mean and the variance \( \alpha^2 \sigma^2 \), and let \( \mu \) denote the Gaussian distribution with zero mean and the variance \( \sigma^2 \). Then, the function \( f^* \) is given as follows:

\[
f^*(t) = P^{-1}_\nu(\nu_p(t))
\]

(31)

where \( P^{-1}_\nu \) is the inverse cumulative distribution function of the Gaussian distribution \( \nu \), and \( P_\nu \) is the cumulative distribution function of the Gaussian distribution \( \nu \). It is easy to show that

\[
P^{-1}_\nu(z) = \sqrt{2\alpha \sigma^2} \text{erf}^{-1}\left(\frac{2z-1}{2}\right),
\]

(32)

\[
P_\nu(t) = \frac{1}{2} \left[ 1 + \text{erf}\left(\frac{t}{\sigma \sqrt{2}}\right)\right]
\]

(33)

By substituting (32) and (33) into (31), we can deduce that

\[
f^*(t) = \alpha t
\]

(34)

which implies that the optimal way of altering the host signal on \( v_j \) is given by \( v_j(t) = \alpha v_j(t) \).

## References


**Abstract**—This work proposes a novel scheme for separable reversible data hiding in encrypted images. In the first phase, a content owner encrypts the original uncompressed image using an encryption key. Then, a data-hider may compress the least significant bits of the encrypted image using a data-hiding key to create a sparse space to accommodate some additional data. With an encrypted image containing additional data, if a receiver has the data-hiding key, he can extract the additional data though he does not know the image content. If the receiver has the encryption key, he can decrypt the received data to obtain an image similar to the original one, but cannot extract the additional data. If the receiver has both the data-hiding key and the encryption key, he can extract the additional data and recover the original content without any error by exploiting the spatial correlation in natural image when the amount of additional data is not too large.

**Index Terms**—Image encryption, image recovery, reversible data hiding.

**I. Introduction**

In recent years, signal processing in the encrypted domain has attracted considerable research interest. As an effective and popular means for privacy protection, encryption converts the ordinary signal into unintelligible data, so that the traditional signal processing usually takes place before encryption or after decryption. However, in some scenarios that a content owner does not trust the processing service provider, the ability to manipulate the encrypted data when keeping the plain content unrevealed is desired. For instance, when the secret data to be transmitted are encrypted, a channel provider without any knowledge of the cryptographic key may tend to compress the encrypted data due to the limited channel resource. While an encrypted binary image can be compressed with a lossless manner by finding the syndromes of low-density parity-check codes [1], a lossless compression method for encrypted gray image using progressive decomposition and rate-compatible punctured turbo codes is developed in [2]. With the lossy compression method presented in [3], an encrypted gray image can be efficiently compressed by discarding the excessively rough and fine information of coefficients generated from orthogonal transform. When having the compressed data, a receiver may reconstruct the principal content of original image by retrieving the values of coefficients. The computation of transform in the encrypted domain has also been studied. Based on the homomorphic properties of the underlying cryptosystem, the discrete Fourier transform in the encrypted domain can be implemented [4]. In [5], a composite signal representation method packing together a number of signal samples and processing them as a unique sample is used to reduce the complexity of computation and the size of encrypted data.

Manuscript received June 20, 2011; revised November 07, 2011; accepted November 08, 2011. Date of publication November 15, 2011; date of current version March 08, 2012. This work was supported by the National Natural Science Foundation of China under Grant 61073190, Grant 61103181, and Grant 6083810, by the Research Fund for the Doctoral Program of Higher Education of China under Grant 2011130811010, and by the Alexander von Humboldt Foundation. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Alessandro Piva.

The author is with the School of Communication, Shanghai University, Shanghai 200072, China (e-mail: xzhang@shu.edu.cn).

