented below.

A. PROBLEM FORMULATION

Assume that $I^r(x, y)$ and $I(x, y)$ are the reconstructed and original binary images, respectively. Then the objective of lossy reconstruction is to make $I^r(x, y)$ be sufficiently close to $I(x, y)$. This is equivalent to maximize the peak signal-to-noise ratio (PSNR) between $I^r(x, y)$ and $I(x, y)$ and keep the MRF of $I^r(x, y)$ to be nearly identical to that of $I(x, y)$.

By deploying the Kullback-Leibler divergence, $D_{KL}(p(F^r)||p(F))$, to measure the MRF difference, where F^r and F are MRFs for $I^r(x, y)$ and $I(x, y)$, respectively, we formulate the lossy reconstruction problem as follows, i.e.,

$$\begin{aligned} &\max_{F^r} \ PSNR\left(I^r(x, y), I(x, y)\right) - \lambda D_{KL}\left(p(F^r)||p(F)\right) \\ &subject\ to\ I^r(L_{dwn}) = I\ (L_{dwn}), \end{aligned} \tag{12}$$

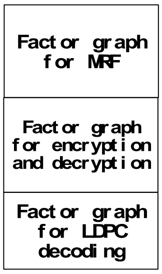
where $PSNR(I^r(x, y), I(x, y))$ denotes the function computing the PSNR between $I^r(x, y)$ and $I(x, y)$, λ is a positive multiplier, and L_{dwn} contains coordinates for the down-sampled pixels of the encrypted image.

As the optimization problem in (12) is troublesome to solve in a qualitative way, we turn to the factor graph to address it by taking into account the fact that the factor graph can well represent the MRF and facilitate the efficient computation of the marginal function. In more detail, we represent the MRF of $I^r(x, y)$ with a factor graph and fix its VNs (variable nodes) at coordinates L_{dwn} to be $I\ (L_{dwn})$, which satisfies the constraint in (12). We then use a potential function to characterize the statistical relationship between the connected VN and FN (factor node), which would probably keep the MRF of F^r to be nearly identical to that of F when the same potential function is adopted for both F^r and F. We finally run the SPA on the constructed factor graph iteratively to recover the missing pixels of $I(x, y)$, which sufficiently exploits the spatial statistics represented with the MRF and thus would result in a PSNR as high as one could. In this way, the optimization problem in (12) can be well solved.

Actually, employing the factor graph to solve (12) leads to another advantage. That is, the factor graph for lossy reconstruction can be seamlessly integrated with those for LPDC decoding and decryption since the LPDC decoding and decryption can be well represented with the factor graph, in which the statistics may also facilitate the LDPC decoding and consequently improve the compression efficiency. This gives rise to the JFG-LR, which is designed in the next subsection.

B. CONSTRUCTION OF JFG-LR

As aforementioned, the receiver conducts the successive operations of LDPC decoding, decryption, and MRF-based recovery of missing pixels to reconstruct the original image in a lossy way. Therefore, by applying the theory of factor graph [24] to construct separate factor graphs for LPDC decoding, decryption, and MRF followed by cascading them together, we can build the JFG-LR accordingly, as illustrated in Fig. 4. Specific design for three separate factor graphs is given below.


Fact or graph for MRF

Fact or graph for encrypti on and decrypti on

Fact or graph for LDPC decodi ng

FIGURE 4. **Illustration of the JFG-LR.**

1) FACTOR GRAPH FOR LDPC DECODING

According to [1], channel codes like the LDPC can be used to compress encrypted signals. Specifically, let H be a parity-check matrix of the LDPC with size $M \times N$ and $Y = \{Y_i | i = 1, \ldots, N\}$ be the encrypted one-dimensional (D) sequence. Then the Y can be compressed by generating the LDPC syndrome, i.e., $S = H \cdot Y$.

Through the LDPC decoding, the compressed signal S can be de-compressed, which is denoted as $\hat{Y}$. According to [28], the LDPC decoding can be conducted via the Tanner graph, which denotes message bits and the parity-check constraint with VNs and check nodes (CNs), respectively, and connects VN i $(i = 1, \ldots, N)$ and CN j $(j = 1, \ldots, M)$ if H_{ij} is equal to 1. Considering that the MRF is represented with the factor graph, we modify the Tanner graph for LDPC decoding to be a factor graph, aiming to integrate the LDPC decoding, decryption, and MRF-based lossy reconstruction seamlessly. In more detail, by taking into account the fact that the conventional LDPC constrains the parity check to be zero while the LDPC decoding in our case requires the parity check to be S_j $(j = 1, \ldots, M)$, we characterize each parity check of the LDPC with an FN (factor node) and represent message and syndrome bits with VNs. As a result, we construct the factor graph for the LPDC decoding in our scheme, as shown in Fig. 5.

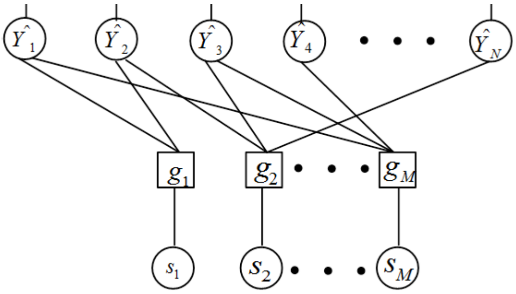


FIGURE 5. **The factor graph for LDPC decoding in our scheme. Th $S_j (j = 1, \ldots, M)$, g_j, and $\hat{Y}_i (i = 1, \ldots, N)$ denote syndrome bits, parity-check constraint, and encrypted image pixels.**

2) FACTOR GRAPH FOR DECRYPTION

In our scheme, we encrypt the cover image $I(x, y)$ via the stream cipher, i.e., $Y = I \oplus K$, where $\oplus$ and K stand for the bit-wise XOR and a pseudo-random sequence generated via a secret key, respectively. Therefore, the decryption can be formulated as:

$$t\left(\hat{F}_i,\ K_i,\ \hat{Y}_i\right) = \left[\hat{F}_i \oplus K_i \oplus \hat{Y}_i = 0\right] \tag{13}$$

where $\hat{F}_i$ $(i = 1, \ldots, N)$ and $[P]$ denote the decrypted image pixel and the "Iverson's convention" [29] that takes on value 1 if P is true and 0 otherwise. By representing the $\hat{F}_i$, K_i, and $\hat{Y}_i$ with VNs and the function $t(\hat{F}_i,\ K_i,\ \hat{Y}_i)$ with an FN, the factor graph for decryption can be built accordingly, as illustrated in Fig. 6.

3) FACTOR GRAPH FOR MRF-BASED LOSSY RECONSTRUCTION

Assume that $\hat{I}(x, y)$ is the reconstructed image of size $W \times B$ and $\hat{F}_{y,x}$ denotes its MRF. According to (9)-(10), joint probability for the $\hat{F}_{y,x}$ can be factorized into a number of

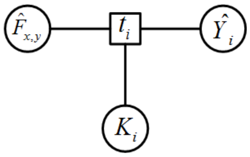


FIGURE 6. The factor graph for decryption.

local probability functions, and thus the factor graph can be employed to represent the MRF.

In constructing the factor graph for the MRF $\hat{F}_{y,x}$, a VN is used to represent an image pixel, while an FN is to characterize the potential constraint between neighboring pixels. According to Section 2.2, the potential constraint depends on the adopted potential function. For simplification and effectiveness, in this paper we deploy the two-order potential function. It uses the one-order neighborhood system (see also Fig. 2) and contains five cliques, i.e., $(\hat{I}(x,y))$, $(\hat{I}(x,y),\ \hat{I}(x-1,y))$, $(\hat{I}(x,y),\ \hat{I}(x+1,y))$, $(\hat{I}(x,y),\ \hat{I}(x,y-1))$, and $(\hat{I}(x,y),\ \hat{I}(x,y+1))$. As the clique has no any potential with respect to itself, the other four cliques are considered. Therefore, the factor graph for $\hat{F}_{y,x}$ can be constructed as shown in Fig. 7.

