function in (1) only scrambles the pixel values, but does not shuffle the pixel locations. This means geometric information of the pixels is still preserved, which is leveraged by the downsampling operation. After the downsampling, each sub-image is encoded independently using Slepian-Wolf codes, and the resulting syndrome bits are transmitted from the lowest resolution to the highest.

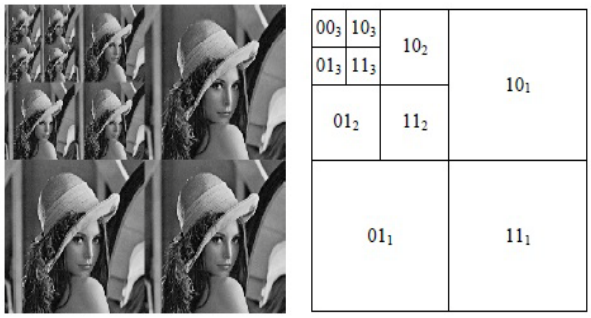

Figure 2: Sub Images and its codes.

The Real-world image data is highly non-stationary, hence it is desired to have the interpolation adapted to the local context. For example, for a pixel on an edge, it is preferable to interpolate along the edge orientation. Similar efforts can be found in conventional lossless image compression, where the median edge detector (MED) and the gradient adaptive predictor (GAP) are two successful context adaptive predictors. However, they process the pixels in a raster-scanning order, thus cannot be directly applied to our scheme.

5. Image Encryption Via prediction Error Clustering and Random Permutation

If From the perspective of the whole ETC system, the design of the encryption algorithm should simultaneously consider the security and the ease of compressing the encrypted data. To this end, we propose an image encryption scheme operated over the prediction error domain. The schematic diagram of this image encryption method is depicted in Figure 3.

For each pixel I(i,j) of the image I to be encrypted, a prediction I(i,j) is first made by using an image predictor, e.g. GAP or MED, according to its causal surroundings. In this work, the GAP is adopted due to its excellent de-correlation capability. The prediction result I(i, j) can be further refined to ~I(i,j) through a context-adaptive, feedback mechanism.

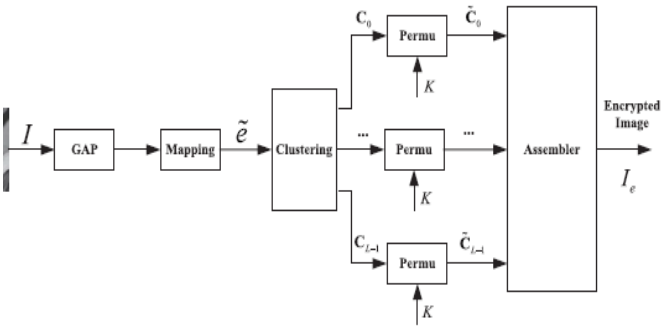

Figure 3. Schematic diagram of image encryption.

Consequently, the prediction error associated with I(i, j) can be computed by:

$$e_{(i,\,j)} = [I_{(i,\,j)}] - [\sim I_{(i,\,j)}] \quad (1)$$

Although for 8-bit images, the prediction error $e_{(i,\,j)}$ can potentially take any values in the range [−255, 255], it can be mapped into the range [0, 255], by considering the fact that the predicted value ˜ Ii, j is available at the decoder side. From (1), we know that $e_{(i,\,j)}$ must fall into the interval [− ~Ii, j , 255− (~ $I_{(i,\,j)}$)], which only contains 256 distinct values. More specifically, if ~Ii, j ≤ 128, we rearrange the possible prediction errors in the order 0,+1,−1, . . . ,+ ~Ii, j ,− ~Ii, j , ~Ii, j + 1, ~Ii, j + 2, . . . , 255 − ~Ii, j , each of which is sequentially mapped to a value between 0 to 255. If ˜ Ii, j > 128, a similar mapping could be applied. Note that, in order to reverse the above mapping, the predicted value ~Ii, j needs to be known. In the sequel, let us denote the mapped prediction error by $\sim e_{(i,\,j)}$, which takes values in the range [0, 255].

The algorithmic procedure of performing the image encryption is then given as follows:
Step 1: Compute all the mapped prediction errors ~e(i,j) of the whole image I .
Step 2: Divide all the prediction errors into L clusters Ck, for 0 ≤ k ≤ L − 1, where k is determined by (5), and each Ck is formed by concatenating the mapped prediction errors in a raster-scan order.
Step 3: Reshape the prediction errors in each Ck into a 2-D block having four columns and |Ck |/4 rows, where |Ck| denotes the number of prediction errors in Ck.
Step 4: Perform two key-driven cyclical shift operations to each resulting prediction error block, and read out the data in raster-scan order to obtain the permuted cluster ˜Ck.