Digital Object Identifier 10.1109/TIFS.2011.2176120
The reversible data hiding in encrypted image is investigated in [12]. Most of the work on reversible data hiding focuses on the data embedding/extraction on the plain spatial domain [13]–[17]. But, in some applications, an inferior assistant or a channel administrator hopes to append some additional message, such as the origin information, image notation or authentication data, within the encrypted image though he does not know the original image content. And it is also hoped that the original content should be recovered without any error after image decryption and message extraction at receiver side. Reference [12] presents a practical scheme satisfying the above-mentioned requirements and Fig. 1 gives the sketch. A content owner encrypts the original uncompressed image using an encryption key to produce an encrypted image. Then, the data-hider compresses the least significant bits (LSB) of the encrypted image using a data-hiding key to create a sparse space to accommodate the additional data. At the receiver side, the data embedded in the created space can be easily retrieved from the encrypted image containing additional data according to the data-hiding key. Since the data embedding only affects the LSB, a decryption with the encryption key can result in an image similar to the original version. When using both of the encryption and data-hiding keys, the embedded additional data can be successfully extracted and the original image can be perfectly recovered by exploiting the spatial correlation in natural image.

Fig. 1. Sketch of non-separable reversible data hiding in encrypted image.

II. PROPOSED SCHEME

The proposed scheme is made up of image encryption, data embedding and data-extraction/image-recovery phases. The content owner encrypts the original uncompressed image using an encryption key to produce an encrypted image. Then, the data-hider compresses the least significant bits (LSB) of the encrypted image using a data-hiding key to create a sparse space to accommodate the additional data. At the receiver side, the data embedded in the created space can be easily retrieved from the encrypted image containing additional data according to the data-hiding key. Since the data embedding only affects the LSB, a decryption with the encryption key can result in an image similar to the original version. When using both of the encryption and data-hiding keys, the embedded additional data can be successfully extracted and the original image can be perfectly recovered by exploiting the spatial correlation in natural image.

A. Image Encryption

Assume the original image with a size of $N_1 \times N_2$ is in uncompressed format and each pixel with gray value falling into [0, 255] is represented by 8 bits. Denote the bits of a pixel as $b_{i,j,0}, b_{i,j,1}, \ldots, b_{i,j,7}$, where $1 \leq i \leq N_1$ and $1 \leq j \leq N_2$, the gray value as $p_{i,j}$, and the number of pixels as $N = N_1 \times N_2$. That implies

$$b_{i,j,n} = \left \lfloor p_{i,j} / 2^n \right \rfloor \mod 2, \quad n = 0, 1, \ldots, 7$$

(1)
The data-hider may represent the values of $M$, $L$, and $S$ as 2, 14, and 4 bits, respectively, and replace the LSB of selected encrypted pixels with the 20 bits.

In the following, a total of $(N - N_P) \cdot S/L$ bits made up of $N_P$ original LSB of selected encrypted pixels and $(N - N_P) \cdot S/L - N_P$ additional bits will be embedded into the pixel groups. For each group, calculate

$$
\begin{bmatrix}
B'(k, 1) \\
B'(k, 2) \\
\vdots \\
B'(k, mL - S)
\end{bmatrix}
= G \cdot 
\begin{bmatrix}
B(k, 1) \\
B(k, 2) \\
\vdots \\
B(k, mL)
\end{bmatrix}
$$

where the arithmetic is modulo-2. By (5), $[B(k, 1), B(k, 2), \ldots, B(k, M \cdot L)]$ are compressed as $(M \cdot L - S)$ bits, and a spare space is therefore available for data accommodation. Let $[B'(k, M \cdot L - S + 1), B'(k, M \cdot L - S + 2), \ldots, B'(k, M \cdot L)]$ each group be the original LSB of selected encrypted pixels and the additional data to be embedded. Then, replace the $[B(k, 1), B(k, 2), \ldots, B(k, M \cdot L)]$ with the new $[B'(k, 1), B'(k, 2), \ldots, B'(k, M \cdot L)]$, and put them into their original positions by an inverse permutation. At the same time, the $(8-M)$ most significant bits (MSB) of encrypted pixels are kept unchanged. Since $S$ bits are embedded into each pixel-group, the total $(N - N_P) \cdot S/L$ bits can be accommodated in all groups. Clearly, the embedding rate, a ratio between the data amount of net payload and the total number of cover pixels, is

$$
R = \frac{(N - N_P) \cdot S/L - N_P}{N} \approx \frac{S}{L}.
$$

### C. Data Extraction and Image Recovery

In this phase, we will consider the three cases that a receiver has only the data-hiding key, only the encryption key, and both the data-hiding and encryption keys, respectively.

