Image Encryption and Iterative Reconstruction of an Encrypted Image

N. Nagaraja Kumar [1], M. Sucharitha [2]

[1]Assistant Professor, RGMCET, Nandyal-518501, Kurnool (dist), Andhra Pradesh, India

[2]M.Tech (DSCE), Student, RGMCET, Nandyal-518501, Kurnool (dist), Andhra Pradesh, India

Abstract: ***Encryption is one of the ways to ensure high security for real time applications. In advanced Encryption and data-hiding, Encryption is done only on primary part of image and remaining part is sent as a plain image for secure data communication .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. After encoding and on compression of encrypted data into a down sampled sub image and several data sets with a multiple-resolution construction, an encoder quantizes the sub image and the Hadamard-coefficients of each data set are obtained which tend to reduce the data amount. Obtained set of bit streams can be used to reconstruct the detailed content with an iteratively updating procedure and then decryption of data is performed. Because of this iteratively updating coding procedure, the principal original content with higher resolution can be reconstructed when more bit streams are received.***

Keywords: Encryption, Cryptographic key, Bit-streams, Hadamard transform.

1. Introduction

The processing of encrypted signals has emerged as a new and challenging research field [1] in these years. The combination of cryptographic techniques and signal processing is not new. So far, encryption was always considered as an add on after signal manipulations had taken place. For instance, when encrypting compressed multimedia signals such as audio, images, and video, first the multimedia signals were compressed using state-of-the-art compression techniques, and next encryption of the compressed bit stream using a symmetric cryptosystem took place. Consequently, the bit stream must be decrypted before the multimedia signal can be decompressed.

In several application scenarios, however, it is desirable to carry out signal processing operations directly on encrypted signals. Such an approach is called secure signal processing, encrypted signal processing, or signal processing in the encrypted domain. For instance, given an encrypted image, we can calculate the mean value of the encrypted image pixels. On the one hand, the relevance of carrying out such signal manipulations, that is, the algorithm, directly on encrypted signals is entirely dependent on the security requirements of the application scenario under consideration. On the other hand, the particular implementation of the signal processing algorithm will be determined strongly by the possibilities and impossibilities of the cryptosystem employed. Now a days we have various techniques to provide security to data and secure transmission of data is made possible. Finally, it is very likely that new requirements for cryptosystems [2] will emerge from secure signal processing operations and applications. Hence, secure signal processing poses a joint challenge for both the signal processing and the cryptographic community.

2. Image Encryption

Assume that 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 $N (N = N_1 \times N_2)$.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 [2]. Then, the content owner divides the pseudorandom bit sequence into Npieces, each of which containing 8bits, and converts each piece as an integer number within [0, 255].

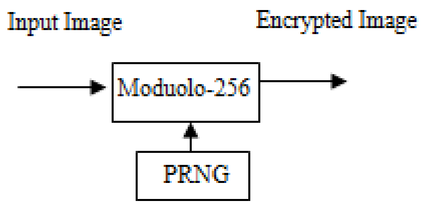

Figure2.1: Image Encryption

$$s^{(0)}(m,n) = mod[p(m,n) + e(m,n), 256], \quad 1 \le m \le N_1 , 1 \le n \le N_2 \quad (1)$$

Where $p(m,n)$represents the gray values of pixels at positions (m,n) , $e(m,n)$ represents the pseudorandom numbers within [0, 255] generated by the PRNG, and $s^{(0)}(m,n)$ represents the encrypted pixel values. Clearly, the encrypted pixel values $s^{(0)}(m,n)$ are pseudorandom numbers since $e(m,n)$ values are pseudorandom numbers. We know that there is no probability polynomial time algorithm to distinguish a pseudorandom number and random number.

