Akash Raj / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.comVol. 3, Issue 2, March -April 2013, pp.444-450

# High Resolution Image Encryption & Reconstruction Using Scalable Codes

**Akash Raj**
Department Of Ece, M.E Embedded System, Sathyabama University, Chennai-119,
Tamil Nadu, India

## ABSTRACT

**This paper proposes a novel scheme of scalable coding for encrypted gray images. Although there have been a lot of works on scalable coding of unencrypted images/videos the scalable coding of encrypted data has not been reported. In the encryption phase of the proposed scheme, the pixel values are completely concealed so that an attacker cannot obtain any statistical information of an original image. Then, the encrypted data are decomposed into several parts, and each part is compressed as a bit stream. At the receiver side with the cryptographic key, the principal content with higher resolution can be reconstructed when more bit streams are received**.

**Keywords-**
Bitstreams,Cryptography,Downsampling, Hadamard transform, Imagecompression ,Image encryption,QuantizationScalable coding.

## I. INTRODUCTION

In recent years, encrypted signal processing has attracted considerable research interests. The discrete Fourier transform and adaptive filtering can be implemented in the encrypted domain based on the holomorphic properties of a cryptosystem, and a composite signal representation method can be used to reduce the size of encrypted data and computation complexity. In joint encryption and data hiding, a part of significant data of a plain signal is encrypted for content protection, and the remaining data are used to carry buyer–seller protocols, the fingerprint data are embedded into an encrypted version of digital multimedia to ensure that the seller cannot know the buyer's watermarked version while the buyer cannot obtain the original product. A number of works on compressing encrypted images have been also presented. When a sender encrypts an original image for privacy protection, a channel provider without the knowledge of a cryptographic key and original content may tend to reduce the data amount due to the limited channel resource

### A. IMAGE ENCRYPTION

The original image is in an uncompressed format and that the pixel values are within [0, 255], and denote the numbers of rows and columns as $N_1$ and $N_2$ and the pixel number as ($N=N_1$ X $N_2$).

Therefore, the bit amount of the original image is 8N. The content owner generates a pseudorandom bit sequence with a length of 8N. Here, we assume the content owner and the decoder has the same pseudorandom number generator (PRNG) and a shared secret key used as the seed of the PRNG. Then, the content owner divides the pseudorandom bit sequence into N pieces, each of which containing 8 bits, and converts each piece as an integer number within [0, 255]. An encrypted image is produced by a one-by-one addition modulo 256 as follows:

$$g^{(0)}(i,j)=\mathrm{mod}\,[p(i,j)+e(i,j),256],\quad 1\le i\le N_1, 1\le i\le N_2$$

Where $p(i,j)$ represents the gray values of pixels at positions $(i,j)$, $e(i,j)$ represents the pseudorandom numbers within [0, 255] generated by the PRNG, and $g^{(0)}(i,j)$ represents the encrypted pixel values. Clearly, the encrypted pixel values $g^{(0)}(i,j)$ are pseudorandom numbers since $e(i,j)$values are pseudorandom numbers. It is well known that there is no probability polynomial time (PPT) algorithm to distinguish a pseudorandom number sequence and a random number sequence until now. Therefore, any PPT adversary cannot distinguish an encrypted pixel sequence and a random number sequence. That is to say, the image encryption algorithm that we have proposed is semantically secure against any PPT adversary.

