Int'l Journal of Computing, Communications & Instrumentation Engg. (IJCCIE) Vol. 3, Issue 1 (2016) ISSN 2349-1469 EISSN 2349-1477

FPGA Implementation of High Security ETC System for Image

Suneera.H, and Rahul.M.Nair

***Abstract*— Image Encryption is the process of transforming information using an algorithm called cipher to make it unintelligible to anyone except those possessing special knowledge, usually referred to as a key. The result of the process is encrypted image referred to as cipher-image. This has led to the problem of how to design a pair of image encryption and compression algorithms such that compressing the encrypted images can still be efficiently performed. Here we design a highly efficient image encryption-then-compression (ETC) system, where both lossless and lossy compression are considered. The proposed image encryption scheme operated in the prediction error domain is shown to be able to provide a reasonably high level of security. In this paper data compression uses Run Length Encoding (RLE) technique because this method has the ability to generate an exact output with low power consumption and reduced time delay. In contrast, most of the existing ETC solutions induce significant penalty on the compression efficiency, so I was done a literature survey for the existing systems and proposed scheme. Here I proposed to implement the system in SPARTAN 6 FPGA and simulated using Xilinx ISE 14.7 software.**

***Keywords*— Compression, Cryptography, Encryption, Security**

I. INTRODUCTION

Cryptography is defined as the science and study of secret writing and anxieties the ways in which communications and data can be encoded to prevent confession of their contents. Image Encryption can be used to protect images at rest, such as images on computers and storage devices in a situation which personal records being exposed through loss or theft of laptops or backup drives. Image encrypting at rest helps protect them from being uncovered and shared. Encryption is also used to protect data in transit, for example data being transferred via networks (Internet, e-commerce), mobile telephones, wireless systems, Bluetooth devices and so on.

In order to facilitate secret communication, image encryption has found a significant place in both public and private services such as military surveillance, satellite information systems, health-care, meteorology, confidential video conferencing, personal photograph album, internet banking transactions, multimedia systems, telemedicine, and medical imaging systems [4]-[9]. A cryptographic algorithm, or cipher, is a mathematical function used in the encryption and decryption process. While considering an image processing system the compression is also places a vital role in the system. With the ever-increasing growth of multimedia applications, security is an important issue in communication and storage of images, and Encryption is a common technique to uphold image security. Image encryption techniques try to convert original image to another image that is hard to understand; to keep the image confidential between users, in other word, it is essential that nobody could get to know the content without a key for decryption. A sort description of security goals will shows in figure1.

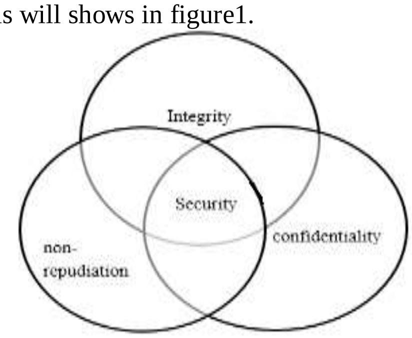

Fig. 1. Security Goals

In a communication network the order of these processes are, compression is followed by encryption operation and it will meets the requirements in secure transmission scenarios. The order of these operations needs to be reversed in some situations like where the content owner is interested only in protecting the privacy of the content, the owner has no incentive to use his limited computational resources to run a compression algorithm and to maximize the network utilization, the network owner is forced to compress the data after it has been encrypted.

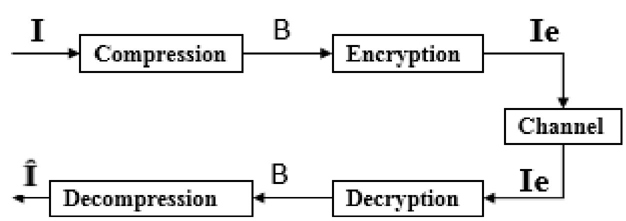

Fig. 2.Conventional Compression Then Encryption System.