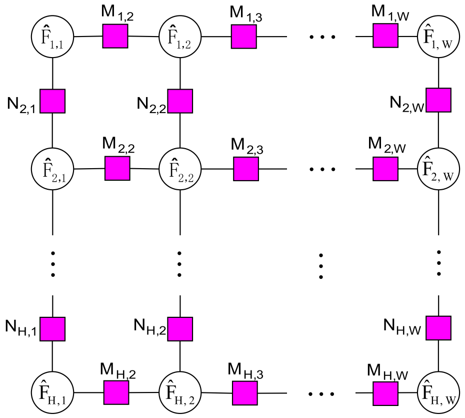


FIGURE 7. Factor graph for the MRF. The $\hat{F}_{y,x}$ ($y \in [1,H]$, $x \in [1,W]$) an $\mathrm{M}_{y,x}/\mathrm{N}_{y,x}$ denotes the random variable for pixel $\hat{I}(x,y)$ and the potential constraint in the horizontal/vertical direction, respectively.

By taking the compression in our scheme into account, the $\hat{I}(x,y)$ actually contains two parts. One is recovered from the compressed and encrypted sequence sent from the cloud side, and the other corresponds to pixels discarded in the down-sampling process. In other words, the portion of $\hat{F}_{y,x}$ corresponding to the first part essentially come from the factor graph for decryption and thus would connect to FNs in the factor graph for decryption, while the others would not. Therefore, to seamlessly integrate the factor graph for the MRF with that for decryption, we modify the factor graph for the MRF by inserting FNs t_i $(i = 1, \ldots, N)$ for $\hat{F}_{y,x}$ s corresponding to the first part and remain the other $\hat{F}_{y,x}$ s unchanged. In addition, to better exploit the priori information of probability, we also introduce another kind of FN, namely $P_{y,x}$, that is connect to each VN $\hat{F}_{y,x}$. These two modifications yield the factor graph for the MRF in our scheme, as illustrated in Fig. 8.

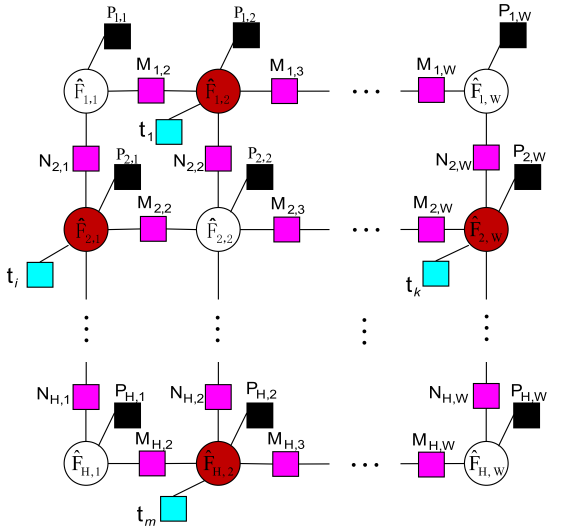


FIGURE 8. Illustration of the factor graph for the MRF in our scheme. Compared with Fig. 6, FNs t_i and $P_{y,x}$ are additionally inserted to represent the connection between VN $\hat{F}_{y,x}$ in the factor graph for the MRF and FN t_i in the factor graph for decryption and the priori information of probability, respectively. Filled circles stand for the down-sampled pixels while blank ones denote the missing pixels discarded in the down-sample process, where the down-sampled coordinates are selected in a certain way (e.g., uniform or random).

C. DERIVATION OF THE SPA ADAPTED TO JFG-LR

By merging the same VNs and FNs of the separate factor graphs in Figs. 5, 6, and 8, we can thus integrate these factor graphs together to yield the JFG-LR (see also Fig. 4). By running the SPA on the JFG-LR, we then reconstruct the original binary image in a lossy way. To this end, we adapt the SPA to the constructed JFG-LR in this subsection.

Fig. 9 plots the flowchart for the SPA. It includes steps of initialization, message update from FNs to VNs, message update from VNs to FNs, optimal estimation of $\hat{Y}$, convergence check, and optimum estimation of $\hat{I}(x,y)$. Regarding that the binary image is used as the cover, messages passed between VNs and FNs are defined as the form of $\ln(p(0)/p(1))$, where $p(0)$ and $p(1)$ stand for the probabilities for bits 0 and 1, respectively. For notational convenience, the message from VN to FN is denoted as $v_{VN\rightarrow FN}$ and that from FN to VN is $\mu_{FN\rightarrow VN}$. By following this notation symbols, details for these steps are presented as follows.

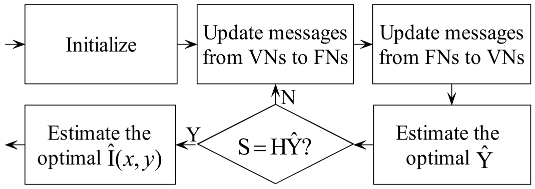

Initialize
Update messages from VNs to FNs
Update messages from FNs to VNs
Estimate the optimal $\hat{Y}$
$S=H\hat{Y}$?
N
Y
Estimate the optimal $\hat{I}(x,y)$

FIGURE 9. Flowchart of the SPA.

1) *Initialization*. This step initializes messages from all VNs to the connected FNs. First consider $\nu_{S_j \to g_j}(j = 1, \ldots, M)$. As syndrome bit S_j is sent from the cloud side and thus deterministic, the $\nu_{S_j \to g_j}$ is calculated:

$$\nu_{S_j \to g_j} = \log\left(\frac{p\left(S_j = 0\right)}{p\left(S_j = 1\right)}\right) = \begin{cases} +\infty & \text{if } S_j = 0 \\ -\infty & \text{otherwise} \end{cases} \tag{14}$$

As pointed out by Johnson *et al.* [1], the encrypted sequence $K = \{K_i | i = 1, \ldots, N\}$ can be taken as the side information for $\hat{Y}$. Thus, K can be used to initialize $\hat{Y}$:

$$\nu_{\hat{Y}_i \to g_j} = \nu_{\hat{Y}_i \to t_i} = \log\left(\frac{p\left(\hat{Y}_i = 0\right)}{p\left(\hat{Y}_i = 1\right)}\right) = \begin{cases} +\infty & \text{if } K_i = 0 \\ -\infty & \text{otherwise} \end{cases} \tag{15}$$

Via the priori probability $P_{y,x}$, messages $\nu_{\hat{F}_{y,x} \to t_i}$, $\nu_{\hat{F}_{y,x} \to M_{y,x}}$, $\nu_{\hat{F}_{y,x} \to M_{y,x+1}}$, $\nu_{\hat{F}_{y,x} \to N_{y,x}}$, and $\nu_{\hat{F}_{y,x} \to N_{y+1,x}}$ are initialized to be $\ln(p(P_{y,x} = 0)/p(P_{y,x} = 1))$. By the way, as the $P_{y,x}$ is a constant, it is not necessary to pass messages from $\hat{F}_{y,x}$ to $P_{y,x}$, and so are the situations for K_i and S_j.

In case of doping (i.e., the encrypted image bit is directly sent to the receiver without any LDPC-based compression, as will be described in the next section), $\hat{Y}_i$ can be determined as the corresponding syndrome bit S_j. As a result, $\nu_{\hat{Y}_i \to g_j}$ and $\nu_{\hat{Y}_i \to t_i}$ are computed with (14). Since $\hat{Y}_i$ has been determined, the $\hat{F}_{y,x}$ connected to t_i can be decided as $\hat{Y}_i \oplus K_i$. Thus, messages from these $\hat{F}_{y,x}$ to their connected FNs can also be initialized via (14).

