International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 4, April - 2014

Encryption of Gray Images using Scalable Codes

Pooja Bhat
Department of E&TC
TSSM's BSCOER
Pune Maharashtra

P. K. Ajmera
Department of E&TC
TSSM's BSCOER
Pune Maharashtra

***Abstract*— In this paper we have used a method for encryption of images using scalable coding. In encryption, a pseudorandom number derived from a secret key is used to encrypt the original pixel values. In encoding block, encrypted data is decomposed into a down sampled sub image and several data sets with a multiple-resolution construction. Thus an encoder quantizes the sub image and calculates the Hadamard coefficients of each data set that will reduce the data amount. Then, this quantized sub image and Hadamard coefficients are regarded as a set of bitstreams. At the receiver end, while decryption of sub image will give the rough information of the original content, the quantized coefficients can be decrypted to regenerate the detailed content with an iteratively updating procedure. At the output we will first obtain a lossy version of image and images of higher resolution can be obtained when more and more bitstreams are received. PSNR of reconstructed images is calculated for all scales. When more number of decomposition levels are used to decompose encrypted data, PSNR performance is good.**

Keywords--Bitstreams, Hadamard transform, Cryptography, Image compression, Image encryption, Signal Processing in the Encrypted Domain

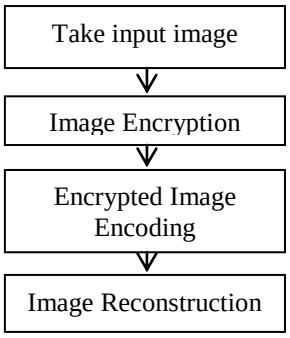
I. INTRODUCTION

In past few years, processing of encrypted signals has emerged as a new and challenging research topic [1]. Depending upon the holomorphic properties of a cryptosystem, the discrete Fourier transform and adaptive filtering [2], [3] can be implemented in the encrypted domain and a composite signal representation method can be used to reduce the size of encrypted data [4]. In case of joint encryption and data hiding, a part of significant data 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.

Lots of work has been done on compressing encrypted images. 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. In [5], it has been proved that the performance of compressing encrypted data may be as good as that of compressing non encrypted data in theory. [6] presents several methods for lossless compression of encrypted gray. In most of the already employed schemes, the syndrome of channel code is exploited to generate the compressed data in a lossless manner. Several methods have been developed for lossy compressing encrypted images. Lossy compression of encrypted images is achieved in [7]. The original gray level image is encrypted by pixel permutation in [8]. Then, the encrypted data are compressed by discarding the excessively rough and fine information of coefficients generated from orthogonal transform.

This paper proposes a method of scalable coding for encryption of gray images. Although lots of works have been done on scalable coding of unencrypted images [9], [10], the scalable coding of encrypted data has not yet been reported. In the encryption phase, the pixel values are completely altered so that an attacker cannot obtain any information of original image. Then, the encrypted data are decomposed into several parts, and each part is compressed as a bitstream. At the receiver side with the help of cryptographic key, the images of higher resolution can be reconstructed when more number of bitstreams is received.

II. BLOCK DIAGRAM



A. *Image Encryption*

The input to this block is uncompressed image with pixel values in the range [0 255] and pixel number as N=N1×N2 where N1 and N2 are number of rows and columns. Thus bit amount of input image is 8N. A pseudorandom bit sequence of 8N is generated to encrypt the input image. Encryption and decryption block has the same pseudorandom number generator (PRNG). An encrypted image is produced by a one-by-one addition modulo 256 as follows:

$$g^{(0)}(i,j) = \mathrm{mod}[p(i,j)+e(i,j),256], \quad 1<i<N1,\ 1\leq j\leq N2 \qquad (1)$$

where p(i,j) represents the gray values of input image pixels at positions (i,j), e(i,j) represents the pseudorandom numbers generated by the PRNG within range [0, 255], and $g^{(0)}(i,j)$ represents the final encrypted pixel values. Clearly, the output of this stage i.e. encrypted pixel values $g^{(0)}(i,j)$ are pseudorandom numbers since e(i,j) values are pseudorandom numbers.