3. Encrypted Image Encoding

The complete encoding procedure is given below:

a) Encoder is used to convert the one form of data into another form. Here, the encoder decomposes the encrypted image into a series of sub images [3] and data sets with a multiple resolution construction.

b) The sub images at the (t+1) and $s^{(t+1)}$ level is generated by down sampling the sub image at the t^{th} level as follows
$s^{(t+1)}(m,n) = s^{(t)}(2m,2n), t = 0,1,\dots,T-1$ (2)

Where $S^{(0)}$ just the encrypted image and T is is the number of decomposition levels.

c) The encrypted pixels that belong to $S^{(t)}$ but do not belong to $S^{(t+1)}$ form data set $Q^{(t+1)}$ as follows:
$Q^{(t+1)} = \{s^{(t)}(m,n) | mod(m,2) = 1 \; or \; mod(n,2) = 1\}$,
$t = 0,1,\dots.,T-1$ (3)

That means each $S^{(t)}$ is decomposed into $S^{(t+1)}$ and $Q^{(t+1)}$, and the data amount of $Q^{(t+1)}$ is three times of that of $S^{(t+1)}$.

d) After the multiple-level decomposition, the encrypted image is updated as

$$S^{(T)}, Q^{(T)}, Q^{(T-1)}, \dots, and\; Q^{(1)}$$

(e) Perform quantization for the values of encoded pixels.

$b(m,n) = \left\lfloor \frac{s^{(T)}(m,n)}{\Delta} \right\rfloor$ (4)
Where ,
$\Delta = \frac{256}{M}$
Here, M is an integer shared by the encoder and the decoder. Clearly
$0 \le b(m,n) \le M-1$ (5)

f) The data of b (m, n) are converted into bit-stream which is denoted as BG. BG is bit amount,
$N_{BG} = \frac{N}{4^T} \cdot \log_2 M$ (6)

For each data set $Q^{(t)}(t = 1,2,\dots,T)$, the encoder permutes and divides encrypted pixels into $K^{(t)}$groups, each of which containing $L^{(t)}$pixels ($K^{(t)} \times L^{(t)} = 3\,N/4^t$. In this way, the $L^{(t)}$pixels [4] in the same group scatter in the entire image. The permutation way is shared by the encoder and the decoder, and the values of $L^{(t)}$. Denote the encrypted pixels of the kth group as $q_k^{(t)}(1), q_k^{(t)}(2), \dots, q_k^{(t)}(L^{(t)})(1 \le k \le K^{(t)})$.

(g) Hadamard transform is applied on each group

$$\begin{bmatrix} C_k^{(t)}(1) \\ C_k^{(t)}(2) \\ \vdots \\ C_k^{(t)|}(L^{(t)}) \end{bmatrix} = \mathbf{H} \cdot \begin{bmatrix} q_k^{(t)}(1) \\ q_k^{(t)}(2) \\ \vdots \\ q_k^{(t)|}(L^{(t)}) \end{bmatrix} \quad (7)$$

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

$$\mathrm{H}' \cdot \mathrm{H} = \mathrm{H} \cdot \mathrm{H}' = L^{(t)} \cdot \mathrm{I}$$

Where H' is a transpose of $\mathrm{H} \cdot \mathrm{I}$ is an $L^{(t)} \times L^{(t)}$ identity matrix, and $L^{(t)}$must be a multiple of 4. For each coefficient $C_k^{(t)}(l)$, calculate

$c_k^{(t)}(l) = \left\lfloor \frac{mod\left[c_k^{(t)}(l),256\right]}{256/M^{(t)}} \right\rfloor, 1 \le k \le K^{(t)}, 1 \le l \le L^{(t)}$ (8)

Where,
$M^{(t)} = round\left(M/\sqrt{L^{(t)}}\right)$

and round(·) finds the nearest integer. In (8), the remainder of $C_k^{(t)}(l)$ modulo-256 is quantized as integer $c_k^{(t)}(l)$, and the quantization stepsare approximately proportional to square roots of $L^{(t)}$. Then, $c_k^{(t)}(l)$ at different levels are converted into bit-streams, which are denoted as $\mathrm{BS}^{(t)}$. Since
$0 \le c(l) \le M^{(t)} - 1$ (9)

(h) The number of $c_k^{(t)}(l)$ at the t^{th} level is $3N/4^t$. Bit amount of $\mathrm{BS}^{(t)}$ is,

$N^{(t)} = \frac{3 \cdot N \cdot \log_2 M^{(t)}}{4^t}, t = 1,2,\dots.,T$ (10)

The bit-streams transmitted from an encoder in an order of $\{\mathrm{BG}, \mathrm{BS}^{(t)}, \mathrm{BS}^{(t-1)}, \dots, \mathrm{BS}^{(1)}\}$. If the channel bandwidth is limited, the latter bit streams may be abandoned.