2) Update $\mu_{FN \to VN}$. This step updates messages from FNs to VNs. According to the theory of LDPC decoding [28], [1], message $\mu_{g_j \to \hat{Y}_i}$ is computed as:

$$\mu_{g_j \to \hat{\mathrm{Y}}_i} = (-1)^{S_j} \log \frac{1 + \prod_{\hat{\mathrm{Y}}_o \in N(g_j) \backslash \hat{\mathrm{Y}}_i} \tanh\left(\frac{\nu_{\hat{\mathrm{Y}}_o \to g_j}}{2}\right)}{1 - \prod_{\hat{\mathrm{Y}}_o \in N(g_j) \backslash \hat{\mathrm{Y}}_i} \tanh\left(\frac{\nu_{\hat{\mathrm{Y}}_o \to g_j}}{2}\right)} \tag{16}$$

In the factor graph for decryption, as $\hat{F}_{y,x} = \hat{Y}_i \oplus K_i$ holds, messages $\mu_{t_i \to \hat{Y}_i}$ and $\mu_{t_i \to \hat{F}_{y,x}}$ are updated as:

$$\mu_{t_i \to \hat{Y}_i} = (-1)^{K_i} \nu_{\hat{F}_{y,x} \to t_i} \tag{17}$$

$$\mu_{t_i \to \hat{F}_{y,x}} = (-1)^{K_i} \nu_{\hat{Y}_i \to t_i} \tag{18}$$

In the factor graph for MRF, messages from FNs to VNs depend on the potential function $V_c(\cdot)$. According to our previous work [7], the message from $M_{y+1,x}$ to $\hat{F}_{y,x}$ is updated as (19), as shown at the bottom of this page. Messages $\mu_{M_{y,x} \to \hat{F}_{y,x-1}}$, $\mu_{N_{y,x} \to \hat{F}_{y,x}}$, and $\mu_{N_{y,x} \to \hat{F}_{y-1,x}}$ can be obtained in a way similar to (19).

Note that as S_j is a constant, it is not necessary to update messages from g_j to S_j. And so is it for messages from t_i to K_i and those from $\hat{F}_{y,x}$ to $P_{y,x}$.

3) Update $\nu_{VN \to FN}$. After completing the message update from FNs to VNs, we further update messages from VNs to FNs. Message $\nu_{\hat{Y}_i \to g_j}$ is calculated via the product operation of the SPA as:

$$\nu_{\hat{Y}_i \to g_j} = \sum_{g_o \in N\left(\hat{\mathrm{Y}}_i\right) \backslash g_j} \mu_{g_o \to \hat{Y}_i} \tag{20}$$

where $N\left(\hat{\mathrm{Y}}_i\right) \backslash g_j$ denotes the neighborhood of $\hat{Y}_i$ excluding g_j. Similarly, message $\nu_{\hat{Y}_i \to t_i}$ is

$$\nu_{\hat{Y}_i \to t_i} = \sum_{g_o \in N\left(\hat{\mathrm{Y}}_i\right) \backslash t_i} \mu_{g_o \to \hat{Y}_i} \tag{21}$$

The message of $\nu_{\hat{F}_{y,x} \to t_i}$ from $\hat{F}_{y,x}$ to t_i is updated as:

$$\nu_{\hat{F}_{y,x} \to t_i} = \sum_{o \in N\left(\hat{F}_{y,x}\right) \backslash t_i} \mu_{o \to \hat{F}_{y,x}} \tag{22}$$

In a similar way, the message of $\nu_{\hat{F}_{y,x} \to M_{y,x}}$ from $\hat{F}_{y,x}$ to $M_{y,x}$ is computed as:

$$\nu_{\hat{F}_{y,x} \to M_{y,x}} = \sum_{o \in N\left(\hat{F}_{y,x}\right) \backslash M_{y,x}} \mu_{o \to \hat{F}_{y,x}} \tag{23}$$

Other messages $\nu_{\hat{F}_{y,x} \to M_{y,x-1}}$, $\nu_{\hat{F}_{y,x} \to N_{y,x}}$, and $\nu_{\hat{F}_{y,x} \to N_{y-1,x}}$ are akin to (23) and thus omitted here for compactness.

Note that messages $\nu_{S_j \to g_j}$ and $\nu_{K_i \to t_i}$ are the same to those in (14) and (15), respectively.

4) Optimal estimation of $\hat{Y}$. To check whether the SPA converges, we need to estimate the encrypted sequence $\hat{Y}$. First, messages from the neighborhood of $\hat{Y}_i$ are accumulated, i.e.,

$$\nu_{\hat{Y}_i} = \sum_{\alpha \in N\left(\hat{Y}_i\right)} \mu_{\alpha \to \hat{Y}_i} \tag{24}$$

Regarding that the message is defined as the logarithm likelihood ratio, $\hat{Y}_i$ is then determined as:

$$\hat{Y}_i = \begin{cases} 0 & \text{if } \nu_{\hat{Y}_i} \geq 0 \\ 1 & \text{otherwise} \end{cases} \tag{25}$$

5) Convergence evaluation. After deciding the sequence of $\hat{Y}$, multiplying the $\hat{Y}$ with the check matrix of H to generate

$$\mu_{M_{y,x} \to \hat{F}_{y,x}} = \ln\left(\frac{\exp\left(\nu_{\hat{F}_{y,x-1} \to M_{y,x}} - \frac{V_c(\hat{F}_{y,x}=0|\hat{F}_{y,x-1}=0)}{T}\right) + \exp\left(-\frac{V_c(\hat{F}_{y,x}=0|\hat{F}_{y,x-1}=1)}{T}\right)}{\exp\left(\mu_{\hat{F}_{y,x-1} \to M_{y,x}} - \frac{V_c(\hat{F}_{y,x}=1|\hat{F}_{y,x-1}=0)}{T}\right) + \exp\left(-\frac{V_c(\hat{F}_{y,x}=1|\hat{F}_{y,x-1}=1)}{T}\right)}\right) \tag{19}$$

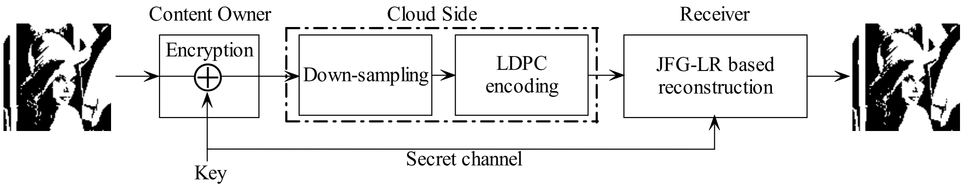


FIGURE 10. Illustration of the proposed scheme.

the syndrome $\hat{S}$, i.e., $\hat{S} = H \cdot \hat{Y}$. If $\hat{S}$ is equal to the received syndrome S, then the SPA is believed to converge and the iteration stops. Otherwise, continue to perform steps 2-4 until the SPA converges or the predefined maximum iteration number reaches. In the latter situation, the LDPC decoding fails, and the code rate of LDPC needs to decrease. That is, the number of syndrome bits should increase.

6) Optimum estimation of $\hat{I}(x, y)$. When the convergence reaches, the $\hat{Y}_i$ is first determined via (24) and (25), and its decrypted version, $\hat{F}_{y,x}$, is then obtained as

$$\hat{F}_{y,x} = \hat{Y}_i \oplus K_i \tag{26}$$

As a result, the down-sampled portion of the reconstructed image $\hat{I}(x, y)$ is decided.