With an encrypted image containing embedded data, if the receiver has only the data-hiding key, he may first obtain the values of the parameters $M$, $L$, and $S$ from the LSB of the $N_P$ selected encrypted pixels. Then, the receiver permutes and divides the other $(N - N_P)$ pixels into $(N - N_P)/L$ groups and extracts the $S$ embedded bits from the $M$ LSB-planes of each group. When having the total $(N - N_P) \cdot S/L$ extracted bits, the receiver can divide them into $N_P$ original LSB of selected encrypted pixels and $(N - N_P) \cdot S/L - N_P$ additional bits. Note that because of the pseudo-random pixel selection and permutation, any attacker without the data-hiding key cannot obtain the parameter values and the pixel-groups, therefore cannot extract the embedded data. Furthermore, although the receiver having the data-hiding key can...
Fig. 3. (a) Original Lena, (b) its encrypted version, (c) encrypted image containing embedded data with embedding rate 0.017 bpp, and (d) directly decrypted version with PSNR 39.0 dB.

successfully extract the embedded data, he cannot get any information about the original image content.

Consider the case that the receiver has the encryption key but does not know the data-hiding key. Clearly, he cannot obtain the values of parameters and cannot extract the embedded data. However, the original image content can be roughly recovered. Denoting the bits of pixels in the encrypted image containing embedded data as $B'_{i,j,0}, B'_{i,j,1}, \ldots, B'_{i,j,7}$ (1 ≤ $i$ ≤ $N_1$ and 1 ≤ $j$ ≤ $N_2$), the receiver can decrypt the received data

\[ k'_{i,j,u} = B'_{i,j,u} \oplus r_{i,j,u} \]  

(7)

where $r_{i,j,u}$ are derived from the encryption key. The gray values of decrypted pixels are

\[ P'_{i,j} = \sum_{u=0}^{7} k'_{i,j,u} \cdot 2^u. \]  

(8)

Since the data-embedding operation does not alter any MSB of encrypted image, the decrypted MSB must be same as the original MSB. So, the content of decrypted image is similar to that of original image. According to (5), if $B(k, M \cdot L - S + 1) = B(k, M \cdot L - S + 2) = \ldots = B(k, M \cdot L) = 0$, there is

\[ B'(k, v) = B(k, v), \quad v = 1, 2, \ldots, M \cdot L - S. \]  

(9)

The probability of this case is $1/2^S$, and, in this case, the original ($M \cdot L - S$) bits in the $M$ LSB-planes can be correctly decrypted. Since $S$ is significantly less than $M \cdot L$, we ignore the distortion at other $S$ decrypted bits. If there are nonzero bits among $B(k, M \cdot L - S + 1)$, $B(k, M \cdot L - S + 2), \ldots, B(k, M \cdot L)$, the encrypted data in the $M$ LSB-planes have been changed by the data-embedding operation, so that the decrypted data in the $M$ LSB-planes differ from the original data. Assuming that the original distribution of the data in the $M$ LSB-planes is uniform, the distortion energy per each decrypted pixel is

\[ D_E = 2^{-2M} \cdot \sum_{\alpha=0}^{2M-1} \sum_{\beta=0}^{2M-1} (\alpha - \beta)^2. \]  

(10)
Because the probability of this case is \(2S - 1\)/2, the average energy of distortion is

\[
A_E = \frac{2S - 1}{2^S}, \quad 2^{-2M} \cdot \sum_{i=0}^{2M-1} \sum_{j=0}^{2M-1} (\alpha - \beta)^2.
\]  

(11)

Here, the distortion in the \(N_r\) selected pixels is also ignored since their number is significantly less than the image size \(N\). So, the value of PSNR in the directly decrypted image is

\[
PSNR = 10 \cdot \log_{10}(A_E).
\]  

(12)