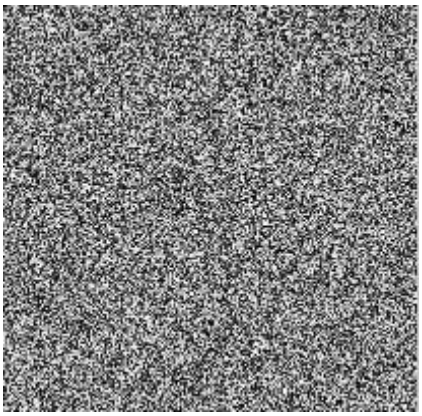


Fig.1 Input Image and Encrypted image.

B. Encoding of encrypted images

Encrypted data is compressed as a set of bitstreams. Encoding procedure is explained in detail as follows:

The encoder first decomposes the encrypted image into a number of subimages and data sets with a multiple-level resolution. The subimage at the (*t+1*)th level $G^{(t+1)}$ is produced by downsampling the subimage at the *t*th level is explained below:

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

where $G^{(0)}$ is the encrypted image and T is the number of decomposition levels.

The encrypted pixels that belongs to $G^{(t)}$but do not belong to $G^{(t+1)}$ form a data set $Q^{(t+1)}$ as explained below:

$$Q^{(t+1)} = \{g^{(t)}(i,j) \mid \text{mod}(i,2)=1 \text{ or } \text{mod}(j,2)=1\}$$
$$t=0,1,\ldots\ldots\ldots\ldots,T-1 \qquad (3)$$

The output of this block is reorganized as $G^{(T)}, Q^{(T)}, Q^{(T-1)}$…….. and $Q^{(1)}$.

The encoder quantizes each value of subimage G(t) using a step Δ as follows:

$$b(i,j) = \left\lfloor \frac{g^{(T)}(i,j)}{\Delta} \right\rfloor \qquad (4)$$

$$\Delta = \frac{256}{M} \qquad (5)$$

where M is an integer that will be shared by encoder and decoder. Then, the data of b(i,j) are converted into a bitstream denoted by BG.

For each data set $Q^{(t)}$, the encoder divides encrypted pixels in it into $K^{(t)}$ groups, each of which contains $L^{(t)}$ pixels ($K^{(t)} * L^{(t)}=3N/4^t$). In this way, the $L^{(t)}$ pixels in the same group get scattered in the entire image. The encrypted pixels of the *k*th group are denoted as $q_k^{(t)}(1)$, $q_k^{(t)}(2)\ldots$ $q_k^{(t)}(L^{(t)})$ and perform the Hadamard transform in each group as given below:

$$\begin{bmatrix} C_k^{(t)}(1) \\ C_k^{(t)}(2) \\ . \\ . \\ C_k^{(t)}(L^{(t)}) \end{bmatrix} = H. \begin{bmatrix} q_k^{(t)}(1) \\ q_k^{(t)}(2) \\ . \\ . \\ q_k^{(t)}(L^{(t)}) \end{bmatrix} \qquad (6)$$

where H is a Hadamard matrix of order $L^{(t)} * L^{(t)}$ made up of +1 or -1.

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

$$C_k^{(t)}(l) = \left\lfloor \frac{\text{mod}\,[C_k^{(t)}(l), 256]}{256/M^{(t)}} \right\rfloor, \quad 1 \le k \le K^{(t)},\ 1 \le l \le L^{(t)} \qquad (7)$$

where $M^{(t)}$= round ($M/\sqrt{L^{(t)}}$) (8)

The encoder transmits the bitstreams with an order of {BG,$BS^{(T)}$,$BS^{(T-1)}\ldots$ $BS^{(1)}$}. We cab discard later bitstreams in case of limited bandwidth

