Journal of Computer Applications ISSN: 0974 – 1925,
Volume-5, Issue EICA2012-4, February 10, 2012

Iterative Reconstruction of an Encrypted Image Using LOCO-I Algorithm

Santhana Bharathi R
Dept. of Electronics and Communication Engineering
Paavai College of Engineering
Namakkal, India
E-mail: bharathi.tuti@gmail.com

Vijaya Baskar B
Dept. of Electronics and Communication Engineering
Arunai Engineering College
Thiruvannamalai, Tamil Nadu, India
E-mail: vijaybaskar.me@gmail.com

Abstract - A pseudorandom permutation is used to encrypt an original image, and the encrypted data are efficiently compressed by discarding the excessively rough and fine information of coefficients generated from orthogonal transform. After receiving the compressed data, with the aid of spatial correlation in natural image, a receiver can re-construct the principal content of the original image by iteratively updating the values of coefficients. This way, the higher the compression ratio and the smoother the original image, the better the quality of the reconstructed image.

Key words: *Image compression, image encryption, image reconstruction.*

I. INTRODUCTION

In recent years, compression of encrypted data has attracted considerable research interest. The traditional way of securely and efficiently transmitting redundant data is to first compress the data to reduce the redundancy, and then to encrypt the compressed data to mask its meaning. At the receiver side, the decryption and decompression operations are orderly performed to recover the original data. However, in some application scenarios, a sender needs to transmit some data to a receiver and hopes to keep the information confidential to a network operator who provides the channel resource for the transmission. That means the sender should encrypt the original data and the net- work provider may tend to compress the encrypted data without any knowledge of the cryptographic key and the original data. At receiver side, a decoder integrating decompression and decryption functions will be used to reconstruct the original data. Several techniques for compressing / decompressing encrypted data have been developed. Based on the theory of source coding with side information at the decoder, the performance of compressing encrypted data may be as good as that of compressing nonencrypted data in theory. Two practical approaches to lossless compression of encrypted black and white images and to lossy compression of encrypted.

Gaussian sequence are also presented. In the former approach, the original binary image is encrypted by adding a pseudorandom string, and the encrypted data are com- pressed by finding the syndromes with respect to low-density parity-check (LDPC) channel code [2]. In the latter one, the original data is encrypted by adding an i.i.d. Gaussian sequence, and the encrypted data are quantized and compressed as the syndromes of trellis code. The compression of encrypted data for both memoryless sources and sources with hidden Markov correlation using LDPC codes is also studied [3]. By employing LDPC codes into various bit-planes and exploiting the spatial and cross-plane correlation among pixels, a few methods for lossless compression of encrypted gray and color images are introduced in [4].

In [5], the encrypted image is decomposed in a progressive manner, and the most significant bits in high levels are compressed using rate-compatible punctured turbo codes. The decoder can observe a low-resolution version of the image, study local statistics based on it, and use the statistics to obtain the content in high levels. Furthermore, by developing statistical models for source data and extending these models to video, [6] presents some algorithms for compressing encrypted data and demonstrate blind compression of encrypted video. In [7], a compressive sensing technique is introduced to achieve lossy compression of encrypted image data, and a basis pursuit algorithm is appropriately modified to enable joint decompression and decryption. The signal processing in the encryption domain using homomorphic calculation is also discussed in [8] and [9].

In most of the above-mentioned schemes for compressing encrypted image, the syndrome of channel code is exploited to generate the compressed data in lossless manner. The network provider may remove the redundant and trivial data from the encrypted image, and a receiver can retrieve the principal content of the original image using an iterative procedure. The compression ratio and the quality of the reconstructed image are dependent on the values of compression parameters. Generally, the higher the compression ratio and the smoother the original image, the better the quality of the reconstructed image. Compared with the previous lossless encrypted-image compression approaches, with a cost of slight degradation of encryption security and reconstruction quality, the proposed scheme can significantly improve the compression efficiency.

II. COMPRESSION AND DECOMPRESSION OF ENCRYPTED IMAGE

In the proposed scheme, a pseudorandom permutation is used to encrypt an original image. Then, the encrypted data can be efficiently compressed by discarding the excessively rough and fine information of coefficients in the transform domain. When having the compressed data and the permutation way, with the aid of spatial correlation in natural image, the receiver can reconstruct the principal content of the original image by iteratively updating the values of the coefficients.

