International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

A Survey on Efficient ETC (Encryption-Then-Compression) Techniques for Image Data Security

Ganesh Lamkhade[1], **Ajay Kumar Gupta**[2]

[1]Institute of Knowledge COE, Savitribai Phule Pune University, Pimple Jagtap, Pune, Maharashtra, India

[2]Assistant Professor, Institute of Knowledge COE, Savitribai Phule Pune University, Pimple Jagtap, Pune, Maharashtra, India

Abstract: ***As of now, many image encryption and compression techniques have been emerged to provide security to the data in the images, but still image encryption is to be carried out prior to the image compression. This creates a problem of how to synchronize both these operations so that compression of the Encrypted images is done with maximum efficiency. In this paper we analyze various techniques emerged for Encryption then Compression process and ultimately design a new and efficient technique that provides maximum efficiency in terms of encryption as well as in terms of compression.***

Keywords: Encryption, Compression, Error Prediction, Security, Image Processing.

1. Introduction

As the world has been totally digitized, along with digitalism, use of multimedia has also rapidly increased. But with sudden increase in use of multimedia has raised an important issue of securing the multimedia data as these data prone to being getting hacked or leaked due to its availability. As the multimedia data is transmitted over networks on large scale, we need to have a reliable technique to prevent data getting leaked or attacked. The security mechanism should typically be a reliable method to protect the images as well as videos with same potential. Good encryption makes a source look completely random, traditional algorithms are unable to compress encrypted data. For this reason, traditional systems make sure to compress before they encrypt. We are using the concept of public key encryption, for the encryption and decryption of image. In this public key‘s of sender and receiver is known to both but private key‘s are kept secret. Neither the security nor the compression efficiency will be sacrificed by performing compression in the encrypted domain. Many techniques require the complex computation to be done to encrypt the images and videos but it takes a larger overhead of computing values for non sensitive data too. So the solution to this problem is to do selective encryption. Government, military and private business organizations make use of great deal of confidential images about their patients (in Hospitals), geographical areas (in research processes), enemy positions (in defence and military operations) product, financial-status. Most of this information is now captured, processed and stored on electronic computers and transmitted across the vast geographical areas network to other computer, if these confidential images about enemy positions ,patient ,and geographical areas fall into the maliciously behaving hands, than such a breach of security could lead to lots of war, wrong treatment etc. So Protecting confidential images is an ethical and legal requirement. We store information in computer system in the form of files. File is considered as a basic entity for keeping the information in electronic form. Therefore the problem of securing image data or information on computer system can be defined as the problem of securing file data. It is worldwide accepted fact that it's a great challenge for all of us for securing file data in today‘s computing environment. **C**onsider an application scenario in which a content owner A wants to securely and efficiently transmit an image I to a recipient B, via an un-trusted third party channel provider C. Conventionally, this could be done as follows: A with the intention to reduce the data size, first compresses I into I_c, and then encrypts I_c into I_e using an encryption function EN (∗), where N denotes the secret key. The encrypted data I_e is then passed to C, who simply forwards it to B. Upon receiving I_e, B sequentially performs decryption and decompression to get a reconstructed or the original image I. Even though the above Compression-then-Encryption (CTE) paradigm meets the requirements in many secure transmission scenarios, the order of applying the compression and encryption needs to be reversed in some other situations. A Being the data owner A is concerned with protecting the data privacy of the images through encryption, as he/she is least bothered of reducing the space of the data and highly concerned about the data security.

So A avoids doing compression of the data and focuses on doing the Data encryption and save his/her limited computational resources by not doing the compression before encryption. This is a specific case where A is using resource deprived device to do the process. On the other hand, C being the network provider is keen to make maximum resource utilization, has an overriding interest in compressing all the network traffic so as to maximize the network utilization. It is therefore much desired if the compression task can be delegated by C, who typically has abundant computational resources. A big challenge within such Encryption-then-Compression (ETC) framework is that compression has to be conducted in the encrypted domain, as C does not access to the secret key K. This type of ETC system is demonstrated in Figure 1.

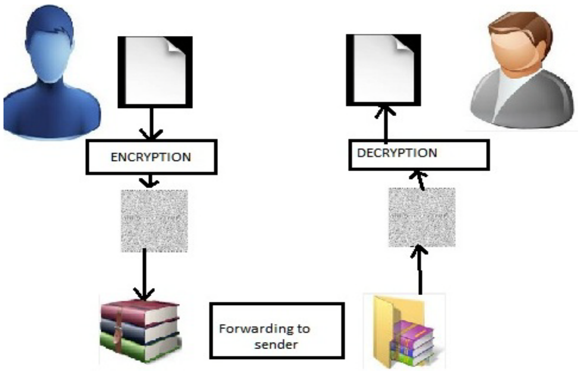

Figure 1: Encryption Then Compression.

The possibility of processing encrypted signals directly in the encrypted domain has been receiving increasing attention in recent years. At the first glance, it seems to be infeasible for C to compress the encrypted data, since no signal structure can be exploited to enable a traditional compressor. Various researches has been carried out on this ETC systems and results have been evaluated. We further see how CALIC and Other techniques carry out the ETC operation and the experimental analysis is done on the data sets provided.

2. CALIC Encryption Then Compression Technique

CALIC encodes and decodes images in single raster scan method. The encoding procedure takes prediction template that only involve previous two scanned lines of code. So the encoding and decoding process just needs the double buffer to hold the two rows of the lines. In predictive coding such as the benchmark codec, over 50% of the computations come from the entropy coding part, assuming that the adaptive AC is adopted. This implies that if Alice has to compress the prediction errors via adaptive AC, the computational burden will at least be doubled. CALIC operates in two modes: binary mode and continuous tone method. Binary mode is the situation where in current locality of the input image doesn't have the more than two distinct intensities values, and so designed for a more general class of images than the general class of black and white images.