### B. ENCRYPTED IMAGE ENCODING

Although an encoder does not know the secret key and the original content, he can still compress the encrypted data as a set of bitstreams. The detailed encoding procedure is as follows. First, the encoder decomposes the encrypted image into a series of subimages and data sets with a multiple-resolution construction. The subimage at the $(t+1)$th level $G^{(t+1)}$ is generated by downsampling the subimage at the $t$ th level as follows:

$$g^{(t+1)}(i,j)=g^{(t)}(2i,2j),$$
$$t=0,1,\cdots,T-1$$

Where $G^{(0)}$ just the encrypted image and T is is the number of decomposition levels. In addition, the encrypted pixels that belongs to $G^{(t+1)}$ but do not belong to form data set $Q^{(t+1)}$ as follows:

$$Q^{(t+1)}=\{g^{(t)}(i,j)|\mathrm{mod}(i,2)=1 or \mathrm{mod}(j,2)=1\},$$
$$t=0,1\cdots,T-1.$$

That means each $G^{(t)}$ is decomposed into $G^{(t+1)}$ and $Q^{(t+1)}$, and the data amount of $Q^{(t+1)}$ is three times of that of $G^{(t+1)}$. After the multiple-level decomposition, the encrypted image is reorganized as $G^{(T)}, Q^{(T)}, Q^{(T-1)}, \cdots$ and $Q^{(t)}$

For the subimage $G^{(T)}$, the encoder quantizes each value using a step Δ as follows:

$$b(i,j)=\left\lfloor \frac{g^{(T)}(i,j)}{\Delta} \right\rfloor$$

Where the operator $\lfloor \bullet \rfloor$ takes an integer toward minus infinity and

$$\Delta=256/M$$

Here, M is an integer shared by the encoder and the decoder, and its value will be discussed later. Clearly

$$0\le b(i,j)\le M-1$$

Then, the data of b (i,j) are converted into a bitstream, which is denoted as BG. The bit amount of BG is

$$N_{BG}=\frac{N}{4^T}.\log_2 M.$$

For each data set $Q^{(t)}(t=1,2,\cdots T)$ the encoder permutes and divides encrypted pixels in it into $K^{(t)}$ groups, each of which containing $L^{(t)}$ pixels $(K^{(t)} x L^{(t)}=3N/4^t)$. In this way, the $L^{(t)}$ pixels in the same group scatter in the entire image. The permutation way is shared by the encoder and the decoder, and the values of $L^{(t)}$ will be discussed later. Denote the encrypted pixels of the Kth group as $q_k^{(t)}(1), q_k^{(t)}(2), \cdots, q_k^{(t)}(L^{(t)})(1\le k\le K(t))$, and perform the Hadamard transform in each group as follows:

$$\begin{bmatrix} C_k^{(t)}(1) \\ C_k^{(t)}(2) \\ \vdots \\ C_k^{(t)}(L^{(t)}) \end{bmatrix} = H \bullet \begin{bmatrix} q_k^{(t)}(1) \\ q_k^{(t)}(2) \\ \vdots \\ q_k^{(t)}(L^{(t)}) \end{bmatrix}$$

Where **H** is a $L^{(t)}$ x $L^{(t)}$ Hadamard matrix made up of +1 or -1. That implies the matrix **H** meets

**H'*H = H*H' = $L^{(t)}$ * I**

Where $H^{(t)}$ is a transpose of H,I is an $L^{(t)}$ x $L^{(t)}$ identity matrix, and $L^{(t)}$ must be a multiple of 4.

For each coefficient $C_k^{(t)}(l)$, calculate

$$C_k^{(t)}(l)=\left\lfloor \frac{\mathrm{mod}[C_k^{(t)}(l),256]}{256/M^{(t)}} \right\rfloor$$

$$1\le k\le K^{(t)}, 1\le k=l\le L^{(t)}$$

Where

$$M^{(t)}=\mathrm{round}(M/\sqrt{L^{(t)}})$$

and round (.) finds the nearest integer. The remainder of $C_k^{(t)}(l)$ modulo 256 is quantized as integer $C_k^{(t)}(l)$, $L^{(t)}$, and the quantization steps are approximately proportional to square roots of $L^{(t)}$. Then, $C_k^{(t)}(l)$ at different levels are converted into bitstreams, which are denoted as $BS^{(t)}$. Since

