Hardware Implementation of Novel Image Compression-Encryption System on a FPGA

Ahmet Çağrı Bağbaba, Berna Örs

Istanbul Technical University, Turkey
bagbaba@itu.edu.tr, Siddika.Ors@itu.edu.tr

Abstract

With the development of digital technologies, compression and encryption have become important aspects of information. In many applications, encryption of compressed images is considered vital for hiding data. In this work, we implemented the image compression-encryption hardware on a FPGA. JPEG was used as a compression standard and Tiny Encryption Algorithm was used as an encryption algorithm. Discrete Cosine Transform coefficients were encrypted by using two methods. First, only DC coefficients of each 8x8 block of image were encrypted. Then, along with DC coefficients, first five AC coefficients of each 8x8 block of image were encrypted. In the result, encrypted images and PSNR values were given in order to determine success of the work. This work is novel from the point of view encryption methods.

1. Introduction

Minimization of the number of information carrying units is important for image data compression [1]. Also, since the main target is reducing the memory and decreasing the bandwidth in communication, image compression is very useful [1]. In this area, Joint Photographic Experts Group (JPEG) is international standard due to the fact that it has high compression ratio and it does not cause of deformation of images [2]. Nowadays, digital images and the security of videos come into prominence in areas such as television broadcast, video conferences, medical imaging [3].

Security of data transmission is crucial as well. It is necessary to provide security of data or image by sending them in a channel [4]. In this work, Tiny Encryption Algorithm (TEA) was used for encryption. TEA is suitable for embedded systems owing to high performance, ease of implementation, high speed, low energy consumption, low cost, and being lightweight.

In this paper, we designed image compression-encryption hardware by using JPEG and TEA on a Field Programmable Gate Array (FPGA). Firstly, all blocks of JPEG and TEA were designed by using Verilog Hardware Design Language (HDL). Then, two encryption methods were applied by considering Discrete Cosine Transform (DCT) coefficients. In the ***first method***, only DC coefficients of all 8x8 DCT matrices were encrypted. In the ***second method***, along with all DC coefficients, first five AC coefficients of all 8x8 DCT matrices were encrypted in order to increase security level. Moreover, compression ratios were compared for both two methods. This work is novel from the point of used encryption algorithm (TEA) and applied encryption methods. Also, compression and encryption were implemented together in order to increase security level. Methods are not like encryption of one image which is in JPEG format. Instead, compression and encryption blocks were implemented as one hardware. Hence, if compressed-encrypted image is decrypted, it will not possible to obtain original image since the encryption was performed at the pixel level.

There are several works dealing with JPEG image encryption in the litarature. Bit Recirculation Image Encryption [5], Fuzzy PN Code Based Color Image Encryption Method [6], Combinational Permutation Method [7], SCAN Based Methods [8][9], and Chaos Based Methods [10-13] are examples of JPEG image encryption.

The rest of this paper is organized as follows: In Section 2, TEA and JPEG standard are reviewed briefly. In Section 3, proposed methods are introduced and implementation of compression-encryption system is described. In Section 4, hardware design on a FPGA is represented. In Section 5, implementation results and some comparisons are given. Finally, Section 5 concludes this paper.

2. Overview

2.1. Tiny Encryption Algorithm

TEA was for Encryption block of the system. The architecture of TEA is shown in Fig. 1. TEA is suitable for embedded systems owing to high performance for embedded systems, ease of implementation, high speed, low energy consumption, low cost, and being lightweight [14]. The TEA design's main aim is to provide minimum memory space and have maximum speed. Moreover, it uses Feistel Encryption type. As a result of this, when plain text is changed 1 bit, this reflects to output, which name is chipper text, as 32-bit. 128-bit key is used for encryption.

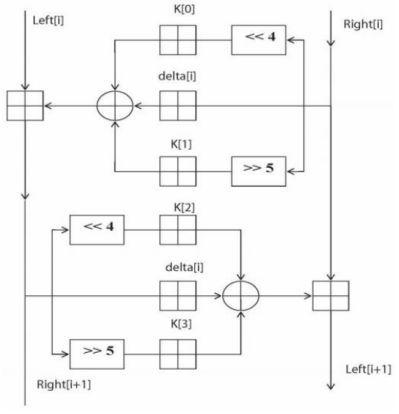

Fig. 1. Tea Cycle [14]

It can be seen in Fig. 1, there are Logical Shift, Add and XOR blocks in TEA Cycle. All bits of data and key are mixed through double shift [14]. Moreover, 128-bit key is used by splitting it into four 32-bit blocks as *K[0], K[1], K[3],* and *K[4].* Encryption of plain text is comprised from 64 Feistel Cycle. The plain text is processed by dividing into two parts as 32-bit. Different keys are used for each cycle. End of the 64 cycle, cipher text is derived. The constant number, delta, is calculated from the golden number ratio [15].

$$Delta = (\sqrt{5} - 1) \times 2^{31} = 9E3779B9_h \quad (1)$$

2.2. Joint Photographic Experts Group

JPEG standard includes DCT, Quantization, and Entropy Coder blocks and it can be seen in Fig. 2. Entropy coder in Fig. 3 concludes zigzag reordering, zero run length encoding, difference encoding, and Huffman Encoding.