With all the Ck , the decoding of the pixel values can be performed in a raster-scan order. For each location (i, j), the associated error energy estimator Δ (i, j) and the predicted value ~Ii, j can be calculated from the causal surroundings that have already been decoded. Given Δ(i,j), the corresponding cluster index k can be determined by. The first 'unused' prediction error in the kth cluster is selected as ~(ei,j), which will be used to derive e(i, j) according to ~I(i,j) and the mapping rule. Afterwards, a 'used' flag will be attached to the processed prediction error. The reconstructed pixel value can then be computed by:

$$I_{i,\,j} = \tilde{}I_{i,\,j} + e(i,\,j) \quad (2)$$

As the predicted value ~Ii, j and the error energy estimator Δ(i,j) are both based on the causal surroundings, the decoder can get the exactly same prediction ~Ii, j . In addition, in the case of lossless compression, no distortion occurs on the prediction error e(i,j) , which implies ^I(i,j) = I(i, j) , i.e., error-free decoding is achieved.

6. High Resolution Image Encryption & Reconstruction Using Scalable Codes

New scheme of the scalable coding for encrypted gray images is proposed in this paper. There are lots of work on

the scalable coding on normal images, but scalable coding of the encrypted images is not proposed yet. In this paper Akash Raj proposed new scheme [5] that keeps the pixel secrete from the attacker and they cannot obtain the original information of an original image. Also he uses the new concept pseudorandom number generator (PRNG) that assumed same at the both owner and decoder side. The content sender generates this 8N bit sequence here N is the total number of pixels in the image. First, sender decomposes the encrypted image into a series of subimages and datasets with different resolution construction. Downsampling the sample image at the t^{th} level

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

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

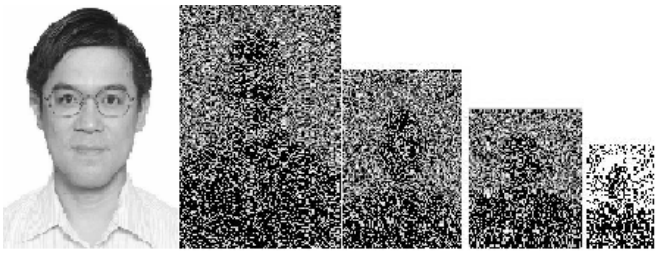

Figure 4:. Encrypted and compressed images

To perform encryption of the subimages Hadamard matrix is generated. The encoder transmit the bitstreams with an order of {BG, $BS^{(T)}$, $BS^{(T-1)}$ ….., $BS^{(1)}$}. If the channel bandwidth is limited then latter bitstreams may discarded. At the receiving side, by using bitstreams and secret key receiver can reconstruct the content of the original image and resolution of the reconstructed image is dependent on the number of bitstreams received.

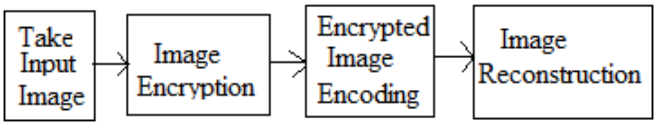

Figure 5: Block Diagram

7. Analysis of all the Reviewed Techniques

All the above analyzed methods are up to the mark but any algorithm is said to be good quality algorithm if its complexity is less and efficiency is low. The CALIC [1] algorithm has two modes of operation that increases its complexity and doubles the processing overhead. The LOCO-I[2] algorithm is again a complex part and time consuming process as the the image is processed sample by sample in a pre defined order. The Resolution progressive compression of encrypted images [3] samples an image into sub images 00,01,10,11 and then further down sampling of these images takes place. The process again becomes too crazy and difficult to process. All the algorithms process and produce the efficient output, but the algorithmic parameters are neglected which we need to work on.

References

[1] Xiaolin Wu, Nasir Menon, “CALIC – A Context based adaptive lossless image codec,” ICASSP-96. Conference Proceedings, 1996 IEEE International Conference on pp. 1890 - 1893 (Volume:4).
[2] M. J. Weinberger, G. Seroussi, and G. Sapiro, “The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS,” IEEE Trans. Imag. Process., vol. 9, no. 8, pp. 1309–1324, Aug. 2000.
[3] W. Liu, W. J. Zeng, L. Dong, and Q. M. Yao, “Efficient compression of encrypted grayscale images,” IEEE Trans. Imag. Process., vol. 19, no. 4, pp. 1097–1102, Apr. 2010.
[4] Jiantao Zhou, Xianming Liu, Oscar C. Au, Yuan Yan Tang, “Designing an Efficient Image Encryption-Then-Compression System via Prediction Error Clustering and Random Permutation”, Ieee Transactions On Information Forensics And Security, Vol. 9, No. 1, January 2014
[5] Akash Raj, “High Resolution Image Encryption & Reconstruction Using Scalable Codes”, International Journal of Engineering Research and Applications (IJERA), Vol. 3, Issue 2, March -April 2013, pp.444-450

Author Profile

Ganesh Lamkhade received the B.E. (Computer Engineering) from University of Pune in 2011 and pursuing M.E. degree in Computer Engineering from Institute of Knowledge College of Engineering, Savitribai Phule Pune University, Pimple Jagtap pune, Maharashtra, India in 2014-15. During 2010-2011, he worked on Mobile Application as project for fulfillment of his Bachelor’s Degree. Also in 2012 worked as software developer.

Mr. Ajay Kumar Gupta is working as Assistant Professor **in** Institute of Knowledge of COE, Savitribai Phule Pune University, Pimple Jagtap, Pune, Maharashtra, India.