International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

VLSI Design of an Efficient Image Encryption-Then-Compression System

Rose Mariya Kuriakose†* and Sunitha S Pillai†

†Department of Electronics & Communication Engg., St. Joseph's College of Engineering & Technology, Palai, Kerala, India

Accepted 20 Nov 2015, Available online 26 Nov 2015, **Vol.5, No.6 (Dec 2015)**

Abstract

In many practical scenarios, secure and efficient data transfer plays a vital role and it is the main aspect of the communication system. The classical way of transmitting redundant data over a bandwidth constrained insecure channel is to first compress it and then encrypt. This project investigates the novelty of reversing the order of compression and encryption, without compromising either the information secrecy or the encryption efficiency. 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. This image encryption-then-compression (ETC) system proposes an encryption scheme operated in the prediction error clustering and run length encoding, shown to be able to provide a reasonably high level of security. Lossless/lossy image coders can be exploited to efficiently compress the encrypted images, and both techniques have their own advantages. 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. This entire system is implemented by writing Verilog description and is simulated using Xilinx ISE software simulation tools.

Keywords: *Encryption, Compression, Run Length Encoding, Prediction error.*

1. Introduction

Cryptography, defined as the science and study of secret writing concerns the ways in which communications and data can be encoded to prevent disclosure of their contents through eavesdropping or message interception, using codes, ciphers and other methods, so that only certain people can see the real message. Cryptography enables us to store sensitive information or transmit it across insecure networks so that it cannot be read by anyone except the intended recipient. While cryptography is the science of securing data, Cryptanalysis is the science of analyzing and breaking secure communication. A cryptographic algorithm, known as the cipher, is a mathematical function used in the encryption and decryption process. A cryptographic algorithm works in combination with a key to encrypt the message and all possible keys and protocols is known as a Cryptosystem.

Encryption is the process of transforming a piece of information an algorithm to make it unreadable to anyone except those possessing special knowledge, usually referred to as a key. The output is known as the cipher text. The reverse process of transforming cipher text to plaintext is known as decryption .This is shown in Figure 1. With the advent of multimedia, the necessity for the storage of large numbers of high quality images is increasing. One obstacle that has to be overcome is the large size of image files. Compression saving storage space, as well as compressed files takes less time to transmit via modem, so money can be saved on both counts.

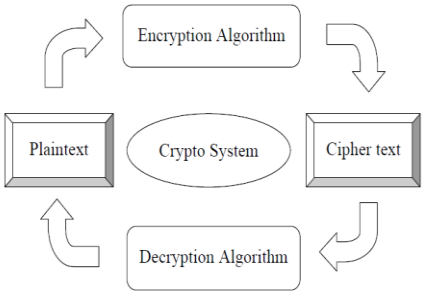

Fig.1 General Block Diagram of Cryptosystem

The choice of compression algorithm involves several conflicting considerations. These include degree of compression required, and the speed of operation. Obviously if one is attempting to run programs direct from their compressed state, decompression speed is

*Corresponding author **Rose Mariya Kuriakose** is a PG Scholar and **Sunitha S Pillai** is working as Assistant Professor

paramount. The other consideration is size of compressed file versus quality of compressed image.
There are essentially two sorts of data compression. Lossless compression works by reducing the redundancy in the data. The decompressed data is an exact copy of the original, with no loss of data. Huffman Encoding and Run Length Encoding are two examples of lossless compression algorithms. There are times when such methods of compression are unnecessarily exact. Lossy compression sacrifices exact reproduction of data for better compression. It removes redundancy and creates an approximation of the original. The JPEG standard is currently the most popular method of lossy compression. Obviously, a lossless compression method must be used with programs or text files, while lossy compression is really only suitable for graphics or sound data, where an exact reproduction is not necessary.