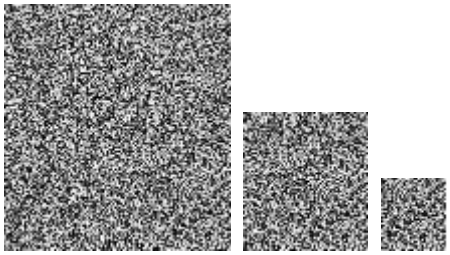


Fig 2. Compressed Images

C. Reconstruction of Image

The receiver first reconstructs the lossy version of the image with the bitstreams and the secret key and the resolution of the reconstructed image can be increased depending on the number of bitstreams received. BG provides rough information of the original image that is it generates a lossy version of the image, $BS^{(t)}$is used to reconstruct the detailed content with an iteratively updating procedure. The image reconstruction procedure is as explained below:

The decoder will first decrypt the bitstream BG to obtain the values of b(i,j) and decrypts them as a subimage i.e

Fig 3. Reconstructed Image using

$$\{BS^{(B)}\}, \{BS^{(B)}, BS^{(3)}\}, \{BS^{(B)}, BS^{(3)}, BS^{(2)}\}, \{BS^{(B)}, BS^{(3)}, BS^{(2)}, BS^{(1)}\}$$

$p^{(T)}(i,j)= \text{mod}\,[b(i,j).\Delta - e(2^T.\, i, 2^T.\, j), 256] + \frac{\Delta}{2}$ (9)

where e $(2^T.\, i, 2^T.\, j)$ are derived from the secret key.

The bitstreams $BS^{(T)}$ are decrypted to obtain image with a size of $N_1/\ 2^{(T-1)} \times N_2/\ 2^{(T-1)}$. First, upsample the subimage $p^{(T)}(i,j)$ to yield an $N_1/\ 2^{(T-1)} \times N_2/\ 2^{(T-1)}$ image as follows:

$r(2^{(T-T-1)}.\, i, 2^{(T-T-1)}.\, j) = p^{(T)}(i,j)$ (10)

and the values of other pixels are estimated according to the pixel values in (14) using a bilinear interpolation method. Then, the interpolated pixels are reorganized as data sets with multiple-resolution construction. Denote the interpolated pixel values of the *k*th group at the tth level as $r_k^{(t)}(1)$, $r_k^{(t)}(2)$, $\ldots r_k^{(t)}(L^{(t)})$ and their corresponding original pixel values as $p_k^{(t)}(1)$, $p_k^{(t)}(2)$, $\ldots p_k^{(t)}(L^{(t)})$.

The errors of interpolated values are

$\Delta p_k^{(t)}(l) = p_k^{(t)}(l) - r_k^{(t)}(l)$ (11)

The encrypted values of $r_k^{(t)}(l)$ are defined as

$\hat{r}_k^{(t)}(l) = \text{mod}[r_k^{(t)}(l) + e_k^{(t)}(l), 256]$ (12)

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

$\Delta p_k^{(t)}(l) = q_k^{(t)}(l) - \hat{r}_k^{(t)}(l) \bmod 256$ (13)

Denoting

$$\begin{bmatrix} \hat{C}_k^{(t)}(1) \\ \hat{C}_k^{(t)}(2) \\ . \\ . \\ \hat{C}_k^{(t)}(L^{(t)}) \end{bmatrix} = H \times \begin{bmatrix} \hat{r}_k^{(t)}(1) \\ \hat{r}_k^{(t)}(2) \\ . \\ . \\ \hat{r}_k^{(t)}(L^{(t)}) \end{bmatrix} \quad (14)$$

We also define $\Delta C_k^{(t)}(l) = C_k^{(t)}(l) - \hat{C}_k^{(t)}(l) \bmod 256$ (15)

With the bitstreams $BS^{(T)}$, the values of $c_k^{(t)}(l)$ can be retrieved, 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.