$$0\le C_k^{(t)}(l)\le M^{(t)}-1$$

and the number of $C_k^{(t)}(l)$ at the th level is $3N/4^t$ the bit amount of $BS^{(t)}$ is

$$N^{(t)}=\frac{3\bullet N\bullet \log_2 M^{(t)}}{4^t}, \qquad t=1,2,\cdots,T$$

The encoder transmits the bitstreams with an order of $\{BG, BS^{(T)}, BS^{(T-1)}, \cdots, BS^{(1)}\}$. If the channel bandwidth is limited, the latter bitstreams may be abandoned. A higher resolution image can be reconstructed when more bitstreams are obtained at the receiver side. Here, the total compression ratio $R_C$, which is a ratio between the amount of the encoded data and the encrypted image data, is

$$R_C=\frac{N_{BG}}{8N}+\frac{1}{8N}\sum_{t=1}^{T}N^{(t)}=\frac{\log_2 M}{8\bullet 4^t}+\frac{3}{8}\bullet\sum_{t=1}^{T}\frac{\log_2 M^{(t)}}{4^t}$$

In the figure below, fig (a) is the original image, fig (b) is encrypted image of size 512x512.after encryption I compressed the imge .i.e fig (c) is 256x256 size image. Fig (d) is 128x128, fig (e) is 64x64 size image.

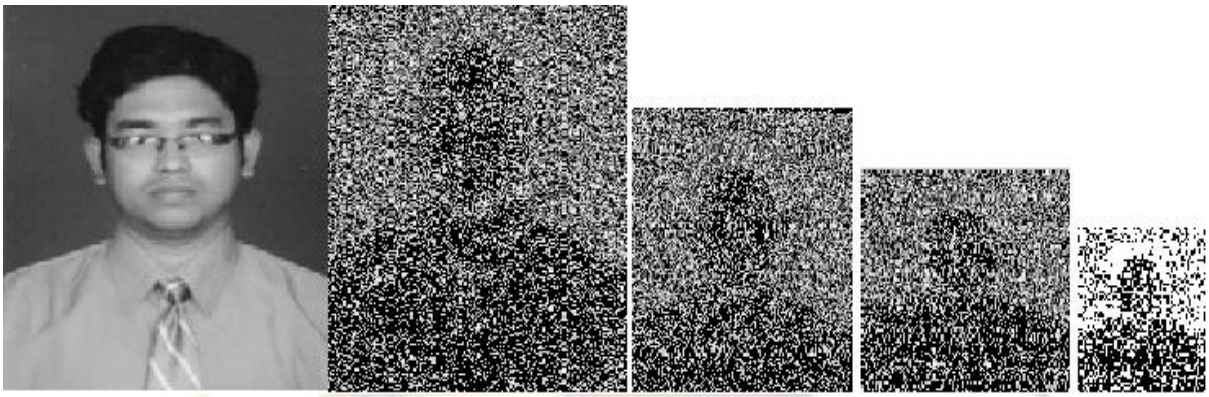


Fig (a) fig (b) fig (c) fig (d) fig (e)

Fig (1) Encrypted and Compressed Images

## C. IMAGE RECONSTRUCTION

With the bitstreams and the secret key, a receiver can reconstruct the principal content of the original image, and the resolution of the reconstructed image is dependent on the number of received bitstreams. While BG provides the rough information of the original content, $BS^{(t)}$ can be used to reconstruct the detailed content with an iteratively updating procedure. The image reconstruction procedure is as follows.