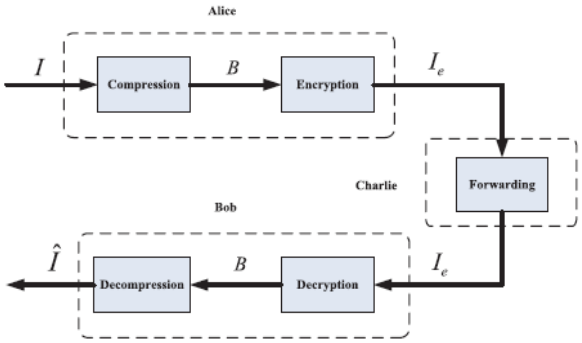

Fig.2 Compression-then-Encryption (CTE) system

Consider the problem of transmitting a confidential image over an insecure, bandwidth - constrained communications channel. It is desirable to both compress and encrypt the data. If a person holding some image I, say Alice wants to transmit it to a recipient Bob, via an un trusted channel provider Charlie. In conventional methods this could be occur as follows. Alice will compress I into B using any desirable technique at first, and then encrypts B into Ie. using an encryption algorithm having a secret key K, as shown in Figure 2. The channel provider Charlie will forward the encrypted data Ie, which is passed to him to Bob. On receiving Ie, Bob sequentially performs decryption using the same secret key K and decompression to get a reconstructed image Î.

This work investigate the novelty of reversing the order of these steps i.e., the order of applying the compression and encryption needs to be reversed in some situations, as shown in Figure 3. The compressor must be able to compress the encrypted data (cipher text) I.e. without any knowledge of the original source (plaintext) and the cryptographic key. The output of an encryptor will looks very random and it may leads to a minimal compression gain. The design of decoder should done carefully to accomplish both decompression and decryption are performed in a joint step. The decoder can use the cryptographic key to assist in the decompression of the received bit stream and finally generates reconstructed image Î.

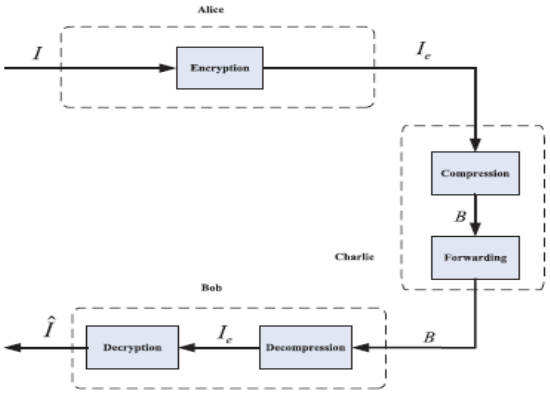

Fig. 3 Encryption-then-Compression (ETC) system

By applying LDPC codes in various bit planes and exploiting the inter/intra correlation, several methods for lossless compression of encrypted grayscale/color images (Lazzeretti, *et al*, 2008). Furthermore, this approach applied to the prediction error domain and achieved better lossless compression performance on the encrypted grayscale/color images (Makur, *et al*, 2008). This paper proposes compression of encrypted grayscale image. Aided by rate compatible punctured turbo codes, developed a progressive method to losslessly compress stream cipher encrypted grayscale/color images (Liu, *et al*, 2008). More recently, extended Johnsons framework to the case of compressing block cipher encrypted data (Klinc, et al, 2012). To achieve higher compression ratios, lossy compression of encrypted data was also studied (Yao, et al, 2009). The proposed encrypted images via a multi resolution construction using a scalable lossy coding framework (Zang, et al, 2012). . Furthermore, due to the high sensitivity of prediction error sequence against disturbances, reasonably high level of security could be retained.

The rest of this paper is organized as follows. Section 2 gives the details of our proposed ETC system, where reuse of predicted values secured with improved algorithm. Experimental results are reported in Section 3 to validate the findings and it's discussions are also included. The conclusion and future works are at the last section.

2. System Description

There are three key components in the proposed ETC system, namely, image encryption conducted by Alice, image compression conducted by Charlie, and the sequential decryption and decompression conducted by Bob.