For the missing portion of $\hat{I}(x, y)$, they can be recovered in the following way. First, messages passed to VN $\hat{F}_{y,x}$ that represents a missing pixel are accumulated with the same way as (24). Subsequently, the missing pixel is determined via the same soft-decision in (25).

It is worth pointing out that although the down-sampled portion of $\hat{I}(x, y)$ can also be recovered via the method for the reconstruction of the missing portion, it would possibly lead to a number of errors due to the interference from the missing portion.

IV. PROPOSED SCHEME

In this section, we present the proposed lossy compression scheme for encrypted binary images, as illustrated in Fig. 10. It involves three parts, i.e., image encryption, compression using the technique of down-sampling and LDPC encoding, and the lossy reconstruction exploiting the JFG-LR. Details for these three parts are given below.

A. IMAGE ENCRYPTION

We encrypt the given binary image $I(x, y)$ of size $W \times B$ via the stream cipher, which is secure according to Shannon [30]. First, a sequence of pseudo random bits with length $W \times B$ is generated via a secret key KEY_1, which is denoted as $K = \{K_i | i = 1, \ldots, WB\}$. Second, Image $I(x, y)$ is scanned row-by-row to form a sequence of length $W \times B$, i.e., I_i $(i = 1, \ldots, WB)$. Finally, the I_i is encrypted by imposing the XOR operation, i.e.,

$$C_i = I_i \oplus K_i \; (i = 1, \ldots, WB) \tag{27}$$

The encrypted image sequence C is sent to the cloud side through a public communication channel, while key KEY_1 is passed to the receiver via a secure channel.

B. DOWN-SAMPLED AND LDPC-BASED COMPRESSION

To save the bandwidth and storage space, the cloud side needs to compress the received encrypted sequence C. As the encryption key masks the cover image, the cloud side without access to the encryption key could no longer exploit statistics of the cover image to compress the encrypted signal. Regarding that the encryption key can be taken as the side information at the receiver and the MRF can be used to facilitating the recovery of missing pixels, we adopt the down-sampling and LDPC to compress the C. Details are presented below.

1) DOWN-SAMPLING BASED COMPRESSION

Suppose that $R_d (R_d \geq 0)$ denotes the down-sampling ratio. Then a sequence of length $N = \lceil WBR_d \rceil$ is extracted from C as the compressed sequence, says D, where $\lceil \cdot \rceil$ denotes the ceiling function. To achieve higher compression rate as well as reconstruction quality, we design the following four candidate down-sampling ways.

1) Down-sample the C uniformly. First compute the step as $t = \lfloor WB/N \rfloor$, where $\lfloor \cdot \rfloor$ stands for the flooring function, and then select encrypted bits at location $it + 1$ $(i = 0, 1, \ldots, N - 1)$ as the down-sampled sequence.
2) Down-sample the C in a random way. First generate a pseudo random sequence with a secret key KEY_2, and then sort the sequence in ascending order. Next, obtain N coordinates corresponding to the first N values of the sorted sequence. Finally take the encrypted bits located on the selected N coordinates as the down-sampled version.
3) Down-sample the C in a block-uniform way. After re-shaping the C into an encrypted image of size $W \times B$, divide the resulted image into non-overlapped blocks with size $U \times U$. Then extract randomly a sequence of length $\lceil U^2 R_d \rceil$ from each block and obtain a down-sampled sequence with total length N.
4) Down-sample the C in a comprehensive manner. Specifically, if $R_d \geq 0.25$ holds, then extract the encrypted bits at locations $(2i - 1)W + 2j$ $(i \in [1, B/2],\; j \in [1, W/2])$ of sequence C as one part of the down-sampled portion, which leads to $WB/4$ encrypted bits. Next, remove the down-sampled part from the C, which results in a sub-sequence, says C'. Finally, choose $N - WB/4$ encrypted bits from the C' in a random way similar to method 2, which forms the other part of the down-sampled portion.

In the situation of $R_d \in [0.11,\ 0.25)$, encrypted bits at locations $(3i-1)W+3j$ are down-sampled from the C, and $N-WB/9$ encrypted bits are extracted randomly with the similar approach for $R_d \geq 0.25$. This down-sampling technique can be extended to other cases like $R_d \in [0.06,\ 0.11)$, $R_d \in [0.04,\ 0.06)$, etc.

In summary, the first down-sampling approach is totally uniform, the second is completely random, and the last two take into account both uniformness and randomness. Among them, the one lead to the best compression efficiency for encrypted binary images would be chosen via experimental simulations as the practically optimum one, which will be deferred to Section 5 for compactness of this section.

2) LDPC-BASED COMPRESSION

As demonstrated by Johnson [1], the ETC (encryption-then-compression) system can be formulated as the problem of distributed source coding with side information at the receiver side and the channel code like the LPDC can thus be employed to compress the encrypted signal. By following this framework, we conduct the LDPC encoding on the D to further compress the down-sampled encrypted sequence D. In more detail, denote the check matrix of LDPC as H, and then generate the syndrome S in the following way, i.e.,

$$S = H \cdot D \tag{28}$$

where sizes of H and D are $M \times N$ and $N \times 1$, respectively. The M depends on the LDPC code rate R_c ($R_c > 0$), i.e., $M = N(1-R_c)$.

In recovering the D from the syndrome sequence of S, the subsequence of K, says K', that is used to generate the D is taken as the side information at the receiver. That is, the correlation between K' and D is essentially exploited in the LDPC decoding to recover the D. When the number of bits 0 and 1 in the cover image is nearly the same, however, the correlation between K' and D would decrease significantly, which in turn would make the LDPC decoding fail at the same R_c. In this situation, the doping technique is deployed [3], which directly sends the encrypted bits to the receiver for facilitating the LDPC decoding. As the doped bits are not imposed with the LDPC encoding and they can be directly decrypted to recover the original unencrypted bits and thus help to lead the LDPC decoding to find directions towards convergence.

As directly sending an encrypted bit b_{enck} ($k = 1,\ 2,\ \ldots$) to the receiver is essentially equivalent to multiply a vector $v = [0,\ \ldots,\ 0,\ 1,\ 0,\ \ldots.,\ 0]$ with $b = [b_{enc1},\ \ldots,\ b_{enc(k-1)},\ b_{enck},\ b_{enc(k+1)},\ \ldots.,\ b_{encN}]$, where v has only one "1" and the index of 1 in v is the same as that of b_{enck} in b. In view of this, the doping can be practically implemented via the modified $\bar{H}$ that is constructed as

$$\bar{H} = \begin{bmatrix} H \\ V \end{bmatrix}, \tag{29}$$

where V contains a number of v. Suppose that R_p denotes the doping rate, i.e., the ratio between the number of directly sent encrypted bits and the M. Then the size of V is calculated as $(M \times R_p) \times N$. By substituting the H in (28) with the $\bar{H}$ in (29), we can obtain the final compressed and encrypted sequence $\bar{S}$.

The cloud side transmits the $\bar{S}$ and the keys for down-sampling to the receiver via public and secure channels, respectively. Therefore, the compression rate is computed as:

$$CR_{\text{MRF}} = \frac{M\left(1+R_p\right)}{WB} = \frac{\lceil WBR_d \rceil\,(1-R_c)\left(1+R_p\right)}{WB}. \tag{30}$$

C. LOSSY RECONSTRUCTION USING THE JFG-LR

After receiving the $\bar{S}$ and related keys (i.e., the encryption key KEY_1 and the down-sampling key KEY_2), the receiver deploys the JFG-LR to reconstruct the original binary image in a lossy way. In more detail, it executes the adapted SPA in Section III-C on the JFG-LR to reconstruct both the down-sampled and missing portions, as illustrated in Fig. 9.