Table I gives the theoretical values of PSNR with respect to \(S\) and \(M\). If the receiver has both the data-hiding and the encryption keys, he may aim to extract the embedded data and recover the original image. According to the data-hiding key, the values of \(M\), \(L\) and \(S\), the original LSB of the \(N_r\) selected encrypted pixels, and the \((N - N_r) \cdot S/L \cdot N_r\) additional bits can be extracted from the encrypted image containing embedded data. By putting the \(N_r\) LSB into their original positions, the encrypted data of the \(N_r\) selected pixels are retrieved, and their original gray values can be correctly decrypted using the encryption keys. In the following, we will recover the original gray values of the other \((N - N_r)\) pixels. Considering a pixel-group, because \(B^r(k, 1), B^r(k, 2), \ldots, B^r(k, M \cdot L - S)\) in (5) are given, \([B(k, 1), B(k, 2), \ldots, B(k, M \cdot L)]^T\) must be one of the vectors meeting

\[
v = [B^r(k, 1), B^r(k, 2), \ldots, B^r(k, M \cdot L - S)00 \cdots 0]^T + a \cdot H
\]  

(13)

where \(a\) is an arbitrary binary vector sized 1 \(\times S\), and \(H\) is an \(S \times ML\) matrix made up of the transpose of \(Q\) and an \(S \times S\) identity matrix

\[
H = [Q^T I_S].
\]  

(14)

In other words, with the constraint of (5), there are \(2^S\) possible solutions of \([B(k, 1), B(k, 2), \ldots, B(k, M \cdot L)]^T\). For each vector \(v\), we attempt to put the elements in it to the original positions to get an encrypted pixel-group and then decrypt the pixel-group using the encryption key. Denoting the decrypted pixel-group as \(G_k\) and the gray values in it as \(t_{i,j}\), calculate the total difference between the decrypted and estimated gray values in the group

\[
D = \sum_{(i,j) \in G_k} |t_{i,j} - \tilde{t}_{i,j}|
\]  

(15)

where the estimated gray values is generated from the neighbors in the directly decrypted image, by (16), as shown at the bottom of the page. Clearly, the estimated gray values in (16) are only dependent on the MSB of neighbor pixels. Thus, we have \(2^S\) different \(D\) corresponding to the \(2^S\) decrypted pixel-group \(G_k\). Among the \(2^S\) decrypted pixel-group, there must be one that is just the original gray values and possesses a low \(D\) because of the spatial correlation in natural image. So, we find the smallest \(D\) and regard the corresponding vector \(v\) as the actual \([B(k, 1), B(k, 2), \ldots, B(k, M \cdot L)]^T\) and the decrypted \(t_{i,j}\) as the recovered content. As long as the number of pixels in a group is sufficiently large and there are not too many bits embedded into each group, the original content can be perfectly recovered by the spatial correlation criterion. Since the \(2^S\) different \(D\) must be calculated in each group, the computation complexity of the content recovery is \(O(N \cdot 2^S)\). On the other hand, if more neighboring pixels and a smarter prediction method are used to estimate the gray values, the performance of content recovery will be better, but the computation complexity is higher. To keep a low computation complexity, we let \(S\) be less than ten and use only the four neighboring pixels to calculate the estimated values as in (16).

III. EXPERIMENTAL RESULTS

The test image Lena sized 512 \(\times\) 512 shown in Fig. 3(a) was used as the original image in the experiment. After image encryption, the eight encrypted bits of each pixel are converted into a gray value to generate an encrypted image shown in Fig. 3(b). Then, we let \(M = \frac{3}{4}, L = \frac{128}{256}, S = 2\) to embed 4.4 \(\times\) 10^{8} additional bits into the encrypted image. The encrypted image containing the embedded data is shown in Fig. 3(c), and the embedding rate \(R\) is 0.017 bit per pixel (bpp). With an encrypted image containing embedded data, we could extract the additional data using the data-hiding key. If we directly decrypted the encrypted image containing embedded data using the encryption key, the value of PSNR in the decrypted image was 39.0 dB, which verifies the theoretical value 39.1 dB calculated by (12). The directly decrypted image is given as Fig. 3(d). By using both the data-hiding and the encryption keys, the embedded data could be successfully extracted and the original image could be perfectly recovered from the encrypted image containing embedded data.