(i) The total compression ratio R_C, which is a ratio between the amount of the encoded data and the encrypted image data, is

$$R_C = \frac{N_{BG}}{8N} + \frac{1}{8N}\sum_{t=1}^{T} N^{(t)} = \frac{\log_2 M}{8 \cdot 4^T} + \frac{3}{8} \cdot \sum_{t=1}^{T} \frac{\log_2 M^{(t)}}{4^t} \quad (11)$$

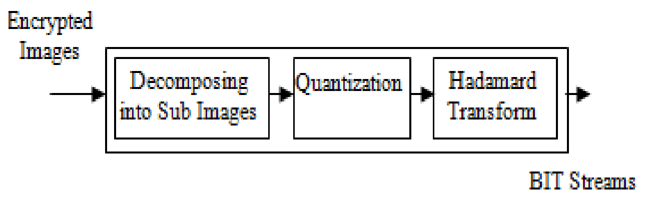

Figure 3.1: Image Encoding

4. Image Reconstruction

In Image Reconstruction process the procedure we follow is similar to Image encoding but the decryption of original data is performed firstly by subtracting the pseudo random number we have added to the image. Hence we get the original data and then we multiply the step size since we quantized the data by dividing with step size during encoding to obtain detail content. The bit streams and the secret key obtained, a decoder can reconstruct the principal content of the original image and if more number of bit-streams are received at receiver it leads to get high resolution of an image. While BG provides the rough information of the original content, $\mathrm{BS}^{(t)}$can be used to reconstruct the detailed content with an iteratively [5] updating procedure. The image reconstruction procedure is as follows.

When having the bit-stream BG, the decoder may obtain the values of $b(m,n)$and decrypts them as a sub-image, i.e.

$p^{(T)}(m,n) = mod\,[b(m,n) \cdot \Delta - e(2^T \cdot m, 2^T \cdot n), 256] + \frac{\Delta}{2}, 1 \le m \le \frac{N_1}{2^T}, 1 \le n \le \frac{N_2}{2^T}$ (12)

Where $(2^T \cdot m, 2^T \cdot n)$are derived from the secret key.

If the bit streams $\mathrm{BS}^{(t)}$ $(\tau \leq t \leq T)$are also received, an image with a size of $N_1/2^{(\tau-1)} \times N_1/2^{(\tau-1)}$will be reconstructed. First, up sample the sub-image $p^{(T)}(m,n)$ by factor $2^{(T-\tau+1)}$to yield an $N_1/2^{(\tau-1)} \times N_1/2^{(\tau-1)}$ image as follows:

$$r\left(2^{(T-\tau+1)} \cdot m, 2^{(T-\tau+1)} \cdot n\right) = p^T(m,n), 1 \leq m \leq \frac{N_1}{2^T}, 1 \leq n \leq \frac{N_2}{2^T} \quad (13)$$

and estimate the values of other pixels according to the pixel values in (13) using a bilinear interpolation method. Bilinear interpolation method can be used to calculate and assign appropriate intensity values to pixels. Bilinear interpolation uses only the 4 nearest pixel values which are located in diagonal directions from a given pixel values in order to find the appropriate color intensity values of that pixel. Then, the interpolated pixels are reorganized as data sets with multiple-resolution[6] construction, and the data in each set are permuted and divided into a series of groups. Denote the interpolated pixel values of the k_{th} group at the t_{th} level as $r_k^{(t)}(1), r_k^{(t)}(2), \ldots, r_k^{(t)}\left(L^{(t)}\right)$ and their corresponding original pixel values as $p_k^{(t)}(1), p_k^{(t)}(2), \ldots, p_k^{(t)}\left(L^{(t)}\right)$.