A. Image Encryption

Assume the original image is in uncompressed format and each pixel with a gray value falling into [0, 255] is represented by 8 bits. Denote the numbers of the rows and the columns in the original image as N_1 and N_2, and the number of all pixels as N (N1 × N_2) then, the amount of bits of the original image is 8 · N. For image encryption, the data sender pseudorandomly permutes the N pixels and the permutation way is determined by a

secret key. The permuted pixel-sequence is viewed as the encrypted data.

A number of permutation-based image encryption methods can be used here [10], [11]. Since only the pixel positions are permuted and the pixel values are not masked in the encryption phase, an attacker without knowledge of the secret key can know the original histogram from an encrypted image. However, the number of possible permutation ways is N! so that it is unpractical to perform a brute force search when N is fairly large. That means the attacker cannot recover the original content from the encrypted image with ordinary size and fluctuation. Although there is a leakage of statistical information, the permutation-based encryption can be used in most scenarios without a requirement of perfect secrecy.

B. Compression of Encrypted Image

In the compression procedure, a majority of pixels are converted to a series of coefficients using an orthogonal transform, and then the excessively rough and fine information in the coefficients is removed, leading to a reduced data amount. The detailed procedure is as follows.

1) When having the permuted pixel sequence, the network provider divides it into two parts: the first part made of $\alpha \cdot N$ pixels and the second one containing the rest of the $(1 \cdot \alpha)$N pixels. Denote the pixels in the first part as $p_1, p_2, \cdots, p_{\alpha \cdot N}$and the pixels in the second part as $q_1, q, \cdots, q_{(1-\alpha)\cdot N}$. The value of is within (0,1) and will be discussed in Section III. Here, the data in the first part will be reserved while the data redundancy in the second part will be reduced. We call the pixels in the first part rigid pixels and the pixels in the second part elastic pixels.

2) Perform an orthogonal transform in the elastic pixels to calculate the coefficients $Q_1, Q_2, \cdots, Q_{(1-\alpha)\cdot N}$.

$$[Q_1, Q_2, \cdots, Q_{(1-\alpha)\cdot N} = [q_1, q_2, \cdots, q_{(1-\alpha)\cdot N}1] \cdot H \qquad (1)$$

Here, H is a public orthogonal matrix with a size of $(1\alpha) \cdot N \times (1 - \alpha) \cdot N$, and it can be generated from orthogonalizing a random matrix.

3) For each coefficient, calculate

$$s_k = \mathrm{mod}\left[\mathrm{round}\left(\frac{Q_k}{\frac{\Delta}{M}}\right), M\right], \quad = 1, 2, \ldots, (1 \quad \alpha) \cdot N \qquad (2)$$

Where Δ and M are system parameters and will be discussed in Section III. The round operation returns the nearest integer and the mod operation gets the remainder. By (2), $Q_{k\ is}$ converted into an integer s_k within [0, M 1]. With a small M, the data amount for representing the elastic pixels is reduced. As Q_k can be rewritten in the following manner

$$Q_k = r_k \cdot \Delta + s_k \cdot \frac{\Delta}{M} \quad (3)$$

Where r_k and s_k are integers and $0 \leq s_k \leq M\ 1;\ \frac{\Delta}{2M} \leq t_k \leq \frac{\Delta}{2M}$ (4)

It can be seen that the rough information r_k and the fine information t_k are discarded, while only the information on the medium level s_k remains. Note that the rough information r_k will be retived retrieved by an iterative image recon- struction procedure, and the loss of the fine information t_k cannot seriously affect the quality of the reconstructed image.