A. Encryption.
B. Compression.
C. Decryption and Decompression.

A. Encryption

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, the propose an image encryption scheme operated over the prediction error domain. In our work, the GAP is adopted due to its excellent de-correlation capability. The prediction result Īij can be further refined to Îij through a context-adaptive, feedback mechanism. Consequently, the prediction error associated with Iij can be computed by

$$e_{ij} = I_{ij} - \bar{I}_{ij} \quad (1)$$

P0 is the current pixel in the raster scan order and let n = S(P2), w = S(P1), ne = S(P4), nw = S(P3), nn = S(P6), ww = S(P5), and nne = S(P9), which represent the north, west, northeast, northwest, north-north, west-west, and north-northeast neighbours of P0, respectively. Two gradient functions (vertical and horizontal gradients) can then be estimated by

$$dh = |w - ww + n - nw + n - ne| \quad (2)$$

$$dv = |w - nw + n - nn + ne - nne| \quad (3)$$

More specifically, for each pixel location (i,j), the error energy estimator is defined by the core algorithm of GAP predictor is described as follows:

```
IF (dv - dh > 80) Iij:=w
ELSE IF (dv - dh<-80)  Iij:=n
ELSE {
Iij:=(w+n)/2 + (ne - nw)/4
IF (dv - dh > 32)      Iij:=(Iij + w)/ 2
ELSE IF (dv - dh > 8)     Iij:=(3 x Iij + w)/ 4
ELSE IF (dv - dh<-32)  Iij:=(Iij + n)/ 2
ELSE IF (dv - dh<-8)    Ii;j:=(3 x Iij + n)/ 4
}
```

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 Ck, for 0 ≤ k ≤ L-1, where k is determined by any key generation algorithm, 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 Ik .

Step 5: The assembler concatenates all the permuted clusters C̄k, for 0 ≤ k ≤ L1, and generates the final encrypted image.

Step 6: Pass Ie to Charlie, together with the length of each cluster C̄k, for 0 ≤ k ≤ L2. The values of |Ck| enable Charlie to divide Ie into L clusters correctly. In comparison with the file size of the encrypted data, the overhead induced by sending the length |Ck| is negligible.

B. Compression

The compression of the encrypted file Ie needs to be performed in the encrypted domain, as Charlie does not have access to the secret key K. The lossless compression of Ie. Assisted by the side information |Ck| for 0 ≤ k ≤ L2, a de-assembler can be utilized to parse Ie into L segments C̄0, C̄1, ... C̄L-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 |Ck| into a binary bit stream Bk. Note that the generation of all Bk can be carried out in a parallel manner to improve the throughput. Eventually, an assembler concatenates all Bk to produce the final compressed and encrypted bit stream B, namely,

$$B = B0\ B1\ \ldots\ldots BL - 1 \quad (4)$$

C. Decryption and Decompression

Upon receiving the compressed and encrypted bit stream B, Bob aims to recover the original image I. The schematic diagram demonstrating the procedure of sequential decryption and decompression is provided in Figure. 4. According to the side information jBkj, Bob divides B into L segments Bk for 0 ≤ k ≤ L - 1, each of which is associated with a cluster of prediction errors.

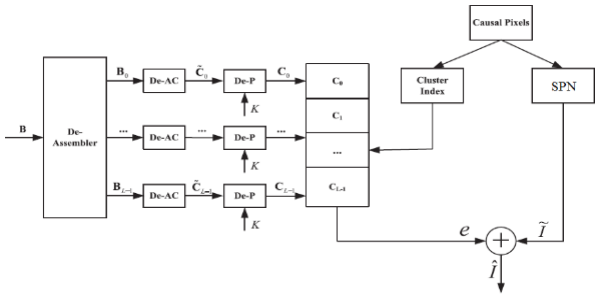

Fig. 4 Schematic diagram of sequential decryption and decompression

For each Bk, an adaptive arithmetic decoding can be applied to obtain the corresponding permuted prediction error sequence Ck. As Bob knows the secret key K, the corresponding de-permutation operation can be employed to get back the original Bk. With all the Bk, the decoding of the pixel values can be performed in a raster-scan order. For each location