The selection between these two modes is performed on the fly based on context:

2.1 Continuous Tone Mode

In continuous-tone mode, the neighborhood of the pixel to be encoded has more than two distinct grey levels. In thismode CALIC algorithm performs four operations:
a) initial prediction,
b) context classification,
c) error feedback, and
d) entropy encoding.

The initial prediction is obtained for the pixel to be encoded using Gradient Adjusted Predictor (GAP). GAP is a simple non-linear predictor that utilizes gradients at pixel neighborhood. In context classification, each pixel is classified to one of the 576 predefined contexts. The context selection is based on comparing the value of the initial prediction with the pixel neighborhood's values. For each context, CALIC assumes that the GAP predictor is consistently repeating a similar prediction error. To compensate for this error, CALIC incorporates an error feedback stage, at which a bias value is added to the initial prediction. This bias value is the expectation of the prediction errors at the pixel's context.

2.2 Binary Mode

Binary mode is considered when a pixel's neighborhood has no more than two distinct grey levels. In such case, it may be suitable to encode a pixel's value directly from neighboring pixel values. However, when the pixel to be encoded has a different grey level than any of neighboring pixels, CALIC triggers an escape sequence that switches the algorithm to continuous-tone mode. In this case the pixel will be treated as if it was in a continuous-tone region, despite the fact that the pixel's neighborhood is actually discrete. As a result of this, the pixel will be encoded using the GAP initial predictor and the error feedback scheme.

Table 1: CALCI Performance on various images

Image Class	*Calci with conditional error feedback*
Geographical	1.58:1
Graphical	4.58:1
Medical	2.46:1
Natural	1.86:1
Binary	8.87:1

3. LOCO-I Lossless Image Compression Technique

A LOCO-I (LOw COmplexity LOssless COmpression for Images) is the algorithm at the core of the new ISO/ITU standard for lossless and near-lossless compression of continuous-tone images, JPEG-LS. Lossless data compression schemes often consist of two distinct and independent components: modeling and coding. The modeling part can be formulated as an inductive inference problem, in which the data (e.g., an image) is observed sample by sample in some pre-defined order (e.g., raster-scan, which will be the assumed order for images in the sequel).

4. Resolution Progressive Compression of Encrypted Images

The The encoder gets the ciphertext Y and decomposes it into four sub-images, namely, the 00, 01, 10 and 11 sub-images. Each sub-image is a downsampled-by-two version of the encrypted image. The name of a sub-image denotes the horizontal and vertical offsets of the downsampling. The 00 sub-image is further down sampled to create multiple resolution levels. We use 00_n to represent the 00 sub-image in the n-th resolution level. The 00n sub-image can be losslessly synthesized from the 00_{n+1}, 01_{n+1}, 10_{n+1} and 11_{n+1} sub-images. An example of the decomposition is illustrated in Figure 2. Here the image is supposed to be an encrypted one. We show it in plaintext just for a better illustration. Meanwhile, we would like to point out that the stream cipher 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.

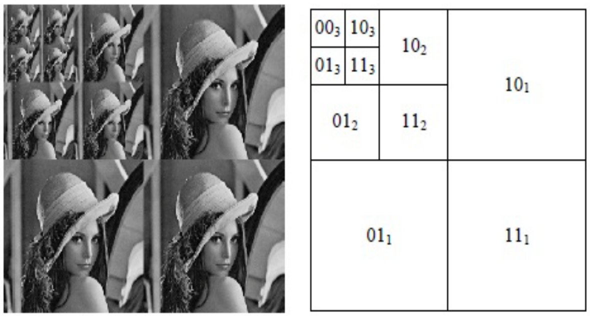

Figure 2: Sub Images and its codes.

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.

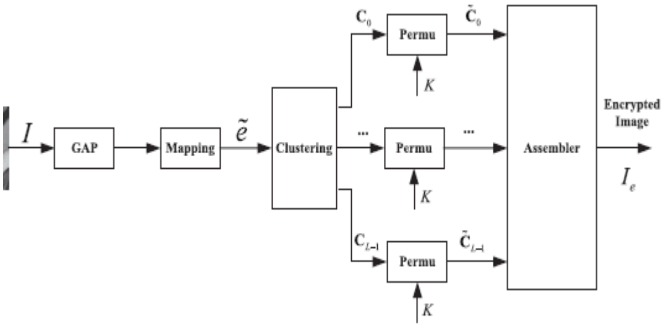

Figure 3. Schematic diagram of image encryption.

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

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

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 ~$e_{(i,\ j)}$, 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:

$$I_{i,\,j} = {}^{\sim}I_{i,\,j} + e_{(i,\,j)}\ (2)$$

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 tth 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.

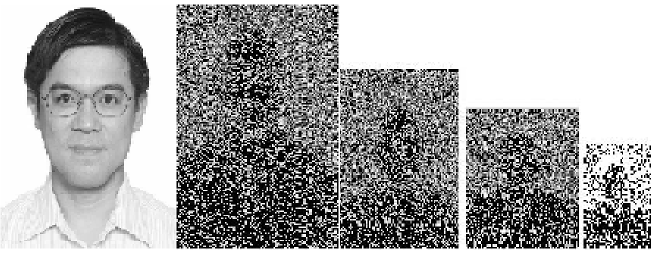

Figure 4:. Encrypted and compressed images

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.

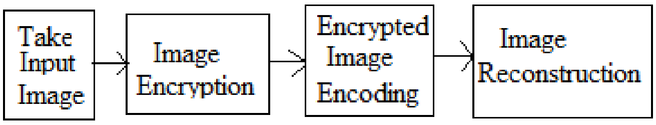

Figure 5: Block Diagram

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.