When having the bitstream BG, the decoder may obtain the values of $b(i,j)$ and decrypts them as a subimage, i.e.,

$$p^{(T)}(i,j) = mod[b(i,j).\Delta - e(2^T . i. 2^T . j).256] + \frac{\Delta}{2}, \quad 1 \le i \le \frac{N_1}{2^T}, \quad 1 \le j \le \frac{N_2}{2^T}$$

Where $e(2^T . i. 2^T . j)$ are derived from the secret key.If the bitstreams$BS^{(t)}$ $(\tau \le t \le T)$ are also received, an image with a size of $N_1/2^{(\tau-1)} \times N_2/2^{(\tau-1)}$ will be reconstructed. First, upsample the subimage $p^{(T)}(i,j)$ by factor $2^{(T-\tau+1)}$ to yield an $N_1/2^{(\tau-1)} \times N_2/2^{(\tau-1)}$ image as follows:

$$r\left(2^{(T-\tau+1)} . i . 2^{(T-\tau+1)} . j\right) = p^{(T)}(i,j), \quad 1 \le i \le \frac{N_1}{2^T}, \quad 1 \le j \le \frac{N_2}{2^T}$$

and estimate the values of other pixels according to the pixel values using a bilinear interpolation method.

Denote the interpolated pixel values of the *K*th group at the *t*th level as

$$r_k^{(t)}(1), r_k^{(t)}(2) \dots\dots r_k^{(t)}\left(L^{(t)}\right) \left(1 \le k \le K^{(t)}, \tau \le t \le T\right)$$

and their corresponding original pixel values as $p_k^{(t)}(1), p_k^{(t)}(2) \dots\dots p_k^{(t)}\left(L^{(t)}\right)$. The errors of interpolated values are

$$\Delta p_k^{(t)}(l) = p_k^{(t)}(l) - r_k^{(t)}(l), \quad 1 \le l \le L^{(t)}, \ 1 \le k \le K^{(t)}, \tau \le t \le T.$$

Define the encrypted values of $r_k^{(t)}(l)$ as

$$\hat{r}_k^{(t)}(l) = mod\left[r_k^{(t)}(l) + e_k^{(t)}(l), 256\right], \quad 1 \le l \le L^{(t)}, \ 1 \le k \le K^{(t)}, \tau \le t \le T.$$

Where $e_k^{(t)}(l)$ are pseudorandom numbers derived from the secret key and corresponding to $r_k^{(t)}(l)$. Then

$$\Delta p_k^{(t)}(l) \equiv q_k^{(t)}(l) - \hat{r}_k^{(t)}(l) mod\ 256.$$

We also define

$$\begin{bmatrix} \Delta C_k^{(t)}(1) \\ \Delta C_k^{(t)}(2) \\ \cdot \\ \cdot \\ \cdot \\ \Delta C_k^{(t)}(L^{(t)}) \end{bmatrix} = \boldsymbol{H} \cdot \begin{bmatrix} \Delta p_k^{(t)}(1) \\ \Delta p_k^{(t)}(2) \\ \cdot \\ \cdot \\ \cdot \\ \Delta p_k^{(t)}(L^{(t)}) \end{bmatrix}$$

Where **H** is a $L^{(t)} \times L^{(t)}$ Hadamard matrix made up of +1 or -1. Since only the addition and subtraction are involved in the Hadamard transform

$$\begin{bmatrix} \Delta C_k^{(t)}(1) \\ \Delta C_k^{(t)}(2) \\ \cdot \\ \cdot \\ \cdot \\ \Delta C_k^{(t)}(L^{(t)}) \end{bmatrix} \equiv \boldsymbol{H} \cdot \begin{bmatrix} \Delta p_k^{(t)}(1) \\ \Delta q_k^{(t)}(2) \\ \cdot \\ \cdot \\ \cdot \\ \Delta q_k^{(t)}(L^{(t)}) \end{bmatrix} - \boldsymbol{H}. \begin{bmatrix} \hat{r}_k^{(t)}(1) \\ \hat{r}_k^{(t)}(2) \\ \cdot \\ \cdot \\ \cdot \\ \hat{r}_k^{(t)}(L^{(t)}) \end{bmatrix} mod\ 256$$