(i,j), the associated error energy estimator e_{ij} and the predicted value $\bar{I}ij$ can be calculated from the causal surroundings that have already been decoded. The unused prediction error in the k^{th} cluster is selected as eij, which will be used to derive e_{ij} according to $\bar{I}ij$ and the mapping rule described in previous section. The reconstructed pixel value can then be computed by

$$\hat{I} = \bar{I}ij + eij \quad (5)$$

As the predicted value $\bar{I}ij$ and the error energy estimator $\bar{e}ij$ are both based on the causal surroundings, the decoder can get the exactly same prediction $\bar{I}ij$. In addition, in the case of lossless compression, no distortion occurs on the prediction error eij , which implies $\hat{I} = \bar{I}ij$, i.e., error-free decoding is achieved.

The possibility of ciphertext attack can be reduced to some extent by the use of SPN involutional algorithm. An involutional block cipher encrypts the plaintext P to generate the ciphertext C, and uses the exact same algorithm to generate the plaintext by decrypting the ciphertext. Using the characteristic of involution, for hardware implementations, allows the use of the same circuit for both encryption and decryption. As the reuse of prediction values are security limited operation, involutional block ciphers are an attractive choice. We call a function, F(x), involutional when it is its own inverse, that is

$$F(F(x)) = x \quad (6)$$

A substitution-permutation network (SPN) adopts a structure involving three basic components: substitution, permutation (or more generally, linear transformation), and the addition of round keys. They both have eight rounds of operation employing these components and make use of an 8x8 S-box and a 64-bit block size. The three byte level functions are X(a), X^2(a) = X(X(a)) and X^3(a) = X(X(X(a))).

Although it is possible to implement decryption in such a way that it has the same sequence of operations as encryption. With involutional ciphers, all the operations are involutions. So, it becomes possible to implement the decryption in such a way that only the round keys used are different from that for encryption. Besides the implementation benefit, an involutional structure also implies equivalent security for both encryption and decryption.

3. Results and discussions

1. 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 is shown in, Figure 5.

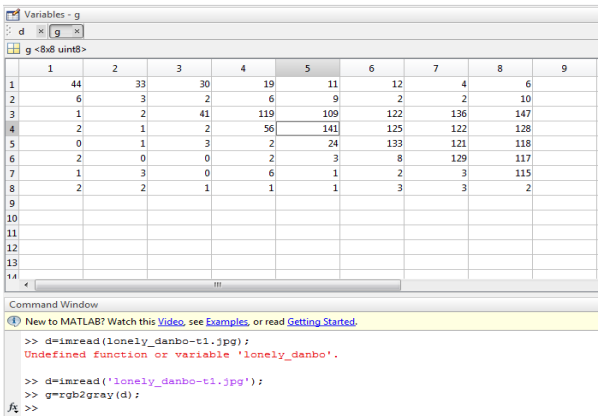

Fig.5 Matlab output - Pixels of image

2. GAP- is the linear predictor which designed as the combination of causal neighbours into improves coding rates. This block will generate prediction value associated with each pixel according to Gradient-adjusted prediction (GAP) algorithm. In this waveform, Figure 6. Dh, Dv represents horizontal and vertical gradient vectors respectively and SP0 represents prediction values.

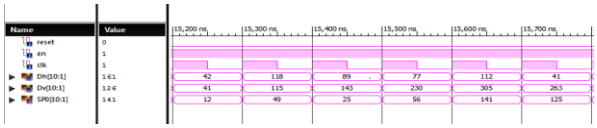

Fig.6 Simulated result-GAP

3. Compression- Next step to be done in this system is the compression of the encrypted data and it is done by Run Length encoding. The output, Figure 7 is represented by RLE, where R represents the Run value that means the magnitude and L represents the Length that is number of times the .

4. Decryption- All the processes done in the transmitter is done in a reverse manner to regenerate the original pixel values. The operations include reverse scanning, reverse shifting of rows and columns, reverse clustering, and finally the reverse mapping is also done. Figure 8 shows the remapped output.