The errors of interpolated values are

$$\Delta p_k^{(t)}(l) = p_k^{(t)}(l) - r_k^{(t)}(l), 1 \leq k \leq L^{(t)}, 1 \leq k \leq K^{(t)}, \tau \leq t \leq T \quad (14)$$

After performing decryption [7] and decoding the original content BG is reconstructed this provides rough information [8].

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

$$\hat{r}_k^{(t)}(l) = mod\left[r_k^{(t)}(l) + e_k^{(t)}(l), 256\right], 1 \leq k \leq L^{(t)}, 1 \leq k \leq K^{(t)}, \tau \leq t \leq T \quad (15)$$

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

$$\Delta p_k^{(t)}(l) \equiv q_k^{(t)}(l) - \hat{r}_k^{(t)}(l) \bmod 256 \quad (16)$$

With the bit-streams $\mathrm{BS}^{(t)}(\tau \leq t \leq T)$, 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 [9] of the reconstructed image by updating the pixel values according to $c_k^{(t)}(l)$. The detailed procedure is as follows.

Step1:
For each group $\left[r_k^{(t)}(1), r_k^{(t)}(2), \ldots r_k^{(t)}\left(L^{(t)}\right)\right]$, calculate $\hat{r}_k^{(t)}(l)$and $\hat{C}_k^{(t)}(l)$ using (16) and (18)

$$\begin{bmatrix} \Delta C_k^{(t)}(1) \\ \Delta C_k^{(t)}(2) \\ \vdots \\ \Delta C_k^{(t)}(L^{(t)}) \end{bmatrix} = \mathbf{H} \cdot \begin{bmatrix} \Delta p_k^{(t)}(1) \\ \Delta p_k^{(t)}(2) \\ \vdots \\ \Delta p_k^{(t)}(L^{(t)}) \end{bmatrix} (17)$$

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

$$\begin{bmatrix} \Delta C_k^{(t)}(1) \\ \Delta C_k^{(t)}(2) \\ \vdots \\ \Delta C_k^{(t)}(L^{(t)}) \end{bmatrix} = \mathbf{H} \cdot \begin{bmatrix} q_k^{(t)}(1) \\ q_k^{(t)}(2) \\ \vdots \\ q_k^{(t)}(L^{(t)}) \end{bmatrix} - \mathbf{H} \cdot \begin{bmatrix} \hat{r}_k^{(t)}(1) \\ \hat{r}_k^{(t)}(2) \\ \vdots \\ \hat{r}_k^{(t)}(L^{(t)}) \end{bmatrix} \bmod 256 \quad (18)$$

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 [4] [5]. Denoting

$$\begin{bmatrix} \widehat{C}_k^{(t)}(1) \\ \widehat{C}_k^{(t)}(2) \\ \vdots \\ \widehat{C}_k^{(t)}(L^{(t)}) \end{bmatrix} = \mathbf{H} \cdot \begin{bmatrix} \hat{r}_k^{(t)}(1) \\ \hat{r}_k^{(t)}(2) \\ \vdots \\ \hat{r}_k^{(t)}(L^{(t)}) \end{bmatrix} (19)$$

We have

$$\Delta C_k^{(t)}(l) \equiv C_k^{(t)}(l) - \hat{C}_k^{(t)}(l) \bmod 256 \quad (20)$$

Step2:
Then calculate,

$$D_k^{(t)}(l) = \mathrm{mod}\left[c_k^{(t)}(l) \cdot \Delta^{(t)} + \Delta^{(t)}/2 - \hat{C}_k^{(t)}(l), 256\right] (21)$$

$$\widetilde{D}_k^{(t)}(l) = \begin{cases} D_k(l), if\ D_k(l) < 128 \\ D_k(l) - 256, if\ D_k(l) \geq 128 \end{cases} (22)$$