4) Since s_k are within[0, M-1; we can regard them as a set of digits in a notational system with a base M. Segment the set of into many pieces with L_1 digits and calculate the decimal value of each digit piece. Then, convert each decimal value into L_2 bits in a binary notational system, where

$$L_2 = \lceil L_1 \cdot log_2 M \rceil \quad (5)$$

For example, the digit set {23 11 41} in a 5-ary nota tional system can be expressed as a binary sequence (0110100110 10101) where L_1=2 and L_2=5. From (5), there must be

$$L_1 \cdot log_2 M \leq L_2 < L_1 \cdot log_2 M + 1 \quad (6)$$

Then,

$$log_2 M \leq \frac{L_2}{L_1} < log_2 M + \frac{1}{L_1} \quad (7)$$

With a large L_1, $\frac{L_2}{L_1} \approx log_2 M$ (8)

So, the total length of bits generated from all pieces of s_k is

$$L_1 = (1-\alpha) \cdot N\ \frac{L_2}{L_1} \approx (1-\alpha) \cdot N \cdot log_2 M \quad (9)$$

In practical implementation, we use $L_1 = 40$.

5) Collect the data of rigid pixels, the bits generated from all pieces of s_k, and the values of parameters including N_1, N_2, $\propto$, Δ, M amd L_1 to produce the compressed data of encrypted image. Since the data amount of parameters is small, the compression ratio R, a ratio between the amounts of the compressed data and the original image data, is approximately

$$R = \frac{\text{[illegible]}}{8N^2} = \text{[illegible]} \quad (10)$$

C. Image Reconstruction

With the compressed data and the secret key, a receiver can perform the following steps to reconstruct the principal content of the original image.

1) Decompose the compressed data and obtain the gray values of rigid pixels, the values of all s_k, and the values of M and L_1, the L_2, and and then get the values of s_k by converting binary blocks with L_2 bits into digit pieces in an M-ary notational system.

2) According to the secret key, the receiver can retrieve the positions of rigid pixels. That means the original gray values at the positions, which distribute over the entire image, can be exactly recovered.

3) For the pixels at other positions, i.e., the elastic pixels, their values are firstly estimated as the values of rigid pixels nearest to them. That means, for each elastic pixel, we find the nearest rigid pixel and regard the value of the rigid pixel as the estimated value of the elastic pixel. If there are several nearest rigid pixels with the same distance, re- gard their average value as the estimated value of the elastic pixel. Because of spatial correlation in the natural image, the estimated values are similar to the corresponding orig- inal values. In the following, the estimation will be itera- tively updated by exploiting the information o f s_k.

4) Rearrange the estimated values of elastic pixels using the same permutation way, and denote them as $q'_1, q'_2, \ldots, q'_{(1-\alpha)\cdot N}$.
Calculate the coefficients

$$[Q'_1, Q'_2, \ldots, Q'_{(1-\alpha)\cdot N}] = [q'_1, q'_2, \ldots, q'_{(1-\alpha)\cdot N}] \cdot H \quad (11)$$

and

$$[d_k = mod\left(\frac{Q'_k}{\frac{\Delta}{M}}, M\right) - s_k, \text{k} = 1,2,..., (1-\propto) \cdot N \quad (12)$$

Modify the coefficients to the closest values consistent with the corresponding s_k

$$Q_k'' = \begin{cases} \left(\left\lfloor \frac{Q'_k}{\Delta} \right\rfloor + \right) \cdot \Delta + s_k \cdot \frac{\Delta}{M}, & d_k \geq \frac{M}{2} \\ \left\lfloor \frac{Q'_k}{\Delta} \right\rfloor \cdot \Delta + s_k \cdot \frac{\Delta}{M}, & \frac{-M}{2} < \frac{M}{2} \\ \left(\left\lfloor \frac{Q'_k}{\Delta} \right\rfloor \right. \cdot \Delta + s_k \cdot \frac{\Delta}{M}, & d_k < \frac{-M}{2} \end{cases}$$

(13)

For example with Q'$_k$ = 42. 6, Δ = 60, and M = 65 , if s_k = 0, d_k is 4.26 according to (12) , so we should modify the value of Q'$_k$ to 60. If $s_k = 3, d_k = 1.26$ and we should modify the value of Q'$_k$ to 30. Then, perfom an inverse transform.

$$[q''_1, q''_2, \ldots, q''_{(1-\alpha)N}] = [Q''_1, Q''_2, \ldots, Q''_{(1-\alpha)N}] \cdot H^{-1}$$