Tables II and III list the embedding rates, PSNR in directly decrypted images and PSNR in recovered images when different \(M\), \(L\) and \(S\) were used for images Lena and Man. As analyzed in (6), the embedding rate is dependent on \(S\) and \(L\), and the larger \(S\) and the smaller \(L\) correspond to a higher embedding rate. On the other hand, the smaller the values of \(M\) and \(S\), the quality of directly decrypted image is better since more data in encrypted image are not changed by data embedding. The “+∞” in Tables II and III indicate that the original images were recovered without any error. Here, the large \(M\), \(L\) and the small \(S\) are helpful to the perfect content recovery since more cover data and less possible solutions are involved in the recovery procedure. If \(M\) and \(L\) are too small or \(S\) is too large, the recovery of original content may be unsuccessful, and the values of PSNR in recovered images are also given in Tables II and III.

Figs. 4–6 show the rate-distortion curves of the four images Lena, Man, Lake and Baboon. Here, three quality metrics were used to measure the distortion in directly decrypted image: PSNR, the Watson metric and a universal quality index \(Q\). While PSNR simply indicates the energy of distortion caused by data hiding, the Watson metric is designed by using characteristics of the human visual system and measures the total perceptual error, which is DCT-based and takes into account three factors: contrast sensitivity, luminance masking and contrast masking [18]. Additionally, the quality index \(Q\) works in
TABLE II
EMBEDDING RATE $R$, PSNR IN DIRECTLY DECRYPTED IMAGES (dB) AND PSNR IN RECOVERED IMAGES (dB) WITH DIFFERENT PARAMETERS FOR TEST IMAGE LENA

<table>
<thead>
<tr>
<th></th>
<th>$S = 1$</th>
<th>$S = 2$</th>
<th>$S = 3$</th>
<th>$S = 4$</th>
<th>$S = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M = 1$</td>
<td>$L = 2000$</td>
<td>0.0005, 54.6, +∞</td>
<td>0.0010, 52.3, +∞</td>
<td>0.0015, 51.6, +∞</td>
<td>0.0020, 51.4, +∞</td>
</tr>
<tr>
<td></td>
<td>$L = 1500$</td>
<td>0.0067, 54.3, +∞</td>
<td>0.0013, 52.2, +∞</td>
<td>0.0020, 51.7, +∞</td>
<td>0.0027, 51.4, +∞</td>
</tr>
<tr>
<td></td>
<td>$L = 1000$</td>
<td>0.00010, 54.3, +∞</td>
<td>0.0020, 52.2, +∞</td>
<td>0.0030, 51.5, 70.4</td>
<td>0.0040, 51.3, 68.2</td>
</tr>
<tr>
<td>$M = 2$</td>
<td>$L = 400$</td>
<td>0.0025, 47.7, +∞</td>
<td>0.0050, 45.3, +∞</td>
<td>0.0075, 44.7, +∞</td>
<td>0.010, 44.3, +∞</td>
</tr>
<tr>
<td></td>
<td>$L = 300$</td>
<td>0.0033, 47.5, +∞</td>
<td>0.0067, 45.3, +∞</td>
<td>0.010, 44.6, +∞</td>
<td>0.013, 44.4, 73.6</td>
</tr>
<tr>
<td></td>
<td>$L = 200$</td>
<td>0.005, 47.6, +∞</td>
<td>0.010, 45.2, +∞</td>
<td>0.015, 44.7, 68.3</td>
<td>0.020, 44.4, 62.6</td>
</tr>
<tr>
<td>$M = 3$</td>
<td>$L = 150$</td>
<td>0.007, 41.4, +∞</td>
<td>0.013, 39.0, +∞</td>
<td>0.020, 38.4, +∞</td>
<td>0.027, 38.1, +∞</td>
</tr>
<tr>
<td></td>
<td>$L = 125$</td>
<td>0.008, 41.4, +∞</td>
<td>0.016, 39.0, +∞</td>
<td>0.024, 38.5, +∞</td>
<td>0.032, 38.1, +∞</td>
</tr>
<tr>
<td></td>
<td>$L = 100$</td>
<td>0.010, 41.0, +∞</td>
<td>0.020, 39.0, +∞</td>
<td>0.030, 38.5, 67.8</td>
<td>0.040, 38.1, 67.9</td>
</tr>
</tbody>
</table>

TABLE III
EMBEDDING RATE $R$, PSNR IN DIRECTLY DECRYPTED IMAGES (dB) AND PSNR IN RECOVERED IMAGES (dB) WITH DIFFERENT PARAMETERS FOR TEST IMAGE MAN