In the reconstruction, the discontinuity-adaptive potential function [31], [7] is adopted in our scheme. The function is expressed as

$$\begin{aligned} V_c(f_1|f_2) = V_c(f_2|f_1) &= \log\left[\delta^2 + (f_1 - f_2)^2\right] \\ &\quad + \frac{1}{\delta^2 + (f_1 - f_2)^2} - \log \delta^2 - \frac{1}{\delta^2}, \end{aligned} \tag{31}$$

where f_1 and f_2 are pairwise elements in a clique of a given random field, and δ denotes a model parameter to control the sharpness of edges. This kind of potential function would constrain the reconstructed local areas to be smooth when $\Delta = f_1 - f_2$ is small, while it would preserve textures and edges of the reconstructed local areas when Δ is large.

As the MRF can well represent the statistical correlation between image pixels, it would really facilitate inferring the missing pixels discarded in the down-sampling process as well as the LDPC decoding for the down- sampled portion. Consequently, it would well improve the compression efficiency and the quality of reconstructed image, as will be demonstrated immediately in the next section.

V. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we assess the proposed lossy compression scheme for encrypted binary images via experimental simulations. In the following subsections, we give experimental settings for the proposed scheme, determine the preferable down-sampling method, illustrate the reconstruction process, and evaluate the performance of the proposed scheme by comparing it with the JBIG2 using the original, unencrypted binary image as input.

A. EXPERIMENTAL SETTINGS

In the simulation, we test eleven 100 × 100 binary images with different texture characteristics, as shown in Fig. 11. These images are obtained from the corresponding grey images through binarization with threshold 128.

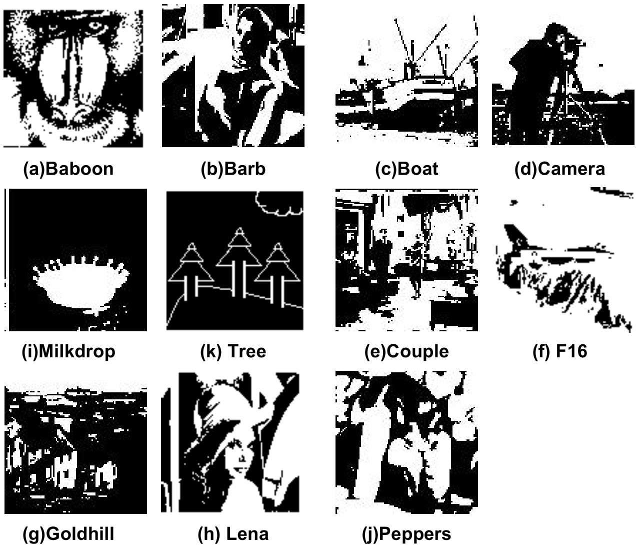


FIGURE 11. Test binary images.

According to (9)-(11) and (31), the MRF (Markov random field) in our scheme involves parameters δ, T, and P. According to our previous work [8], $\delta = 45$ and $T = 0.00049$ are preferable settings for the MRF; and $P = 0.35$ and $P = 0.5$ are used for the non-doping and doping situations, respectively. This is because $P = 0.35$ could provide extra a priori information when the doping technology is not employed. In case of doping, the setting of $P = 0.35$ may deviate from the practical probability distributions and thus would become an interference. As a result, $P = 0.5$ is preferred.

Considering that the proposed scheme is a compression method for encrypted binary images, we employ the metrics of bit per pixel (bpp) and bit error rate (BER) to measure the performance. The bpp is calculated according to (30), and the BER is computed as:

$$BER = \frac{\sum_{x,y} |I(x,y) - I^r(x,y)|}{W \times B}, \tag{32}$$

where $I(x, y)$ and $I^r(x, y)$ denote the original and reconstructed binary images, respectively. On the condition of the same bpp, the smaller the BER is, the better the performance would be. Note that although the peak signal-to-noise ratio (PSNR) can be equivalently obtained from the BER value, the BER better represents reconstructed errors and is thus deployed in our simulation.

B. DETERMINATION OF THE PREFERABLE DOWN-SAMPLING METHOD

As described in Section IV-B, four methods are developed to perform the down-sampling, which are denoted as *UNIFORM*, *RAND*, *BLK-Uni-Rand*, and *Pxl-Uni-Rand* for notational convenience. To evaluate their performance, we adopt the parameter settings in Section V-A and set the down-sampling rates to be $R_d \in [0.3, 1]$ with step 0.1. For each R_d, the LDPC code rates, $R_c \in [0.425, 0.725]$ with step 0.025, are sought to find the desirable code rate leading to the minimum compression ratio. Under this setting, each test image is encrypted, compressed, and reconstructed via the proposed scheme in Section IV.

Because of the involved doping technique, different R_d and R_c may lead to nearly identical BERs. To alleviate this issue, we divide compression ratios in terms of bpp into a number of groups, each of which covers an interval of 0.025 bpp, and then choose the minimum BER and the corresponding bpp in a given group as the preferable bpp-BER performance. Fig. 12 illustrates the bpp-BER performance for images Tree, Lena, Baboon, and F16, where the case of BER=0 corresponds to the lossless compression of a given encrypted binary image [8] while the others denote the lossy one. Other images have similar results.

According to the experimental simulation, image Tree does not need the doping technique for it has a high ratio of bit 0, while the other images involve the doping. It is observed from Fig. 12 that among the four down-sampling methods, the Pxl-Uni-Rand generally leads to the minimum BER on the condition of the same bpp. This can be expected as the Pxl-Uni-Rand well exploits the uniformness and randomness. That is, the uniformly down-sampled part of the Pxl-Uni-Rand facilitates the MRF to infer the discarded pixels from the down-sampled ones, while the randomly down-sampled part probably help to adapt to edges and textures that are usually hard to infer from the neighboring pixels. By following this rationale, the performance for the Blk-Uni-Rand, UNIFORM, and RAND can be well explained, among which the Blk-Uni-Rand trading-off the uniformness and randomness has desirable BER-bpp performance while the RAND with total randomness leads to the worst performance.

Therefore, the Pxl-Uni-Rand is adopted in our scheme as the practically optimum down-sampling method.

C. ILLUSTRATION OF RECONSTRUCTION PROCESS

To illustrate the reconstruction process, in this subsection we take image Lena for example. The image is of size 100 × 100, and has 50.6% non-zero bits. In the simulation, we set the down-sampling rate R_d and the LDPC code rate R_c to be 90% and 0.625, respectively, and seek the optimal doping rate via a binary search. Experimental simulation shows that the resulted compression rate and BER are 0.381 and 0.006, respectively, and the LDPC decoding converges in 91 iterations. Fig. 13 presents the original image, the stream-ciphered version, and the reconstructed images in 21, 41, 61, 71, 81, and 91 iterations. It is found that although the encrypted image has been significantly compressed (i.e., at rate 0.381 bpp), the reconstructed image by means of the proposed scheme is rather close to the original one.