That means the transform of errors in the plain domain is equivalent to the transform of errors in the encrypted domain with the modular arithmetic. Denoting

$$\begin{bmatrix} \hat{C}_k^{(t)}(1) \\ \hat{C}_k^{(t)}(2) \\ \cdot \\ \cdot \\ \cdot \\ \hat{C}_k^{(t)}(L^{(t)}) \end{bmatrix} = \boldsymbol{H}. \begin{bmatrix} \hat{r}_k^{(t)}(1) \\ \hat{r}_k^{(t)}(2) \\ \cdot \\ \cdot \\ \cdot \\ \hat{r}_k^{(t)}(L^{(t)}) \end{bmatrix}$$

We have

$$\Delta C_k^{(t)}(l) \equiv C_k^{(t)}(l) - \hat{C}_k^{(t)}(l) mod\ 256$$

With the bitstreams $BS^{(t)}$ $(\tau \le t \le T)$ , the values of $C_k^{(t)}(l)$ can be retrived, which provide the information of $C_k^{(t)}(l)$. Therefore, the receiver may use an iterative procedure to progressively improve the quality of the reconstructed image by updating the pixel values according to $C_k^{(t)}(l)$. The detailed procedure is as follows.

1) For each group $\left[r_k^{(t)}(1), r_k^{(t)}(2) \ldots \ldots r_k^{(t)}(L^{(t)})\right]$, calculate $\hat{r}_k^{(t)}(l)$ and $\hat{C}_k^{(t)}(l)$.

2) Calculate

$$D_k^{(t)}(l) = mod\left[c_k^{(t)}(l) \cdot \Delta^{(t)} + \Delta^{(t)}/2 - \hat{C}_k^{(t)}(l) \cdot 256\right]$$

$$\widehat{D}_k^{(t)}(l) = \begin{cases} D_k(l), & if\ d_k(l) < 128 \\ D_k(l) - 256, & if\ d_k(l) \ge 128 \end{cases}$$

$\widehat{D}_k^{(t)}(l)$are the differences between the values consistent with the corresponding $c_k^{(t)}(l)$ and $\hat{C}_k^{(t)}(l)$. Then, considering $\widehat{D}_k^{(t)}(l)$ as an estimate of $\Delta C_k^{(t)}(l)$, modify the pixel values of each group as follows:

$$\begin{bmatrix} \bar{r}_k^{(t)}(1) \\ \bar{r}_k^{(t)}(2) \\ \cdot \\ \cdot \\ \cdot \\ \bar{r}_k^{(t)}(L) \end{bmatrix} = \begin{bmatrix} r_k^{(t)}(1) \\ r_k^{(t)}(2) \\ \cdot \\ \cdot \\ \cdot \\ r_k^{(t)}(L) \end{bmatrix} + \frac{\boldsymbol{H}'}{L^{(t)}} \cdot \begin{bmatrix} \widehat{D}_k^{(t)}(1) \\ \widehat{D}_k^{(t)}(2) \\ \cdot \\ \cdot \\ \cdot \\ \widehat{D}_k^{(t)}(L^{(t)}) \end{bmatrix}$$

And enforce the modified pixel values into [0,255] as follows:

$$\bar{r}_k^{(t)}(l) = \begin{cases} 0, & if \bar{r}_k^{(t)}(1) < 0 \\ \bar{r}_k^{(t)}(1), & if\ 0 \le \bar{r}_k^{(t)}(1) \le 255 \\ 255, & if \bar{r}_k^{(t)}(1) > 255 \end{cases}$$

3) Calculate the average energy of difference due to the modification as follows:

$$D = \frac{\sum_{t=\tau}^{T} \sum_{k=1}^{K^{(t)}} \sum_{l=1}^{L^{(t)}} \left[\hat{r}_k^{(t)}(l) - r_k^{(t)}(l)\right]^2}{\sum_{t=\tau}^{T} 3N/4^t}$$

If D is not less than a given threshold of 0.10, for each pixel $\hat{r}_k^{(t)}(l)$, after putting it back to the position in the image and regarding the average value of its four neighbor pixels as its new value $r_k^{(t)}(l)$, go to step 1. Otherwise, terminate the iteration, and output the image as a final reconstructed result.