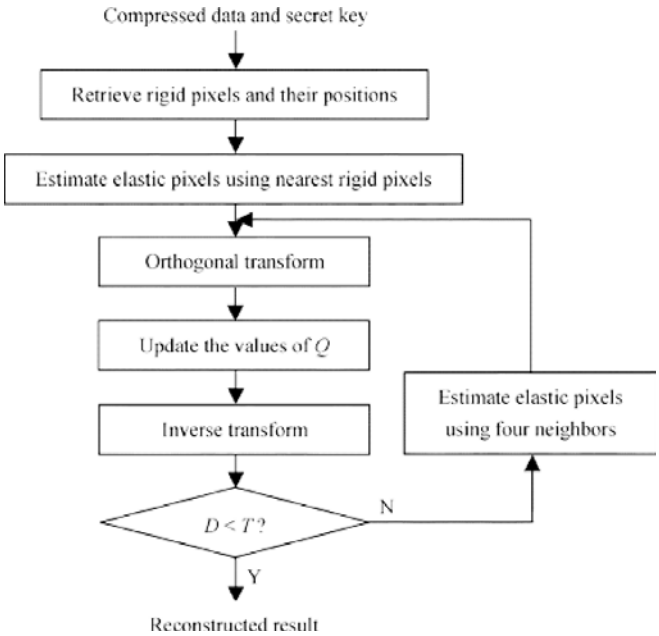

Figure 1: Image Reconstruction Procedure

5) Calculate the average energy of difference between the two versions of elastic pixels

$$D = \frac{1}{(1-\alpha)\cdot N} \cdot \sum_{k=1}^{(1-\alpha)\cdot N} (q''_k q'_k)^2$$

If D is not less than a given threshold T, for each elastic Pixel, regard the average value of its four neighbor pixels as its new estimated value and go to Step 4. Otherwise, terminate the iteration and output the image made up of the rigid pixels and the final version of elastic pixels as a reconstructed result.

Figure 1 sketches the image reconstruction procedure. While the values of rigid pixels are used to give an initial estimation of elastic pixels, the values of s_k provide more detailed infor- mation to produce a final reconstructed result with satisfactory quality. Since the rigid pixels may be not evenly distributed in the image, the estimation of Step 3 in an area with crowded rigid pixels will be more precise than that of an area with sparse rigid pixels. Also, the estimation in the plain area is better than that in the texture/edge area. As long as we have an approximately estimated version as an initialization, the iterative procedure can produce a satisfactory reconstructed result. By the or- thogonal transform, the estimation error of elastic pixels is scattered over all the coefficients. With the iterative update at each coefficient, the reconstructed elastic pixels approach their original values progressively.

Since the coefficients are generated from all elastic pixels, the errors in a final reconstructed result are distributed over the image with an approximately uniform manner. That means the qualities of different parts in the final reconstructed image will be almost the same. If the orthogonal transform is absent, it is hard to exactly reconstruct the elastic pixels in texture/edge areas by updating since their estimated values generated from the neighbors may be very different from their original values. In Step 5, the threshold T recommended as 0.05 to ensure that the last two versions of elastic pixels are

Figure 2: (a) Original image Lena, (b) its Encrypted Version, (c) the Medium Reconstructed image from compressed data with PSNR 27.1 db, and (d) the final reconstructed image with PSNR 39.6 db.

Table 1: Compression Ratio and PSNR (DB) in Reconstructed Image With Different Parameters For Test Image Lena

		$\alpha = 0.15$	$\alpha = 0.10$	$\alpha = 0.07$
$M = 8$	$\Delta = 80$	0.47, 39.6	0.44, 39.4	0.42, 39.2
$M = 8$	$\Delta = 60$	0.47, 42.1	0.44, 41.9	0.42, 41.6
$M = 8$	$\Delta = 50$	0.47, 43.7	0.44, 43.5	0.42, 43.2
$M = 6$	$\Delta = 80$	0.42, 37.1	0.39, 36.9	0.37, 36.7
$M = 6$	$\Delta = 60$	0.42, 39.6	0.39, 39.4	0.37, 39.2
$M = 6$	$\Delta = 50$	0.42, 41.2	0.39, 40.9	0.37, 40.7
$M = 4$	$\Delta = 80$	0.36, 33.6	0.33, 33.4	0.30, 33.2
$M = 4$	$\Delta = 60$	0.36, 36.1	0.33, 35.9	0.30, 35.7
$M = 4$	$\Delta = 50$	0.36, 37.7	0.33, 37.4	0.30, 37.3

close enough and the update does not improve the reconstructed result further.