D. PERFORMANCE EVALUATION

In this subsection, we assess the proposed scheme by comparing it with prior arts. Although there exist many compression approaches for unencrypted binary images, to our best knowledge there are few methods focusing on the compression of

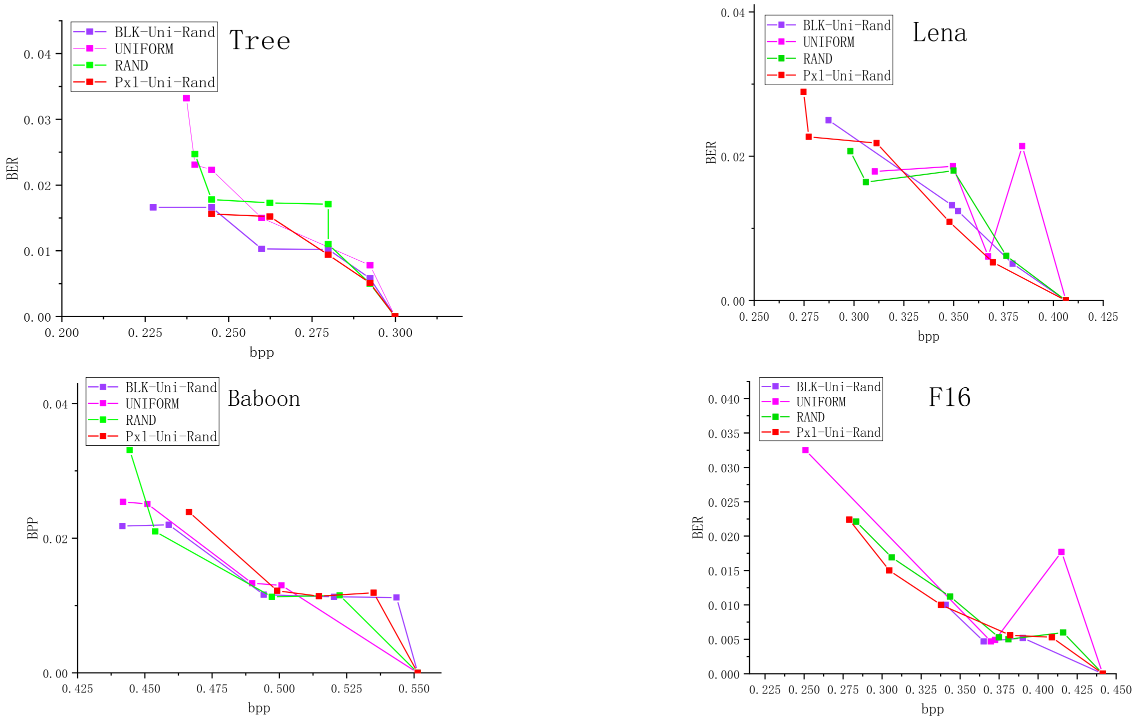


FIGURE 12. The bpp-PSNR performance for different down-sampling methods.

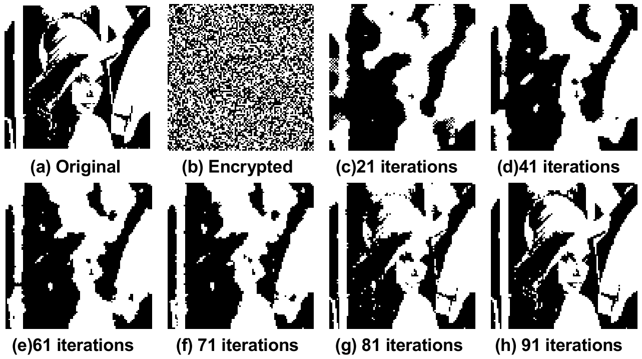

(a) Original
(b) Encrypted
(c)21 iterations
(d)41 iterations
(e)61 iterations
(f) 71 iterations
(g) 81 iterations
(h) 91 iterations

FIGURE 13. Illustration of reconstruction process for Lena.

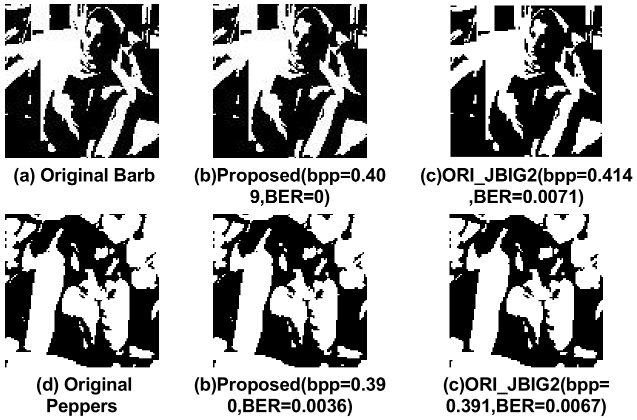

(a) Original Barb
(b)Proposed(bpp=0.409,BER=0)
(c)ORI_JBIG2(bpp=0.414,BER=0.0071)
(d) Original Peppers
(b)Proposed(bpp=0.390,BER=0.0036)
(c)ORI_JBIG2(bpp=0.391,BER=0.0067)

FIGURE 14. Evaluation of reconstructed images.

encrypted binary images. To alleviate this situation, we compare the proposed scheme with the JBIG2. The JBIG2 is an image compression standard for unencrypted binary images, which is developed by the Joint Bi-level Image Experts Group. It is suitable for both lossless and lossy compression ways, but it does not specify the algorithm for lossy compression. Actually, the JBIG2 concatenates an additional block of lossy-compression-based preprocessing with the lossless compression block to achieve the objective of lossy compression.

In the following subsections, we first examine the visual quality of the reconstructed images for both the proposed scheme and the JBIG2, and then evaluate their bpp-BER performance. In the simulation, we adopt the parameter settings in Section V-A and the down-sampling method of Pxl-Uni-Rand for the proposed scheme. For the JBIG2, we employ the JPEG as the compression preprocessing block. That is, we map bits 0 and 1 of a given binary image to be 0 and 255, respectively, then compress the mapped image with different quality factors (QFs), say $q \in [1, 100]$ with step 1, and finally input the compressed image to the JBIG2 with lossless compression mode to yield the compressed sequence. Suppose that the length of the compressed sequence is *Len*. Then the compression rate for the JBIG2 is computed as:

$$CR_{JBIG2} = \frac{Len}{W \times B}. \tag{33}$$

The BER is obtained via (29). As different QFs may result in close BERs, the minimum BER at nearly identical bpp is taken as the BER value for the bpp.

In addition, considering that the JBIG2 uses the unecrypted binary image as input, for better assessment we further take the encrypted binary image as input for the JBIG2. This