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

$$\begin{bmatrix} \bar{r}_k^{(t)}(1) \\ \bar{r}_k^{(t)}(2) \\ \vdots \\ \bar{r}_k^{(t)}(L) \end{bmatrix} = \begin{bmatrix} r_k^{(t)}(1) \\ r_k^{(t)}(2) \\ \vdots \\ r_k^{(t)}(L) \end{bmatrix} + \frac{H'}{L^{(t)}} \cdot \begin{bmatrix} \widetilde{D}_k^{(t)}(1) \\ \widetilde{D}_k^{(t)}(2) \\ \vdots \\ \widetilde{D}_k^{(t)}(L^{(t)}) \end{bmatrix} (23)$$

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

$$\tilde{r}_k^{(t)}(l) = \begin{cases} 0, if\ \bar{r}_k^{(t)}(l) < 0 \\ \bar{r}_k^{(t)}(l), if\ 0 \leq \bar{r}_k^{(t)}(l) \leq 255 \\ 255, if\ \bar{r}_k^{(t)}(l) \geq 255 \end{cases}$$

Step3:
Calculate the average energy of difference due to the modification as follows:

$$D = \frac{\sum_{t=\tau}^{T} \sum_{k=1}^{K^{(t)}} \sum_{l=1}^{L^{(t)}} [\tilde{r}_k^{(t)}(l) - r_k^{(t)}(l)]^2}{\sum_{t=\tau}^{T} 3N/4^t} (24)$$

If $D$is not less than a given threshold of 0.10, for each pixel $\tilde{r}_k^{(t)}(l)$,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 Step1. Otherwise, terminate the iteration, and output the image as a final reconstructed result.

In the iterative procedure, while the decrypted pixels $p^{(T)}(m,n)$are used to give an initial estimation of other pixels, the values of $c_k^{(t)}(l)$, in bit-streams $\mathrm{BS}^{(t)}$ provide more detailed information to produce the final reconstructed result with satisfactory quality. 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 the Error accumulation in a group, so that $\widetilde{D}_k^{(t)}(l)$in (22) may be 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 sub image with lower resolution is weaker.

For instance,$L^{(1)} = 24, L^{(2)} = 8$ and $L^{(3)} = 4$ for $T = 3$

Furthermore, in Step 3, the value of each pixel is assigned as the average of its four neighbors to further approach its original value. Although the estimate of a certain pixel may be very different from its original value, the updating operation in Step 3 can effectively lower the error on the pixel since its neighbors are probably modified well. At last, we terminate the iterative procedure when the reconstruction quality is not improved further. Here, the small threshold of 0.10 ensures the convergence of iterative procedure.

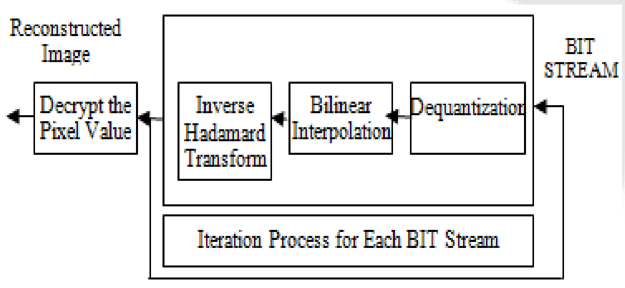

Figure 4.1: Image Reconstruction

5. Design Approach

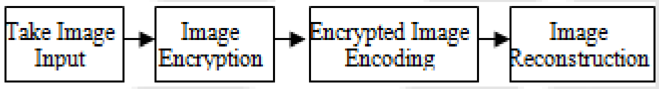

Figure 5.1: Block diagram of proposed design

6. Simulated Results and Discussion

Consider an image of Lena that is sized $to\ 256 \times 256$ and used as the original images in the experiment. We let $T = 3$ and encoded the encrypted images using $M = 24, L^{(1)} = 24, L^{(2)} = 8$and$L^{(3)} = 4$ to produce the bit-streams $\mathrm{BS}^{(\mathrm{B})}, \mathrm{BS}^{(3)}, \mathrm{BS}^{(2)}$and $\mathrm{BS}^{(1)}$. In this case, the total compression ratio $R_C = 0.318$. Fig.5 gives the reconstructed Lena using$\{\mathrm{BS}^{(\mathrm{B})}\}$,, $\{\mathrm{BS}^{(\mathrm{B})}, \{\mathrm{BS}^{(3)}\}$, $\{\mathrm{BS}^{(\mathrm{B})}, \mathrm{BS}^{(3)}, \mathrm{BS}^{(2)}\}$and $\{\mathrm{BS}^{(\mathrm{B})}, \mathrm{BS}^{(3)}, \mathrm{BS}^{(2)}, \mathrm{BS}^{(1)}\}$ respectively.