Conventional compression-then-Encryption (CTE) system is shown in Figure2. Transmitter wants to securely and efficiently transmit an image I to a receiver, via an untrusted

Manuscript received Feb. 20, 2016. This work is done as part of Post-graduation Thesis work.

Suneera.H PG Scholar with Nehru College of Engineering and Research Centre, Pampady, Trichur, Kerala, India

Rahul .M .Nair with Assistant Professor, Department of Electronics and communication Engineering. Nehru College of Engineering and Research Centre, Pampady, Kerala.

channel provider. Transmitter first compress I into B and then encrypts B into Ie using an encryption function EK(.),where K denotes the secret key. The encrypted data Ie is then passed to channel, channel simply forwards it to receiver, upon receiving I_e receiver sequentially performs decryption and decompression to get a reconstruction image Î.

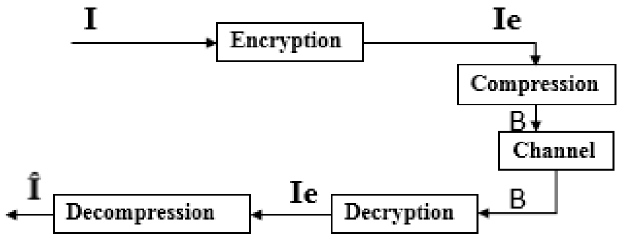

Fig.3. Encryption-then-Compression (ETC) system.

The proposed model of Compression-Then-Encryption (CTE) meets the requirements in many secure transmission scenarios, in order to apply the compression and encryption needs to be reversed in some other situations. A transmitter is always interested in protecting the privacy of the image data through encryption also transmitter has no incentive to compress data, and hence will not use available limited computational resource to run a compression algorithm before encrypting the data. In figure3 shows the proposed Encryption-Then-Compression (ETC), a data can transmitter wants to securely and efficiently transmit an image I to a receiver, via an untrusted channel provider. Transmitter first encryption I into I_e and then encrypts B into I_e using an encryption function EK (.), where K denotes the secret key. The encrypted data I_e is compressed then passed to channel, simply forwards it to receiver, upon receiving I_e receiver sequentially performs decompression and decryption to get a reconstruction image Î.A challenge within such Encryption-then-Compression (ETC) framework is that compression has to be conducted in the encrypted domain. The possibility of processing encrypted signals directly in the encrypted domain has been receiving increasing attention. At the first glance, it seems to be infeasible for channel provider to compress the encrypted data, since no signal structure can be exploited to enable a traditional compressor.

When using GAP instead of linear transformation, GAP is not reversible. So we use the same data as in encryption side GAP output [1]-[5]. That means there is no need for encryption. The main issue is that was designed in Matlab or Other software methods. So the portability is not easy, we need a system. When suggesting DCT method The disadvantage of this method is Most of the computation time required to transform, quantize, de-quantize, and reconstruct an image is spent on forward and inverse DCT calculations [3]. Because these transforms are applied to blocks, the time required is proportional to the size of the image. These times are much longer than for comparable functions written in a low-level language such as C. Size of the image get increases after decryption. The primary focus of this work is on the practical design of a pair of image encryption and compression schemes, in such a way that compressing the encrypted images is almost equally efficient as compressing their original, unencrypted counterparts [2].

The primary focus of this work is on the practical design of a pair of image encryption and compression schemes, in such a way that compressing the encrypted images is almost equally efficient as compressing their original, unencrypted counterparts. There for high level of security needs to be ensured. All the existing system or work related to this field is completely software or DSP related. So the portability of the particular system is not that much easy. Here the proposed method is worked on VLSI domain, it's a vast area. We are focusing on the front end design of proposed work. And implementing it on FPGA. This will improve the speed, efficiency also reduce the area, power, delay and cost. The FPGA implementation will helps to program it in future and we can add more features if necessary. In this work we are using HDL language, Verilog for frontend design.

The rest of this brief is organized as follows. Section II gives the details of the proposed ETC system, simulated output using Verilog codes is added in III section and section IV provides the conclusion.