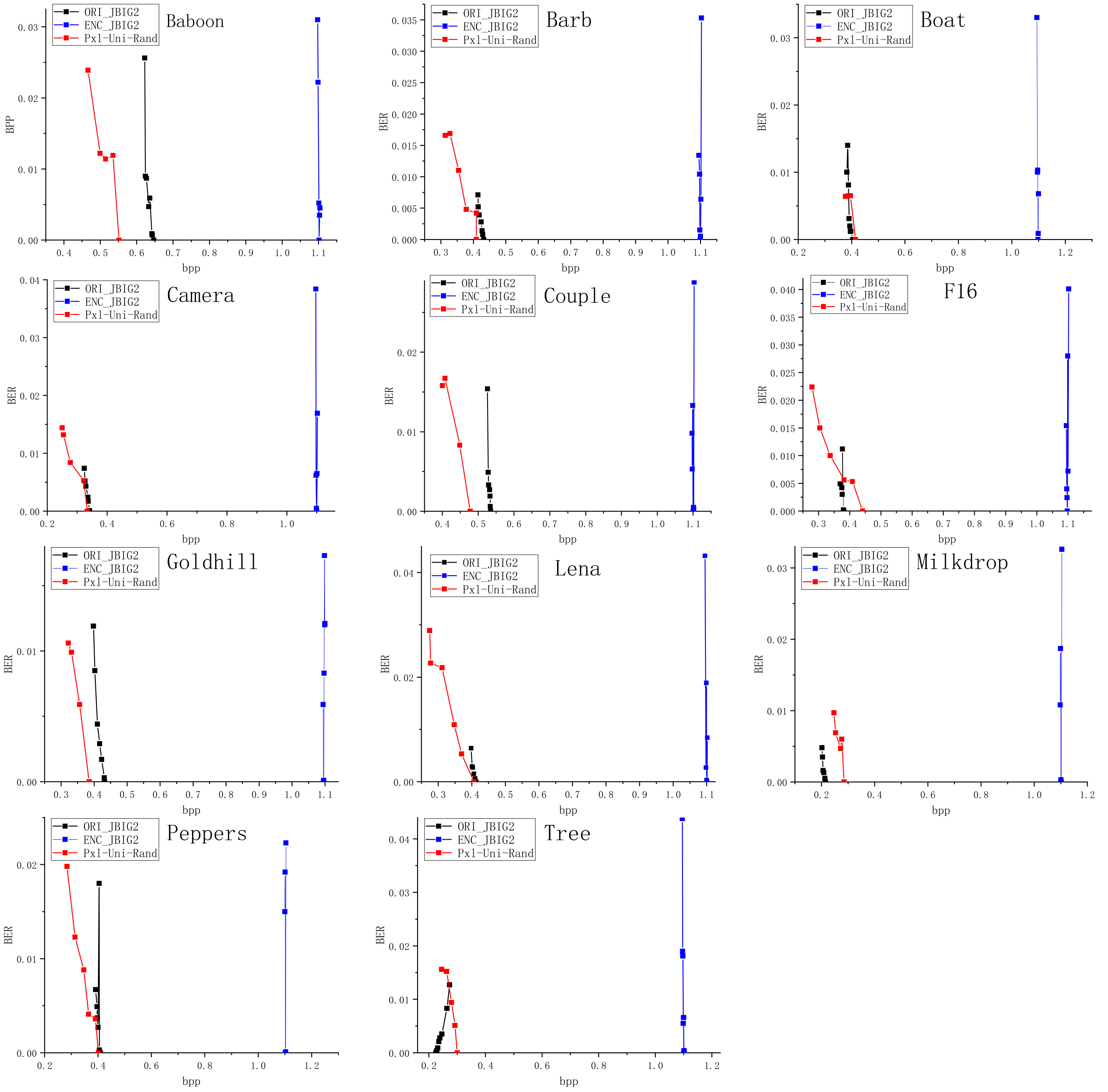


FIGURE 15. The bpp-BER performance for the proposed scheme, ORI_JBIG2, and ENC_JBIG2.

then gives rise to two versions of JBIG2, i.e., employing the original unencrypted and encrypted binary images as input, respectively. For convenience, they are denoted as ORI_JBIG2 and ENC_JBIG2, respectively. Details for performance evaluation on the proposed scheme, ORI_JBIG2, and ENC_JBIG2 are given below.

1) EXAMINATION ON QULITY OF RECONSTRUCTED IMAGES

By deploying the aforementioned settings, we encrypt, compress, and reconstruct test binary images in Fig. 11 via the proposed scheme in Section IV. Also, we conduct the JPEG compression on images mapped from test binary images followed by performing the conventional JBIG2 with the lossless mode. Fig. 14 presents the original binary images of Barb and F16, and their reconstructed versions by the proposed scheme and the ORI_JBIG2. It is seen that the reconstructed images by the proposed scheme and the ORI_JBIG2 are rather close to the original versions. Considering that the ORI_JBIG2 takes the original unencrypted binary image as the input for the compressor while the proposed scheme uses the encrypted one as the input of the compressor, this result well demonstrates the effectiveness of the proposed scheme. Other images have similar results, which are omitted here for compactness.

By the way, as the compression rate for the ENC_JBIG2 is larger than 1, it is hard to make a fair examination on the quality of reconstructed images. Thus, results for the ENC_JBIG2 is not presented here.

2) ASSESSMENT ON THE BPP-BER PERFORMANCE

Fig. 15 summarizes the bpp-BER performance for all 11 test binary images. It is observed that BERs of the proposed scheme gradually decreases when the bpp increases little by little. This is because the increase of bpp implies the decrease of compression rate, i.e., the decrease of discarded image bits, which clearly would reduce the BER. Nevertheless, as the Pxl-Uni-Rand takes into account both the uniformness and randomness, the involved randomness may result in different doping rates and thus lead to small fluctuation of BERs, as shown in curves for Baboon, Barb, and F16.

It is found from Fig. 15 that on the condition of the same bpp, the proposed scheme using the down-sampling method of Pxl-Uni-Rand generally has smaller BERs than the ORI_JBIG2 and ENC_JBIG2. It is conjectured that the proposed scheme exploits the MRF that can well characterize the spatial statistics of binary images and thus can well infer the discarded bits from the down-sampled ones. Nevertheless, for images F16, Milkdrop, and tree, the proposed scheme is somewhat inferior to the ORI_JBIG2, which may attribute to the fact that the ORI_JBIG2 has higher compression efficiency for images with a large portion of smooth regions. Since the ORI_JBIG2 compresses the unencrypted original images while the proposed scheme condenses the encrypted images, the results well demonstrate the feasibility and effectiveness of the proposed scheme.

In addition, it is noted that the performance of the ENC_JBIG2 is far worse than the proposed scheme and the ORI_JBIG2. This is because the input image of the ENC_JBIG2 has been encrypted and the conventional JBIG2 can no longer exploit the statistics of input image to conduct the efficient compression. By the way, as the ENC_JBIG2 cannot compress the encrypted image and extra bits are required to save the auxiliary information of the JBIG2, its compression rate is larger than 1 bpp, as shown in Fig. 15.

VI. CONCLUSION

This paper has presented an MRF-based lossy compression scheme for encrypted binary images. The original binary image is encrypted with the stream cipher, and the encrypted image is then compressed via the successive down-sampling and LDPC encoding. In lossy reconstruction, the reconstruction problem is formulated as an optimization problem, and the joint factor graph, i.e., JFG-LR is constructed to solve this optimization problem. By deriving the SPA (sum-product algorithm) adapted to the JFG-LR followed by running the adapted SPA on the JFG-LR iteratively, the original binary image is thus recovered in a lossy way. Experimental results show that the proposed scheme achieves preferable compression efficiency and is comparable or even better than the JBIG2 with the original, unencrypted binary images as input. This well demonstrates the feasibility and effectiveness of the proposed scheme.

REFERENCES

[1] M. Johnson, P. Ishwar, V. Prabhakaran, D. Schonberg, and K. Ramchandran, "On compressing encrypted data," *IEEE Trans. Signal Process.*, vol. 52, no. 10, pp. 2992–3006, Oct. 2004.

[2] D. Schonberg, S. Draper, and K. Ramchandran, "On compression of encrypted images," in *Proc. Int. Conf. Image Process.*, Atlanta, GA, USA, Oct.. 2006, pp. 269–272.

[3] D. Schonberg, S. C. Draper, C. Yeo, and K. Ramchandran, "Toward compression of encrypted images and video sequences," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 4, pp. 749–762, Dec. 2008.

[4] R. Lazzeretti and M. Barni, "Lossless compression of encrypted grey-level and color images," in *Proc. 16th Eur. Signal Process. Conf. (EUSIPCO)*, Lausanne, Switzerland, Aug. 2008, pp. 1–5.

[5] A. A. Kumar and A. Makur, "Distributed source coding based encryption and lossless compression of gray scale and color images," in *Proc. IEEE 10th Workshop Multimedia Signal Process.*, Cairns, QLD, Australia, Oct. 2008, pp. 760–764.

[6] W. Liu, W. Zeng, L. Dong, and Q. Yao, "Efficient compression of encrypted grayscale images," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 1097–1102, Apr. 2010.

[7] J. Zhou, X. Liu, O. C. Au, and Y. Tang, "Designing an efficient image encryption-then-compression system via prediction error clustering and random permutation," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 1, pp. 39–50, Jan. 2014.

[8] C. Wang, J. Ni, X. Zhang, and Q. Huang, "Efficient compression of encrypted binary images using the Markov random field," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1271–1285, May 2018.

[9] C. Wang, Y. Feng, T. Li, H. Xie, and G.-R. Kwon, "A new encryption-then-compression scheme on gray images using the Markov random field," *Comput., Mater., Continua*, vol. 56, no. 1, pp. 107–121, Aug. 2018.