<table>
<thead>
<tr>
<th></th>
<th>$S = 1$</th>
<th>$S = 2$</th>
<th>$S = 3$</th>
<th>$S = 4$</th>
<th>$S = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M = 1$</td>
<td>$L = 2000$</td>
<td>0.0005, 53.8, +∞</td>
<td>0.0010, 52.2, +∞</td>
<td>0.0015, 51.7, +∞</td>
<td>0.0020, 51.4, +∞</td>
</tr>
<tr>
<td></td>
<td>$L = 1500$</td>
<td>0.0007, 54.2, +∞</td>
<td>0.0013, 52.1, +∞</td>
<td>0.0020, 51.6, +∞</td>
<td>0.0027, 51.3, +∞</td>
</tr>
<tr>
<td></td>
<td>$L = 1000$</td>
<td>0.00010, 54.1, +∞</td>
<td>0.0020, 52.2, 75.0</td>
<td>0.0030, 51.7, 75.1</td>
<td>0.0040, 51.4, 72.2</td>
</tr>
<tr>
<td>$M = 2$</td>
<td>$L = 400$</td>
<td>0.0025, 47.2, +∞</td>
<td>0.0050, 45.1, +∞</td>
<td>0.0075, 44.6, +∞</td>
<td>0.010, 44.3, +∞</td>
</tr>
<tr>
<td></td>
<td>$L = 320$</td>
<td>0.0031, 47.3, +∞</td>
<td>0.0063, 45.2, +∞</td>
<td>0.0094, 44.6, 73.7</td>
<td>0.0125, 44.4, 70.2</td>
</tr>
<tr>
<td></td>
<td>$L = 250$</td>
<td>0.004, 47.1, +∞</td>
<td>0.008, 45.0, 71.6</td>
<td>0.012, 44.5, 66.6</td>
<td>0.016, 44.3, 64.4</td>
</tr>
<tr>
<td>$M = 3$</td>
<td>$L = 200$</td>
<td>0.005, 41.2, +∞</td>
<td>0.010, 39.0, +∞</td>
<td>0.015, 38.2, +∞</td>
<td>0.020, 38.0, +∞</td>
</tr>
<tr>
<td></td>
<td>$L = 150$</td>
<td>0.007, 41.4, +∞</td>
<td>0.013, 38.8, 69.3</td>
<td>0.020, 38.2, +∞</td>
<td>0.027, 38.1, 71.0</td>
</tr>
<tr>
<td></td>
<td>$L = 120$</td>
<td>0.008, 41.1, +∞</td>
<td>0.017, 39.0, +∞</td>
<td>0.025, 38.4, 66.4</td>
<td>0.033, 38.0, 66.6</td>
</tr>
</tbody>
</table>

Fig. 4. Rate-PSNR comparison between the proposed scheme and the method in [12].

Fig. 5. Rate-Watson metric comparison between the proposed scheme and the method in [12].
people. When meeting the perfect recovery condition, the proposed scheme has an average 203% gain of embedded data amount with same PSNR value in directly decrypted image, or an average gain of 8.7 dB of PSNR value in directly decrypted image with same embedded data amount.

IV. CONCLUSION

In this paper, a novel scheme for separable usable data hiding in encrypted image is proposed, which consists of image encryption, data embedding and data-extraction/image-recovery phases. In the first phase, the content owner encrypts the original uncompressed image using an encryption key. Although a data-hider does not know the original content, he can compress the least significant bits of the encrypted image using a data-hiding key to create a sparse space to accommodate the additional data. With an encrypted image containing additional data, the receiver may extract the additional data using only the data-hiding key, or obtain an image similar to the original one using only the encryption key. When the receiver has both of the keys, he can extract the additional data and recover the original content without any error by exploiting the spatial correlation in natural image if the amount of additional data is not too large. If the lossless compression method in [1] or [2] is used for the encrypted image containing embedded data, the additional data can be still extracted and the original content can be also recovered since the lossless compression does not change the content of the encrypted image containing embedded data. However, the lossy compression method in [3] compatible with encrypted images generated by pixel permutation is not suitable here since the encryption is performed by bit-XOR operation. In the future, a comprehensive combination of image encryption and data hiding compatible with lossy compression deserves further investigation.

REFERENCES