II. Proposed System and Design Approach

In this section, we include the details of the three key components in our proposed Encryption Then Compression system.

A. Image encryption
B. Image compression
C. Decryption and decompression.

A. Image encryption

Encryption is one of the widely used techniques for data protection. Encryption can be applied to text, image, and video for data protection. In Image Encryption, image is converted from its original to other form so that information cannot be accessed from the image or data without decrypting the data. The original image is usually referred as plain data and the converted form is called cipher data. Encryption can be defined as the art of converting data into coded form which can be decode by intended receiver only who poses knowledge about the decryption of the ciphered data From the perspective of the whole ETC system, the design of the encryption algorithm should simultaneously onside the security and the ease of compressing the encrypted data. To this end, propose an image encryption scheme operated over the prediction error domain. We implement the Linear Transformation (LT) by using combinational logic, instead of existing methods like GAP, MED [5].

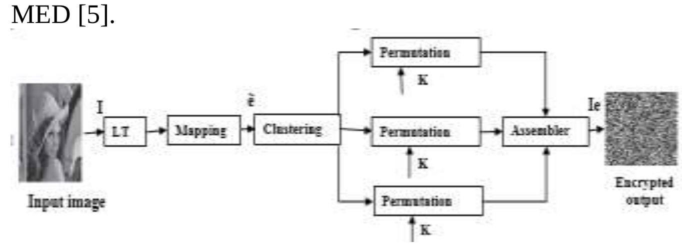

Fig. 4. Schematic diagram of image encryption.

The LT module has mainly 3 steps. The output will send to the mapping section, shown in Figure3. Here the given input is mapped in to other value. Then clustering module, here the input image matrix is divided in to small matrixes. After that the permutation is performed using the key (K). The 8 bit key is divided into 2 bit and distributed over the matrixes. After that these matrixes combined to get a matrix, which has the size of input image using assembler block. This is the working of encryption module. Further calculation and step by step methods are shown below. The prediction result $\bar{I}_{ij}$ can be further refined to $\hat{I}_{ij}$ through a context-adaptive, feedback mechanism. Consequently, the prediction error associated with I_{ij} can be computed by

$$e_{ij} = I_{ij} - \hat{I}_{ij} \qquad (1)$$

The generated error matrix is divided in to groups known as clusters and it is based on the selection of the parameter L needs to balance the security and the encryption complexity. The algorithmic procedure of performing the image encryption is then given as follows:

Step 1: Compute all the mapped prediction errors eij of the whole image I.

Step 2: Divide all the prediction errors into L clusters C_k, for $0 \leq k \leq L-1$, where k is determined by any key generation algorithm, and each C_k is formed by concatenating the mapped prediction errors in a raster-scan order.

Step 3: Reshape the prediction errors in each C_k into a 2-D block having four columns and $|C_k|/4$ rows, where |Ck| denotes the number of prediction errors in C_k.

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 I_k.

Step 5 The assembler concatenates all the permuted clusters C_k, for $0 \leq k \leq L$, and generates the final encrypted image.

Step 6: Pass Ie to Channel Provider together with the length of each cluster C_k, for $0 \leq k \leq L2$. The values of $| C_k |$ enable channel to divide Ie into L clusters correctly. In comparison with the file size of the encrypted data, the overhead induced by sending the length $| C_k |$ is negligible.