III. EXPERIMENTAL RESULTS AND DISCUSSION

The test image Lena sized 512 x 512 shown in Fig. 2 (a) was used as the original in the experiment. After pixel permutation, the encrypted data of the image were produced. For showing their disorder, the encrypted pixel sequence is rearranged as a matrix with size of 512 x 512 and given in Fig. 2(b). Then, we compressed the encrypted data with $\propto = 0.15$, $\Delta = 60$ and M = 6. In this case, thecompression ratio R = 0.42. With the compressed data, the receiver can retrieve the original content by using the image reconstruction procedure. Fig. 2(c) shows a medium reconstructed image generated by Steps 1–3, in which all rigid pixels are recovered and the elastic pixels are estimated as the values of their nearest rigid pixels. The value of PSNR in the medium reconstructed image is 27.1 dB, and the quality of the texture/edge area is worse than that of the plain area.

When finishing the iterative update in Steps 4 and 5, a final decom- pressed image shown in Fig. 2(d) was obtained, and PSNR is 39.6 dB. It can be seen that the iterative procedure significantly improves the reconstruction quality.

Table 2: Compression Ratio and PSNR (DB) in Reconstructed Image with Different Parameters for Test Image Man

		$\alpha = 0.15$	$\alpha = 0.10$	$\alpha = 0.07$
$M = 8$	$\Delta = 80$	0.47, 39.6	0.44, 39.4	0.42, 39.1
$M = 8$	$\Delta = 60$	0.47, 42.0	0.44, 41.7	0.42, 41.4
$M = 8$	$\Delta = 50$	0.47, 42.4	0.44, 42.0	0.42, 41.7
$M = 6$	$\Delta = 80$	0.42, 37.1	0.39, 36.9	0.37, 36.7
$M = 6$	$\Delta = 60$	0.42, 39.6	0.39, 39.3	0.37, 39.0
$M = 6$	$\Delta = 50$	0.42, 40.1	0.39, 39.9	0.37, 39.5
$M = 4$	$\Delta = 80$	0.36, 33.6	0.33, 33.3	0.30, 33.2
$M = 4$	$\Delta = 60$	0.36, 36.0	0.33, 35.7	0.30, 35.5
$M = 4$	$\Delta = 50$	0.36, 36.7	0.33, 36.3	0.30, 36.1

Table 3:Iteration Numbers for Convergence with Test Images Lena and Man

		$\alpha = 0.15$	$\alpha = 0.10$	$\alpha = 0.07$
$M = 8$	$\Delta = 80$	2, 2	2, 3	3, 3
$M = 8$	$\Delta = 60$	2, 4	3, 5	4, 5
$M = 8$	$\Delta = 50$	4, 12	4, 13	5, 17
$M = 6$	$\Delta = 80$	2, 2	2, 3	3, 3
$M = 6$	$\Delta = 60$	3, 4	3, 5	4, 6
$M = 6$	$\Delta = 50$	4, 15	4, 10	5, 17
$M = 4$	$\Delta = 80$	2, 3	2, 3	3, 4
$M = 4$	$\Delta = 60$	3, 6	3, 9	4, 9
$M = 4$	$\Delta = 50$	4, 26	5, 22	6, 29