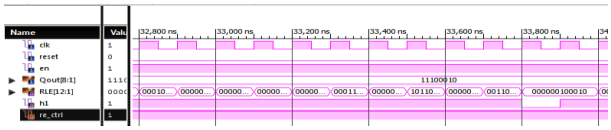

Fig.7 Simulated result-Compression

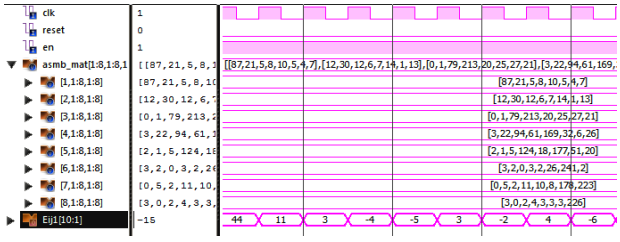

Fig.8 Simulated result-Decryption

Conclusions

In this work, we have designed an efficient image Encryption-then- Compression system. Within the proposed framework, the image encryption has been achieved via prediction error clustering and random permutation. Highly efficient compression of the encrypted data has then been realized by a context adaptive arithmetic coding approach. Both theoretical and experimental results have shown that reasonably high level of security has been retained. More notably, the coding efficiency of our proposed compression method on encrypted images is very close to that of the lossless/lossy image coding, which receive original, unencrypted images as inputs. A big challenge within such Encryption-then-Compression (ETC) framework is that compression has to be conducted in the encrypted domain, and this work is completed. In fact, the significant compression ratio can be achieved if compression is performed after encryption. The key challenge in this phase is to generate the pixel value as close as to the original pixel values at the transmitter side. The security concern about the reuse of prediction values are avoided by the use of SPN block at the transmitter as well as receiver. This work can be extending in the future to an efficient system for colour image processing.

References

Jiantao Zhou, Xianming Liu, Oscar C. Au, Yuan Yan Tang, (2014), 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.

T. 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.

D. Schonberg, S. C. Draper, and K. Ramchandran, (2005), On blind compression of encrypted correlated data approaching the source entropy rate, *in Proc. 43rd Annu. Allerton Conf.*, pp. 1–3.

D. Klinc, C. Hazay, A. Jagmohan, H. Krawczyk, and T. Rabin, (2012), On compression of data encrypted with block ciphers, *IEEE Trans. Inf.Theory,* vol. 58, no. 11, pp. 6989–7001.

R. Lazzeretti and M. Barni, (2008) Lossless compression of encrypted greylevel and color images, *in Proc. 16th Eur. Signal Process. Con.* , pp. 1–5.

A. Kumar and A. Makur, (2008), Distributed source coding based encryption and lossless compression of gray scale and color images*, in Proc. MMSP*, pp. 760–764.

W. Liu, W. J. Zeng, L. Dong, and Q. M. Yao, (2010) Efficient compression of encrypted grayscale images, *IEEE Trans. Imag. Process*., vol. 19, no. 4, pp. 1097–1102.

X. Zhang, G. Feng, Y. Ren, and Z. Qian, (2012), Scalable coding of encrypted Images, *IEEE Trans. Imag. Process.*, vol. 21, no. 6, pp. 3108–3114.

A. Kumar and A. Makur, (2009) Lossy compression of encrypted image by compressing sensing technique, in *Proc. IEEE Region 10 Conf.TENCON*, pp. 1–6.

X. Zhang, Y. L. Ren, G. R. Feng, and Z. X. Qian (2011) Compressing encrypted image using compressive sensing, *in Proc. IEEE 7th IIHMSP*, pp. 222–225.

X.Zhang (2011),Lossy compression and iterative reconstruction for encrypted image, *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 1, pp. 53–58.

Nikhil Joshi, Kaijie Wu, Ramesh Karri (2004), Concurrent Error Detection Schemes for Involution Ciphers, *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA*.

F.Y.Hsiehi, C.M.Wang, C.C Lee, K.C.Fan (2008),A Lossless Image Coder Integrating Predictors and Block-Adaptive Prediction*, Journal Of Information Science And Engineering*, pp.1579-1591.