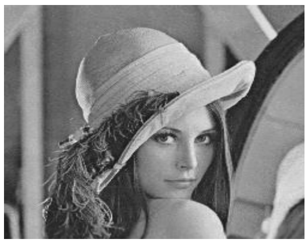

Figure 6.1: Original Image

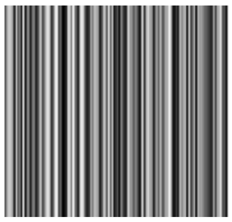

Figure 6.2: Encrypted Image

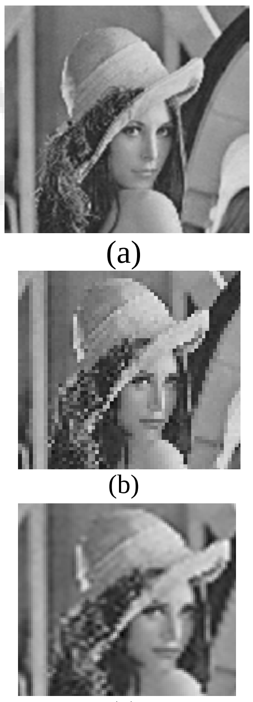

Figure 6.3: (a, b, c)Reconstructed Lena using final image bit-stream 3,bit-stream 2 and bit-stream 1 $\{Bs_{(B)},Bs_{(3)},Bs_{(2)},Bs_{(1)}\}$,$\{Bs_{(B)},Bs_{(3)},Bs_{(2)}\}$,$\{Bs_{(B)}, Bs_{(3)}\}$

Table 1: PSNR value of input image with different M

Input Image	*Integer value(M)*	*PSNR(db)*
Lena	22	35.1
	24	37.8
	26	39.2

Table1 lists the compression ratios; the PSNR in reconstructed results with respect to different M when $T = 3$, $L^{(3)} = 4, L^{(2)} = 8$ and $L^{(1)} = 4$ were used for images of Lena. All the encryption, encoding and reconstruction procedures were finished in several seconds by a personal computer. When the value of M is larger, the compression ratio is higher, and the reconstruction qualities are better since $c_k^{(t)}(l)$, providing more detailed information. As there is less texture/edge content in Lena, the quality of reconstructed Lena is better. In addition, the larger $L^{(t)}$ corresponds to the lower compression ratio and more detailed $c_k^{(t)}(l)$.When we changed $(L^{(3)}, L^{(2)}, L^{(1)})$ from (4, 8, 12) to (4, 12, 32), the compression ratio decreased from 0.318 to 0.283, and the value of PSNR in reconstructed Lena was 37.8.