Tables I and II list the values of compression ratios and PSNR in reconstructed images when different $\propto$, Δ, and M were used for images Lena and Man. The iteration numbers for content re- construction when D is less than the given threshold or does not decrease with more iterative rounds are also given in Table III. decrease with more iterative rounds are also given in Table III. All the encryption, compression, and reconstruction procedures can be finished in several seconds by a personal computer. As in (10), the compression ratio is determined by $\propto$ and M, and the smaller $\propto$ and M correspond to a lower R. On the other hand, the larger the values of $\propto$ and M, the iteration numbers are usually smaller and the qualities of reconstructed images are better since more rigid pixels and more detailed s_k can be used to retrieve the values of elastic pixels. The compression ratio is independent of the value of Δ, and, generally speaking, a smaller Δ, can result in a better reconstructed image since the receiver can exploit more precise information for image recon- struction. However, more iterations are made for getting a final reconstructed result when using a smaller Δ, and, if the value of Δ, is too small, the updating procedure is not convergent. For example with $\propto$ = 0.07, the updating procedure for reconstructing Lena does not converge when Δ is below 35, while the updating Δ is below 45.

The convergence is also dependent on the accuracy of estimating elastic pixels from their neighbors. The better the estimation accuracy, it is easier to get convergence. Since there is more texture/edge content in Man than Lena, the estimation of elastic pixels is more different from the original values, so that the convergence of the updating procedure for Man requires more iteration and larger Δ For making a trade-off between the iterative convergence and the reconstruction quality of most images, we recommended Δ.

Fig. 3 shows PSNR of reconstructed images with respect to compression ratios when four test images Lena, Man, Couple, and Lake sized 512 x 512 were used as the original. Here Δ = 60,$\propto$ = 0.07 with M = 3,4, .., 8, and $\propto$ = 0.10 with M = 8,9,..,11 were used respectively. The smoother the original image, the better is the quality of reconstructed image.

With lossless compression methods for encrypted image [4]–[6], the different bit-planes are always compressed re- spectively, but it is difficult to reduce the data amount in low bit-planes. While only the first two most significant bits (MSB) are compressible by employing a 2-D Markov model [6], the compression can be executed in the first four MSB based on interpolation operation [5].

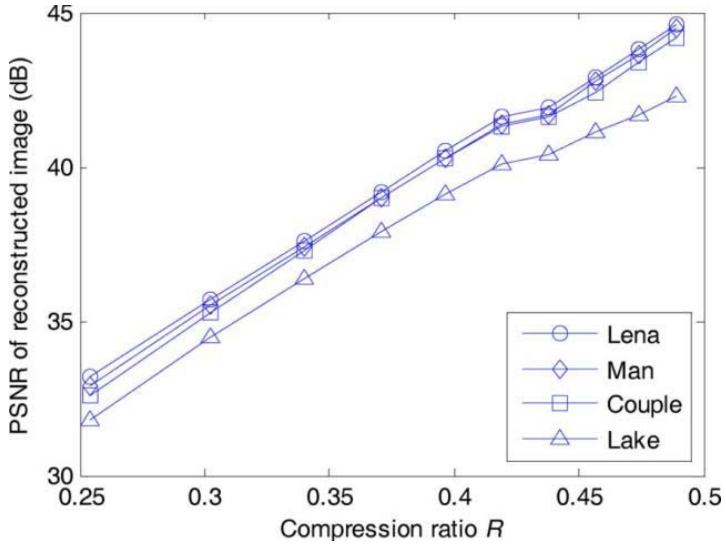

Figure 3: PSNR of Reconstructed Images with Respect to Compression Ratios

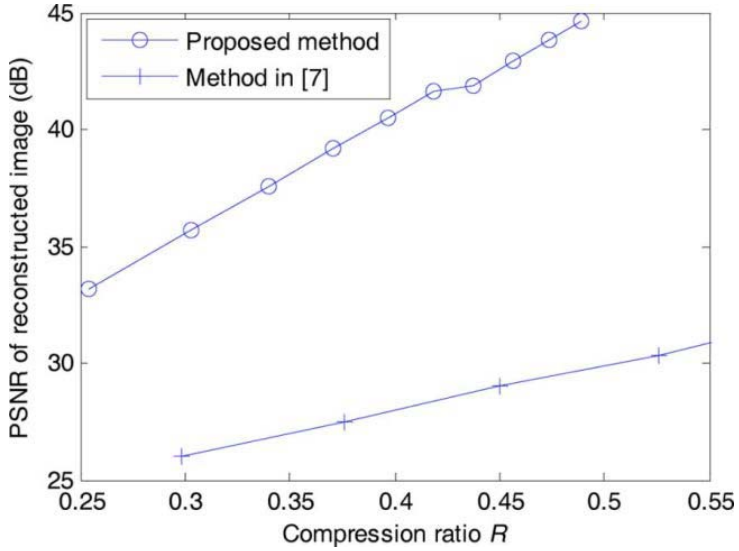

Figure 4: Comparison Between the Proposed Method and Method in [7] when Lena is used