B. Image compression

Image Compression is concerned with minimizing the number of bit required to represent an image. The compression of the encrypted file Ie needs to be performed in the encrypted domain, as channel controller does not have access to the secret key K. In other words, Compression is the art of representing the information in a compact form rather than its original or uncompressed form. When data compression is used in a data transmission application, speed is the primary goal. Speed of transmission depends upon the number of bits sent, the time required for the encoder to generate the coded message and time required for the decoder to recover the original ensemble. In a data storage application, the degree of compression is the primary concern. Compression can be classed as either lossy or lossless. Here we use the encrypted data so the speed is improved compared with existing system. Some of the main techniques used in compression are the Huffman Coding, Run Length Encoding, Arithmetic Encoding and Dictionary Based Encoding. Here we use Run Length Encoding for compressing the data and send to the receiver through channel. Run length encoding (RLE) is a simple technique to compress digital data by representing successive runs of the same value in the data as the value followed by the count, rather than the original run of values. The lossless compression of Ie. Assisted by the side information C_k for $0 \leq k \leq L$, a de-assembler can be utilized to parse Ie into L segments C_0, C_1, and C_{L-1} in the exactly same way as that done at the encryption stage. An adaptive AC is then employed to losslessly encode each prediction error sequence $|C_k|$ into a binary bit stream B_k. Note that the generation of all B_k can be carried out in a parallel manner to improve the throughput. Eventually, an assembler concatenates all B_k to produce the final compressed and encrypted bit stream B, namely,

$$B = B_0 B_1 \ldots B_{L-1} \qquad (2)$$

The goal is to reduce the amount of data needed to be stored or transmitted.

C. Decryption and decompression

Upon receiving the compressed and encrypted bit stream B, we aims to recover the original image I. The schematic diagram demonstrating the procedure of sequential decryption and decompression is provided in Figure. 5. According to the side information jB_{kj}, the Channel provider divides B into L segments B_k for $0 \leq k \leq L - 1$, each of which is associated with a cluster of prediction errors. So here workout the same reversible process done in the compression and encryption, that is de compression and decryption. These two process is done by the receiver, as shown in Figure3. So the receiver know the key. The encrypted image first pixel permutated after that block permutation is applied to get back original image. The size of decryption key is equal to encryption key so large number of decryption key makes various malicious attacks redundant.

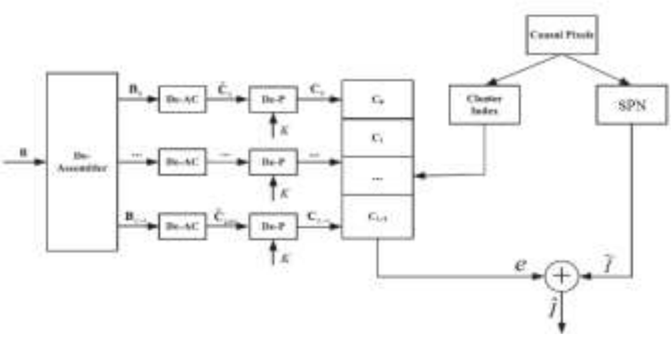

Fig. 5. Schematic diagram of sequential decryption and decompression

III. Simulated Results

Input Image - The pixel values of an image are the primary inputs of the entire system. These values will ranges from 0 to 255 and the pixels of an 8x8 grayscale image obtained using matlab. Then the value is stored in the memory as image input shown in figure 6.

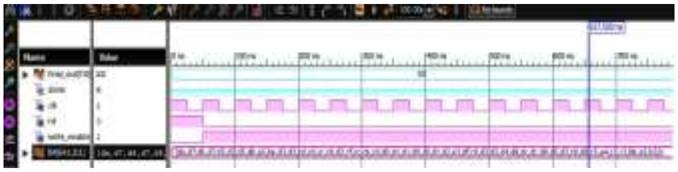

Fig. 6. Pixel values stored in the memory

These value is then encrypted using the process explained above. And the simulated output of encrypted image was shown in figure 7.

Fig. 7. Simulated output of Encryption.

These output will compressed and send to the receiver side through channel. Finally the receiver will get the actual image after decompression and decryption. The simulation results for gray-level images show that the proposed algorithm has great performance.

IV. CONCLUSION

The proposed method proves efficient ETC system using error clustering and random permutation. Our system efficacy can be closely compared to that of the lossless/lossy image coding, which receives original, unencrypted images as inputs. The literature survey sums up efficiency of the proposed system, over the existing techniques. The key challenge in this phase is to generate the pixel value as close as to the original pixel values at the transmitter side. Likewise a well-organized decryption and decompression technique can be implemented at the receiver section. The main advantage of the system is, this was performed in VLSI platform. So we can improve the efficiency, speed and reduce the power, area, and cost. This work can be extended to the future on an efficient system for colour image processing and video processing.