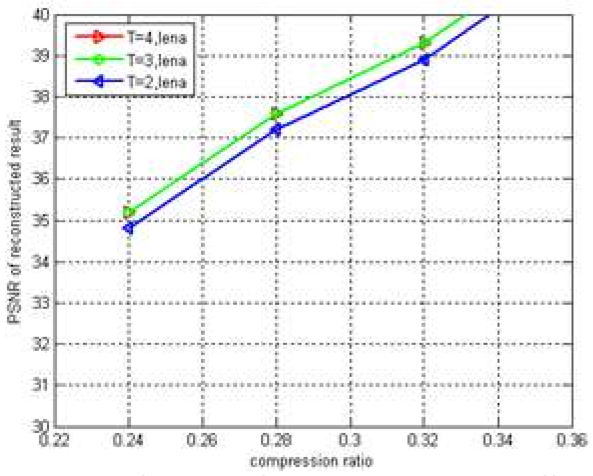

Figure 6.4: Performance of proposed scheme with different T

For Lena, the larger $L^{(t)}$ was helpful to uniformly distribute the errors on pixels into the Hadamard coefficients, and most of $\Delta C_k^{(t)}(l)$ still fell into [-128,128], so that the quality of reconstructed result was better.Fig6.4 gives the $R_c - PSNR_I$ curves with different values of T. When an encrypted image is decomposed within more levels, more data are involved in quantization and compression; therefore, the $R_c - PSNR_I$ performance is better, and more iteration for image reconstruction are required. It is also shown that the performance improvement is not significant when using a higher T more than 3. However, a decoder with higher computation complexity and the decoder's feedback for sending rate of each bit plane are required in the method extended from [5]. That means the proposed scheme is more suitable for real-time decompression and some scenarios without feedback channel.

7. Conclusion

This work proposed a novel idea for encoding an encrypted image and designed a practical scheme made up of image encryption, lossy compression, and iterative reconstruction. The original image is encrypted by moduolo-256 addition of pseudorandom numbers, and then compressed by discarding the excessively rough and fine information of coefficients in the transform domain. 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. In general, the higher the compression ratio and the smoother the original image, the better the quality of the reconstructed image. In encryption technique, Scalable coding algorithm includes more number of advantages like amount of secrecy (key size) determines amount of labor and scalable coding technique has set of keys and the enciphering algorithm is simple. At the receiver side, while the sub-image is decrypted to produce an approximate image, the quantized data of Hadamard coefficients can provide more detailed information for image reconstruction. Since the bit-streams are generated with a multiple-resolution construction, the principal content with higher resolution can be obtained when more bit-streams are received. The Lossless compression and scalable coding for encrypted image with better performance deserves further investigation.

8. Future Scope

In Order to reduce the sizes 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 is its simple underlying concepts and ease of implementation.

References

[1] 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.
[2] M. Kuribayashi and H. Tanaka, "Finger printing protocol for images based on additive homomorphic property," IEEE Trans.
[3] Image Process., vol. 14, no. 12, pp. 2129–2139, Dec. 2005.
[4] 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.
[5] R. Lazzeretti and M. Barni, "Lossless compression of encrypted greyleveland color images," in Proc. 16th EUSIPCO, Lausanne, Switzerland,Aug. 2008 [Online]. Available.
[6] W. Liu, W. Zeng, L. Dong, and Q. Yao, "Efficient compression of encrypted gray scale images," IEEE Trans. Signal Process, vol. 19, no. 4, pp. 1097–1102, Apr. 2010. [6] A. Kumar and A. Makur, "Lossy compression of encrypted image by compressing sensing technique," in Proc. IEEE TENCON, 2009, pp.1–6.
[7] X. Zhang, "Lossy compression and iterative reconstruction for encrypted image," IEEE Trans. Inf. Forensics Security, vol. 6, no. 1, pp.53–58, Mar. 2011.
[8] Bilgin, P. J. Sementilli, F. Sheng, and M. W. Marcellin, "Scalable image coding using reversible integer wavelet transforms," IEEE Trans.Image Process., vol. 9, no. 11, pp. 1972–1977, Nov. 2000.
[9] D. Taubman, "High performance scalable image compression with EBCOT," IEEE Trans. Image Process., vol. 9, no. 7, pp. 1158–1170,Jul. 2000

Author Profile

N. Nagarajakumar received his B-tech degree from RGMCET in the year of 2004 in Electronics and Communication and pursued M.Tech Degree from RGMCET in the year of 2007 in the specialization Digital Systems and Computer Electronics. His research interests are Digital Image Processing and Signal Processing.

M. Sucharitha received her B-tech degree from Sri Krishna Devaraya College of Engineering And Technology in Electronics and Communication and pursuing M. Tech Degree in Rajeev Gandhi Memorial College of Engineering And Technology in the specialization Digital Systems and Computer Electronics. Her research interests are Digital Image Processing and signal processing.