For each group $[r_k^{(t)}(1), r_k^{(t)}(2), \ldots r_k^{(t)}(L^{(t)})]$, calculate $\hat{r}_k^{(t)}(l)$ and $C_k^{(t)}(l)$ using 12 and 14.

1) Calculate

$D_k^{(t)}(l) = \text{mod}\,[c_k^{(t)}(l) \,.\, \Delta^{(t)} + \hat{C}_k^{(t)}(l), 256]$ (16)

$$\tilde{D}_k^{(t)}(l) = \begin{cases} D_k(l) & \text{if } d_k(l) < 128 \\ D_k(l) - 256 & \text{if } d_k(l) \geq 128 \end{cases} \quad (17)$$

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

$$\begin{bmatrix} \bar{r}_k^{(t)}(1) \\ \bar{r}_k^{(t)}(2) \\ . \\ . \\ \bar{r}_k^{(t)}(L) \end{bmatrix} = \begin{bmatrix} r_k^{(t)}(1) \\ r_k^{(t)}(2) \\ . \\ . \\ r_k^{(t)}(L) \end{bmatrix} + \frac{H'}{L^{(t)}} \begin{bmatrix} \tilde{D}_k^{(t)}(1) \\ \tilde{D}_k^{(t)}(2) \\ . \\ . \\ \tilde{D}_k^{(t)}(L^{(t)}) \end{bmatrix} \quad (18)$$

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

$$\hat{r}_k^{(t)}(l) = \begin{cases} 0 & \text{if } \bar{r}_k^{(t)}(l) < 0 \\ \bar{r}_k^{(t)}(l) & \text{if } 0 \leq \bar{r}_k^{(t)}(l) \leq 255 \\ 255 & \text{if } \bar{r}_k^{(t)}(l) > 255 \end{cases} \quad (19)$$

In this iterative procedure, the decrypted pixels $p^{(T)}(i,j)$ gives an initial estimation of other pixels, the values of $c_k^{(t)}(l)$ in bitstreams $BS^{(t)}$ provide more detailed information of image 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 value of actual $\Delta C_k^{(t)}(l)$ may be more than 128 due to the error

accumulation in a group, so that $\bar{D}_k^{(t)}(l)$ in (17) may be not close to $\Delta C_k^{(t)}(l)$.

III. Experimental evaluation

PSNR (DB) for respective Image Size	M	16	18	22	24	26	30
	512*512	35.3984	36.1934	36.6648	36.8650	37.1732	37.4471
	256*256	33.0994	33.6132	33.7732	34.0868	34.2689	34.5077
	128*128	31.4452	32.5327	32.8160	33.2576	33.1831	33.2224
	64*64	38.0271	39.2277	40.4000	40.8736	41.1614	41.7291

Test image of size 512*512 is taken as input image and is decomposed into 3 scales (T=3) and M is varied from 16, 18, 22, 24, 26, 30. PSNR of reconstructed images are calculated. As the value of M is increased, PSNR is improved.

IV. Conclusion

This paper has presented a method for encryption & reconstruction of images using scalable codes. The encryption is achieved by modulo-256 addition of original input image with pseudorandom numbers. Then decomposition of encrypted image is done and is divided into sub image and data sets with a multiple-resolution construction. The quantization of subimage and the Hadamard coefficients of each data set is done to effectively reduce the data amount. The quantized subimage and Hadamard coefficients are considered as a set of bit streams. At the receiver side, the sub image is decrypted to produce a lossy version of image; the quantized data of Hadamard coefficients provide more detailed information for image reconstruction. Since the bit streams are generated with a multiple-resolution construction, when more bit streams are received the resolution of reconstructed image can be improved with iterative updating procedure.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] R. Lazzeretti and M. Barni, "Lossless compression of encrypted greylevel and color images," in Proc. 16th EUSIPCO, Lausanne, Switzerland, Aug. 2008 [Online]. Available: http://www.eurasip.org/Proceedings/Eusipco/Eusipco2008/papers/1569105134.pdf

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

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

[9] 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.

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