ACKNOWLEDGMENT

I express my sincere thanks to Dr. Raj Kumar, HOD, Dept. of ECE, Assistant Prof. Sajitha.A.S, Naslin Sithara, Irshad and project guide Rahul M Nair Dept. of ECE, who have contributed towards the development of the template.

REFERENCES

[1] Jiantao Zhou, Xianming Liu, Oscar C. Au, Yuan Yan Tang, (2014), "Designing an Efficient Image ncryption-Then-Compression System via Prediction Error Clustering and Random Permutation", *IEEE Transactions on Information Forensics and Security*, Vol. 9, No. 1. http://dx.doi.org/10.1109/TIFS.2013.2291625

[2] A. Alfalou, C. Brosseau, N. Abdallah, M. Jridi, "Simultaneous fusion, compression, and encryption of multiple images", *Optics express*, Vol. 19. 24, pp 24023-24029, Nov 2011. http://dx.doi.org/10.1364/OE.19.024023

[3] T. Bianchi, A. Piva, and M. Barni, "Encrypted domain DCT based on homomorphic cryptosystems," *EURASIP* J. Inf. Security, 2009, Article ID 716357. http://dx.doi.org/10.1155/2009/716357

[4] Bianchi, A. Piva, and M. Barni, (2009), "On the implementation of the discrete Fourier transform in the encrypted domain", *IEEE Trans. Inf.Forensics Security*, vol. 4, no. 1, pp. 86–97 http://dx.doi.org/10.1109/TIFS.2008.2011087

[5] Feng-Yang Hsieh1, Chia-Ming Wang, Chun-Chieh Lee And Kuo-Chin Fan, ,short paper , "A Lossless Image Coder Integrating Predictors And Block-Adaptive Prediction" *Journal Of Information Science And Engineering*, 1579-1591 (2008).

[6] W. Liu, W.J. Zeng, L. Dong, and Q.M. Yao, "Efficient Compression Of Encrypted Grayscale Images" *IEEE Trans. on Image.* Process. vol. 19, no. 4, pp. 1097–1102, April 2010. http://dx.doi.org/10.1109/TIP.2009.2038773

[7] Mohammed Ali Bani and Aman Jantan, "An Image Encryption Approach Using a Combination of Permutation Technique Followed by Encryption", *IJCSNS International Journal of Computer Science and Network Security*, VOL.8 , April 2008.

[8] A. Razzaque, N. V. Thakur, "Image compression and encryption: an overview", *International Journal of Engineering Research & Technology*, Vol. 1. 5, pp. 1-7, July 2012.

[9] L. S. Sam, P. Devaraj and R. S. Bhuvaneswaran, "A novel image cipher based on mixed transformed logistic maps", *Journal of Multimedia Tools and Applications*, DOI 10.1007/s11042-010-0652-6, **(2010)**

[10] Xinpeng Zhang, "Lossy Compression and Iterative Reconstruction for Encrypted Images," *IEEE Trans. Information Forensics and Security*, Vol. 6, No. 1, pp. 53-58, Mar. 2011. http://dx.doi.org/10.1109/TIFS.2010.2099114

Suneera.H received the B-tech in Electronics and Communication Engineering Degree from Al Ameen Engineering college, shornur, under the university of Calicut, Kerala, India in 2012, and doing M-tech degree (2014-16) in VLSI Design Department of ECE, from Nehru College Of Engineering and Research Centre, Pampady, under the University of Calicut, Kerala, India.

Co-author :

Rahul M Nair Assistant Professor Nehru College Of Engineering And Research Centre, Pampady, Trichur, Kerala, India. B.Tech in Icet, Muvattupuzha (2009). Completed his M.Tech in Vjcet, Mg University (Vlsi & Embedded Systems -2011), Kerala, India.