[10] A. A. Kumar and A. Makur, "Lossy compression of encrypted image by compressive sensing technique," in *Proc. IEEE Region Conf. (TENCON)*, Singapore, Nov. 2009, pp. 1–6.

[11] X. Zhang, Y. Ren, G. Feng, and Z. Qian, "Compressing encrypted image using compressive sensing," in *Proc. 2011 Seventh Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, Dalian, China, Oct. 2011, pp. 222–225.

[12] C. Song, X. Lin, and X. Shen, "Secure and effective image storage for cloud based e-healthcare systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Atlanta, GA, USA, Dec. 2013, pp. 653–658.

[13] X. Zhang, "Lossy compression and iterative reconstruction for encrypted image," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 1, pp. 53–58, Mar. 2011.

[14] X. Zhang, G. Feng, Y. Ren, and Z. Qian, "Scalable coding of encrypted images," *IEEE Trans. Image Process.*, vol. 21, no. 6, pp. 3108–3114, Jun. 2012.

[15] X. Zhang, G. Sun, L. Shen, and C. Qin, "Compression of encrypted images with multi-layer decomposition," *Multimed Tools Appl.*, vol. 72, no. 1, pp. 489–502, Sep. 2014.

[16] X. Zhang, Y. Ren, L. Shen, Z. Qian, and G. Feng, "Compressing encrypted images with auxiliary information," *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1327–1336, Aug. 2014.

[17] R. Hu, X. Li, and B. Yang, "A new lossy compression scheme for encrypted gray-scale images," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Florence, Italy, May 2014, pp. 7387–7390.

[18] C. Wang and J. Ni, "Compressing encrypted images using the integer lifting wavelet," in *Proc. Int. Conf. Intell. Inf. Hiding Multimedia Signal Process. (IIH-MSP)*, Adelaide, SA, Australia, Sep. 2015, pp. 409–412.

[19] C. Wang, J. Ni, and Q. Huang, "A new encryption-then-compression algorithm using the rate–distortion optimization," *Signal Process., Image Commun.*, vol. 39, pp. 141–150, Nov. 2015.

[20] C. Wang, D. Xiao, H. Peng, and R. Zhang, "A lossy compression scheme for encrypted images exploiting Cauchy distribution and weighted rate distortion optimization," *J. Vis. Commun. Image Represent.*, vol. 51, pp. 122–130, Feb. 2018.

[21] C. Qin, Q. Zhou, F. Cao, J. Dong, and X. Zhang, "Flexible lossy compression for selective encrypted image with image inpainting," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 11, pp. 3341–3355, Nov. 2019.

[22] X. Kang, A. Peng, X. Xu, and X. Cao, "Performing scalable lossy compression on pixel encrypted images," *EURASIP J. Image Video Process.*, vol. 2013, May 2013, Art. no. 32.

[23] J. Zhou, O. C. Au, G. Zhai, Y. Y. Tang, and X. Liu, "Scalable compression of stream cipher encrypted images through context-adaptive sampling," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 11, pp. 1857–1868, Nov. 2014.

[24] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithms," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[25] Y. Rozanov, *Markov Random Fields*. New York, NY, USA: Springer-Verlag, 1982.

[26] S. Li, *Markov Random Field Modeling in Computer Vision*. Tokyo, Japan: Springer-Verlag, 1995.

[27] J. M. Hammersley and P. Clifford, "Markov fields on finitegraphs and lattices," in *Low-Density Parity-Check Codes*, R. C. Gallager, Ed. Cambridge, MA, USA: MIT Press, 1963.

[28] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics*. Reading, MA, USA: Addison-Wesley, 1989.

[29] P. Levy, "A special problem of Brownian motion, and a general theory of Gaussian random unctions," in *Proc. 3rd Berkeley Symp. Math. Statist. Probab.*, Berkeley, CA, USA, 1956.

[30] J. W. Woods, "Two-dimensional discrete Markov random field," *IEEE Trans. Inf. Theory*, vol. IT-18, no. 2, pp. 232–240, Mar. 1972.

[31] R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 533–547, Sep. 1981.

[32] D. Mackay and R. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 33, no. 6, p. 457, Jul. 1997.

[33] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656–715, Oct. 1949.

[34] E. Ising, "Beitrag zur theorie des ferromagnetismus [Report on the theory of ferromagnetism]," *Zeitschrift Physik*, vol. 31, no. 1, pp. 253–258, 1925.

[35] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.

[36] Z. Mouyan, *Deconvolution and Signal Recovery*. Beijing, China: National Defence Industry Press, 2001.

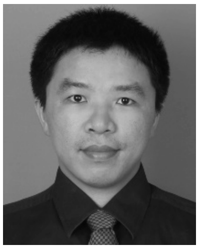
[37] J.-y. Wu and Q.-Q. Ruan, "A new hybrid PDE denoising model based on Markov random field," in *Proc. 1st Int. Conf. Innov. Comput., Inf. Control (ICICIC)*, Beijing, China, vol. 1, Oct. 2006, pp. 363–372.

[38] P. Yahampath, "Joint source-decoding in large scale sensor networks using Markov random field models," *Signal Process.*, vol. 90, no. 12, pp. 3134–3146, Dec. 2010.

[39] O. Al-Shaykh and R. Mersereau, "Lossy compression of noisy images," *IEEE Trans. Image Process.*, vol. 7, no. 12, pp. 1641–1652, Dec. 1998.

[40] A. Amraoui and R. Urbanke. (2003). *LDPCOpt*. [Online]. Available: http://lthcwww.epfl.ch/research/ldpcopt/

[41] X.-Y. Hu, E. Eleftheriou, and D. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.

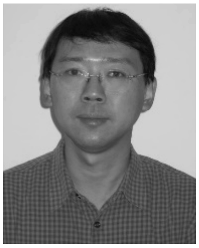


CHUNTAO WANG (Member, IEEE) received the B.S. and Ph.D. degrees from Sun Yat-sen University, in 2002 and 2007, respectively.

From October 2007 to September 2008, he was a Postdoctoral Fellow with Korea University, South Korea. From November 2008 to November 2010, he was a Postdoctoral Researcher with Sun Yat-sen University. He is currently an Associate Professor with the School of Mathematics and Informatics, South China Agricultural University. His research interests include information hiding and multimedia signal processing.

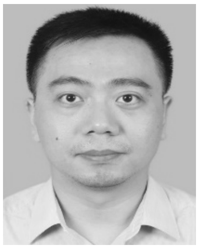
TIANZHENG LI received the B.S. degree in electronic information science and technology from Huizhou University, in 2017. He is currently pursuing the M.S. degree with the College of Mathematics and Informatics, South China Agricultural University, Guangzhou.

His research interests include the processing of encrypted signals and embedding system development.



JIANGQUN NI (Member, IEEE) received the Ph.D. degree in electronic engineering from The University of Hong Kong, Hong Kong, in 1998.

He was a Postdoctoral Fellow for a joint program between Sun Yat-sen University, Guangzhou, China, and the Guangdong Institute of Telecommunication Research, from 1998 to 2000. Since 2001, he has been with the School of Data and Computer Science, Sun Yat-sen University, where he is currently a Professor. His research interests include data hiding and digital image forensics, image-based modeling and rendering, and image/video processing.



QIONG HUANG received the B.S. and M.S. degrees from Fudan University, in 2003 and 2006, respectively, and the Ph.D. degree from the City University of Hong Kong, in 2010.

He is currently a Professor with the College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China. His research interests include cryptography and information security, in particular, cryptographic protocols design and analysis. He has published more than 100 research articles in international conferences and journals, and served as a program committee member in many international conferences.