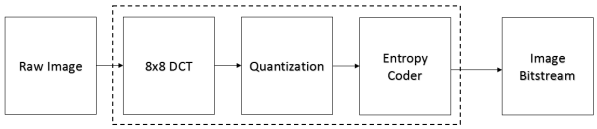

Fig. 2. JPEG Blocks

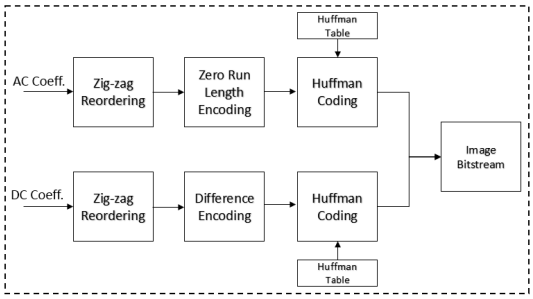

Fig. 3. Entropy Coder

The main aim of the DCT is to transform the value of pixels from spatial domain to frequency domain. The level of detail in an image is related to frequencies. High spatial frequencies correspond to high levels of detail, while lower frequencies correspond to lower levels of detail [16]. The result matrix of DCT consists of 64 DCT coefficients. The first coefficient D(0,0) is the DC coefficient and the other 63 coefficients are AC components. DCT equation is given in Equation 2. In this equation, C is the DCT matrix given in JPEG standard, and X is the 8x8 block of image matrix.

$$Z = CXC^T \quad (2)$$

The next step in the compression process is the quantization of the DCT matrix. It is the process of reducing number of bits needed to store an integer value by reducing the precision of the integer. The quantization process has the key role in the JPEG compression. The quantization cycle has readily apparent effects on an image. The low frequency elements which are close to DC coefficient have been modified. The high frequency elements have been reduced to zero. Due to rounding in quantization, it is the lossy step of JPEG standard. As a result, insignificant data in matrix is discarded.

After doing the DCT transform and quantization over a block of 8x8 values, a new 8x8 block is obtained. All coefficients in these 8x8 blocks are traversed in zig-zag ordering like in Fig. 4. The reason for this zig-zag traversing is that we traverse the 8x8 DCT coefficients in the order of increasing the spatial frequencies [16]. For AC coefficients, zero run length encoding is implemented. For DC coefficients, difference encoding is implemented. In the final step, Huffman Coding is applied for all coefficients.

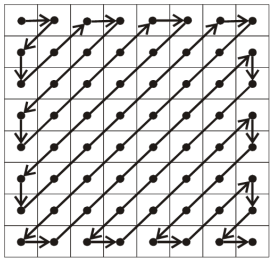

Fig. 4. Zigzag Reordering

3. Proposed Methods and Implementation

In this work, we implemented two encryption methods as explained below:

1. All DC coefficients of all 8x8 DCT matrices are encrypted after zigzag reordering (Fig. 5a).
2. All DC coefficients and first 5 AC coefficients of all 8x8 DCT matrices are encrypted after zigzag reordering (Fig. 5b).

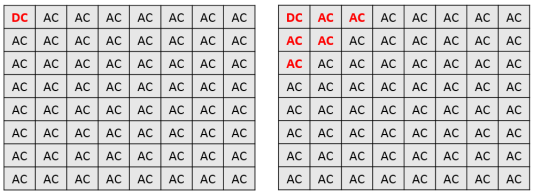

Fig. 5. a. First Method b. Second Method

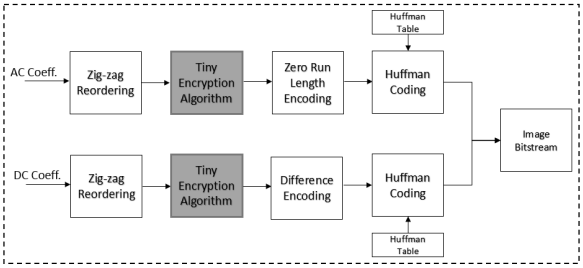

Fig. 6. Encryption

In Fig. 5, methods were explained for better understanding. Two methods were implemented for all 8x8 matrices. First method encrypts only DC coefficients and second method encrypts DC coefficients and first 5 AC coefficients, written in red, according to the increasing order of frequency. For both two

methods, place of encryption block is shown in Fig. 6. Encryption was not implemented after obtaining image in JPEG format. Instead of this, compression and encryption designed together and encryption was applied to only selected pixels. This provides more secure system if someone wants to decrypt the result image. Encryption is applied after zigzag reordering due to the fact that frequency of pixels is vital for this work. First elements of zigzag ordering vectors called as DC coefficients have the lowest frequency also they are the most important pixels in terms of carrying information about image. Therefore, encryption of these pixels causes changing of image dramatically. Also, AC coefficients near DC coefficients have important information about image when compared to other AC coefficients. Hence, we encrypted five of them in the second method in order to obtain more secure system.