Although the receiver can reconstruct the original image without any error (PSNR is infinite), the compression ratio is higher and more channel resource is required for transmitting the compressed data. For example, when using the method in [5] to compress an encrypted Lena, the compression ratio was 61%. With the scheme proposed in this paper, the quality of recompressed Lena was still satisfactory even though the compression ratio R was lowered to 30%. That means the proposed scheme is more suitable for a low bandwidth channel. There are few numerical results reported in the literature for lossy compression of encrypted grayscale images. Fig. 4 compares the proposed scheme and the method in [7], which introduces compressing sensing to perform lossy compression of encrypted image, when using the test image Lena. It can be seen that the proposed scheme significantly outperforms the method in [7].

IV. CONCLUSION

This work proposed a novel idea for compressing and encrypted image and designed a practical scheme made up of image encryption, lossy compression, and iterative reconstruction. The original image is encrypted by pseudorandom permutation, and then compressed by discarding the excessively rough and fine information of coefficients in the transform do- main. When having the compressed data and the permutation way, an iterative updating procedure is used to retrieve the values of coefficients by exploiting spatial correlation in natural image, leading to a reconstruction of original principal content. The compression ratio and the quality of reconstructed image vary with different values of compression parameters. In general, the higher the compression ratio and the smoother the original image, the better the quality of the reconstructed image. In the encryption phase of the proposed system, only the pixel positions are shuffled and the pixel values are not masked. With the values of elastic pixels, the coefficients can be generated to produce the compressed data. On the other hand, the security of encryption used here is weaker than that of standard stream cipher, which can be cooperative with previous lossless compression techniques, since the distribution of pixel-values may be revealed from an encrypted image. The lossy compression of image encrypted by more secure methods will be studied in the future.

REFERENCES

1. Xinpeng Zhang,"Lossy compression and iterative reconstruction for encrypted image "IEEE Trans.Inf.Forensic Security,vol6,No 1,March2011
2. R. G. Gallager, "Low Density Parity Check Codes," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, MA, 1963.
3. D. Schonberg, S. C. Draper, and K. Ramchandran, "On blind compres- sion of encrypted correlated data approaching the source entropy rate,"in *Proc. 43rd Annu. Allerton Conf.*, Allerton, IL, 2005.
4. R. Lazzeretti and M. Barni, "Lossless compression of encrypted grey-level and color images," in *Proc.16th Eur. Signal Processing Conf. (EUSIPCO 2008)*, Lausanne, Switzerland, Aug. 2008 [Online]. Available:http: //www.eurasip.org/Proceedings/ Eusipco /Eusipco2008/papers/1569105134.pdf
5. 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.
6. 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.
7. Kumar.A and A. Makur, "Lossy compression of encrypted image by compressing sensing technique," in *Proc. IEEE Region 10 Conf.(TENCON 2009)*, 2009, pp. 1–6.
8. T. Bianchi, A. Piva, and M. Barni, "Composite signal representation for fast and storage-efficient processing of encrypted signals," *IEEET rans. Inf. Forensics Security*, vol. 5, no. 1, pp. 180–187, Mar. 2010.
9. T. Bianchi, A. Piva, and M. Barni, "On the implementation of the discrete fourier transform in the encrypted domain," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 1, pp. 86–97, Mar. 2009.
10. J.-C. Yen and J.-I. Guo, "Efficient hierarchical chaotic image encryption algorithm and its VLSI realization," *Proc. Inst. Elect. Eng., Vis.Image Signal Process.*, vol. 147, no. 2, pp. 167–175, 2000.
11. N. Bourbakis and C. Alexopoulos, "Picture data encryption usingSCAN patterns," *Pattern Recognit.*, vol. 25, no. 6, pp. 567–581, 1992.