In the iterative procedure, while the decrypted pixels $p^{(T)}(i,j)$ are used to give an initial estimation of other pixels, the values of $c_k^{(t)}(l)$ in bitstreams $BS^{(t)}$ provide more detailed information to produce the final reconstructed result with satisfactory quality. In step 2, by estimating $\Delta C_k^{(t)}(l)$ according to $c_k^{(t)}(l)$, the pixel values are modified to lower the reconstruction errors. If the image is uneven and $L^{(t)}$is big, the absolute value of actual $\Delta C_k^{(t)}(l)$ may be more than 128 due to error accumulation in a group, so that $\widehat{D}_k^{(t)}(l)$ maybe not close to $\Delta C_k^{(t)}(l)$. To avoid this case, we let $L^{(t)}$decrease with a increasing $t$ since the spatial correlation in a subimage with lower resolution is weaker. For instance, $L^{(1)} = 24$, $L^{(2)} = 8$, $L^{(3)} = 4$ for T=3.

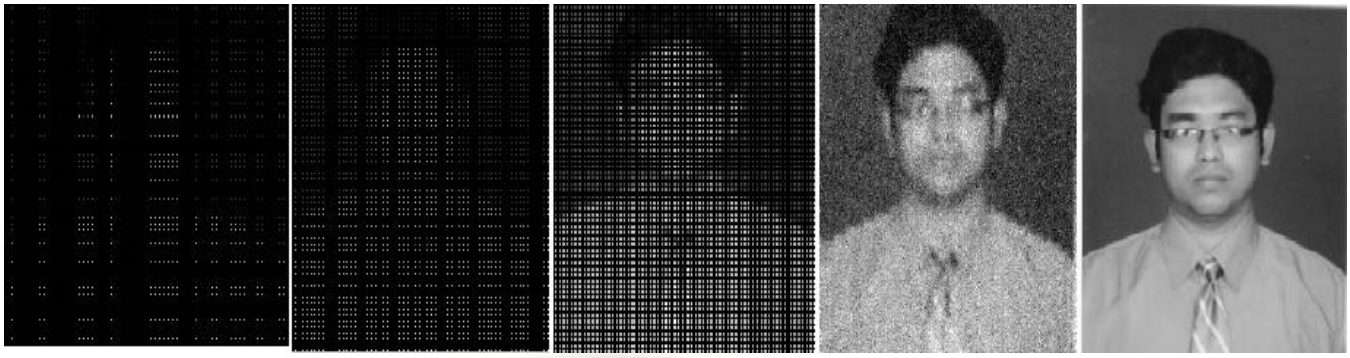


Fig (a) fig (b) fig (c) fig (d) fig (e)

Fig (2) decompressed and decoded images

### D. OVERALL DIAGRAM


Fig (2) block diagram

## II. PROPOSED SYSTEM ALGORITHM

### A. SCALABLE CODING SCHEME

In the proposed scheme, a series of pseudorandom numbers derived from a secret key are used to encrypt the original pixel values. After decomposing the encrypted data into a subimage and several data sets with a multiple-resolution construction, an encoder quantizes the subimage and the Hadamard coefficients of each data set to effectively reduce the data amount. Then, the quantized subimage and coefficients are regarded as a set of bit streams. When having the encoded bit streams and the secret key, a decoder can first obtain an approximate image by decrypting the quantized subimage and then reconstructing the detailed content using the quantized coefficients with the aid of spatial correlation in natural images. Because of the hierarchical coding mechanism, the principal original content with higher resolution can be reconstructed when more bit streams are received.

## III. FUTURE ENHANCEMENT

In Oder to reduce the size of the compressed image we can use block Truncation Coding (BTC) for compression; Block Truncation Code (BTC) is digital technique in image processing using which images can be coded efficiently. BTC has played an important role

in the sense that many coding techniques have been developed based on it. its main attraction being its simple underlying concepts and ease of implementation.

## IV. ADVANTAGES

- The subimage is decrypted to produce an approximate image; the quantized data of Hadamard coefficients can provide more detailed information for image reconstruction.
- Bitstreams are generated with a multiple-resolution construction, the principal content with higher resolution can be obtained when more bitstreams are received.

## V. TABLES

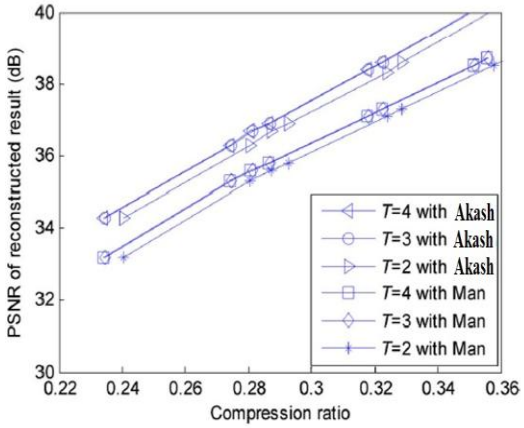
### A. COMPRESSION RATIO

TABLE I

COMPRESSION RATIOS, PSNR IN RECONSTRUCTED RESULTS AND ITERATION NUMBERS WITH DIFFERENT $M$ WHEN $T = 3$, $L^{(3)} = 4$, $L^{(2)} = 8$, AND $L^{(1)} = 24$ WERE USED FOR LENA AND MAN

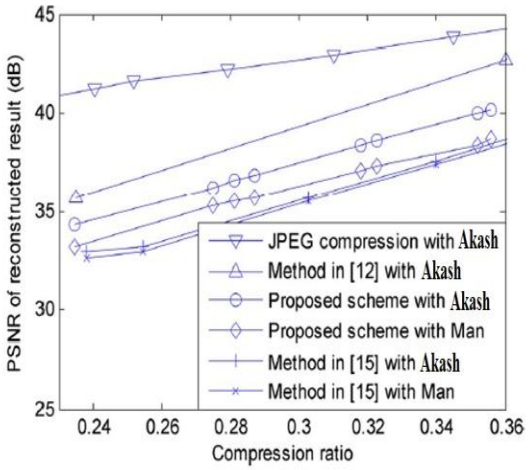
| $M$ | 16 | 18 | 22 | 24 | 26 | 30 |
|---|---|---|---|---|---|---|
| $R_C$ | 0.235 | 0.275 | 0.287 | 0.318 | 0.323 | 0.356 |
| $PSNR_B$(dB) | 34.8, 34.8 | 35.9, 35.8 | 37.6, 37.6 | 38.4, 38.4 | 39.1, 39.1 | 40.2, 40.2 |
| $PSNR_3$(dB) | 31.7, 30.9 | 32.3, 31.4 | 32.7, 32.6 | 34.0, 31.9 | 34.1, 32.8 | 34.3, 34.0 |
| Iteration number | 10, 6 | 10, 10 | 10, 4 | 5, 9 | 9, 9 | 9, 10 |
| $PSNR_2$(dB) | 34.2, 32.6 | 34.7, 32.8 | 36.7, 33.9 | 37.1, 33.9 | 37.7, 35.0 | 39.1, 35.7 |
| Iteration number | 16, 14 | 4, 13 | 9, 10 | 9, 11 | 4, 15 | 4, 11 |
| $PSNR_1$(dB) | 34.3, 33.2 | 36.2, 35.3 | 36.8, 35.7 | 38.4, 37.1 | 38.6, 37.3 | 40.2, 38.7 |
| Iteration number | 6, 25 | 5, 22 | 5, 15 | 5, 15 | 5, 10 | 5, 18 |

### B. COMPRESSION RATIO GRAPH




Performance of the proposed scheme with different $T$.

### C. PERFORMANCE COMPERISION OF SEVERAL COMPRESSION MERTHODS




Performance comparison of several compression methods.

## VI. CONCLUSION

This paper has proposed a novel scheme of scalable coding for encrypted images. The original image is encrypted by a modulo-256 addition with pseudorandom numbers, and the encoded bitstreams are made up of a quantized encrypted subimage and the quantized remainders of Hadamard coefficients. At the receiver side, while the subimage is decrypted to produce an approximate image, the quantized data of Hadamard coefficients can provide more detailed information for image reconstruction. Since the bitstreams are generated with a multiple-resolution construction, the principal content with higher resolution can be obtained when more bitstreams are received.

## REFERENCES

[1]Xinpeng Zhang, *Member*, *IEEE*, Guorui Feng, YanliRen, and ZhenxingQian." Scalable Coding of Encrypted Images". IEEE transactions on image processing, vol. 21, no. 6, June 2012

[2] Z. Erkin, A. Piva, S. Katzenbeisser, R. L. Lagendijk, J. Shokrollahi, G. Neven, and M. Barni, "Protection and retrieval of encrypted multimedia content: When cryptography meets signal processing," EURASIP J. Inf. Security, vol. 2007, pp. 1–20, Jan. 2007.

[3] 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.

[4] J. R. Troncoso-Pastoriza and F. Pérez-González, "Secure adaptive filtering," IEEE Trans. Inf. Forensics Security, vol. 6, no. 2, pp. 469–485, Jun. 2011.

[5] T. Bianchi, A. Piva, and M. Barni, "Composite signal representation for fast and storage-efficient processing of encrypted signals," IEEE Trans. Inf. Forensics Security, vol. 5, no. 1, pp. 180–187, Mar. 2010.

[6] S. Lian, Z. Liu, Z. Ren, and H. Wang, "Commutative encryption and watermarking in video compression," IEEE Trans. Circuits Syst. Video Technol., vol. 17, no. 6, pp. 774–778, Jun. 2007.

[7] M. Cancellaro, F. Battisti, M. Carli, G. Boato, F. G. B. Natale, and A. Neri, "A commutative digital image watermarking and encryption method in the tree structured Haar transform domain," Signal Process.Image Commun., vol. 26, no. 1, pp. 1–12, Jan. 2011.

[8] N. Memon and P. W. Wong, "A buyer-seller watermarking protocol,"IEEE Trans. Image Process., vol. 10, no. 4, pp. 643–649, Apr. 2001.

[9] M. Kuribayashi and H. Tanaka, "Fingerprinting protocol for images based on additive homomorphic property," IEEE Trans. Image Process., vol. 14, no. 12, pp. 2129–2139, Dec. 2005.

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

[11] D. Schonberg, S. C. Draper, and K. Ramchandran, "On blind compression of encrypted correlated data approaching the source entropy rate," in Proc. 43rd Annu. Allerton Conf., Allerton, IL, 2005.

[12] R. Lazzeretti and M. Barni, "Lossless compression of encrypted greylevel and

color images," in Proc. 16th EUSIPCO, Lausanne, Switzerland, Aug. 2008 .

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

[14] 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.

[15] A. Kumar and A. Makur, "Lossy compression of encrypted image by compressing sensing technique," in Proc. IEEE TENCON, 2009, pp. 1–6

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

[17] A. Bilgin, P. J. Sementilli, F. Sheng, and M. W. Marcellin, "Scalable image coding using reversible integerwavelet transforms," IEEE Trans.Image Process., vol. 9, no. 11, pp. 1972–1977, Nov. 2000.

[18] D. Taubman, "High performance scalable image compression with EBCOT," IEEE Trans. Image Process., vol. 9, no. 7, pp. 1158–1170, Jul. 2000.