4. Hardware Design

In this work, we designed JPEG and TEA blocks in FPGA by using Verilog Hardware Description Language (HDL). Hardware designed on Xilinx Virtex-6 FPGA ML605 Evaluation Kit. Hardware design includes DCT, Quantization, Zig-zag reordering, TEA, DC-Difference Encoding, Coding, and Packaging blocks.

In the DCT design, we implemented 1-D and 2-D processes separately. In the 1-D DCT, XC^T was calculated firstly. In the 2-D DCT, the result of 1-D DCT was multiplied with coefficient matrix C given in Equation 2. For this design, it is necessary that all input pixels have to be ready at the same time. In order to provide these inputs, ping-pong buffer [17] was used given in Fig. 7. In ping-pong buffer, there are 8 registers and when the input is available, it is stored at the first register. Previous value of register is shifted to the next register. Hence, the buffer is sampled for each 8 clock cycle. After the 8 inputs are ready in registers, enable is activated in order to send these inputs to outputs. We need this block since inputs of DCT block are serial but 8 pixel values have to be ready at the same time in order to calculate DCT values. During these processes, input registers takes new input pixels. Therefore, there is no need to wait for DCT calculations or new inputs.

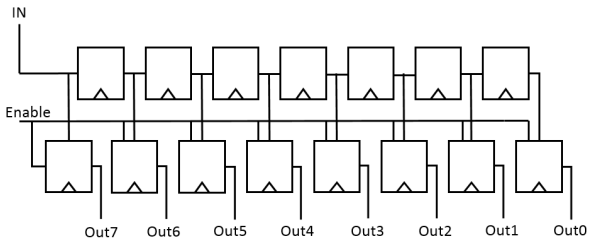

Fig. 7. Ping-Pong Buffer [17]

All results were stored by using dual RAM structure [17] given in Fig. 8. 1-D DCT and 2-D DCT results were stored at these RAMs and they were used alternately. In other words, 1-D DCT results are written to the first RAM while other RAM provides results which were written before. This structure helps us in order to provide continuity of calculations.

For the design of Quantization, block in Fig. 9 was used. Division operation, necessary for quantization, was converted to the multiplication operation because FPGA has dedicated multiplication blocks in it whereas division operation is not efficient in hardware design. Therefore, the result of $65536 * (1/Q)$ (Q is the quantization matrix given in JPEG standard) was stored in

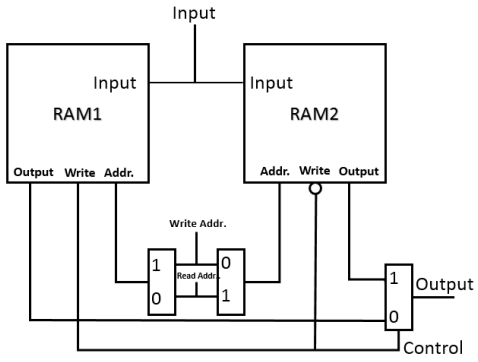

Fig. 8. Dual RAM [17]

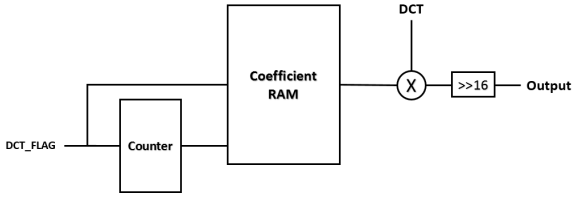

Fig. 9. Quantization Block

Coefficient RAM. It is obvious that multiplication of the result of $65536 * (1/Q)$ with DCT results is equal to the 65536 times of division result. Finally, 16-bit right shift is applied for the exact result. Counter was used to address the Coefficient RAM and it makes sampling for each clock cycle.

In this work, we modified TEA by changing the number of input bits. In standard TEA, input is 64-bit and it is divided left and right side as 32-bit. We modified the input of circuit as 72-bit so our left and right sides are 36-bit. However, key length is still 128-bit. This modification was necessary for compatibility of the hardware. Also, this modification is not the cause of reduction in the security level of the algorithm.

Zigzag reordering, Difference Encoding, Coding, and Packaging Blocks were also designed as a hardware by using blocks given before.

5. Implementation Results

Two images were used in order to test the design. Input images are size of 288x288 and they are bitmap. Original images can be seen in Fig. 10a and Fig. 11a. Also, the result of first and second method can be seen in Fig. 10 and Fig. 11.

Fig. 10b and Fig. 11b are results of the first method. In these images, it can be seen that some patterns of original images can be detected. Therefore, we need to implement second method since encryption of only DC coefficients of 8x8 blocks is not enough to hide all pixels of images. According to the zigzag reordering, AC coefficients which are very close to DC coefficients have lower frequency when compared the other coefficients and also they have important data about image pixels. Therefore, we implemented second method to obtain totally successful image encryption. However, encryption of more pixels decreases compression ratio so in the second method the result image has bigger size than the result of the first method but still we had a good compressed result images due to the fact that great majority of pixels are not encrypted or changed.

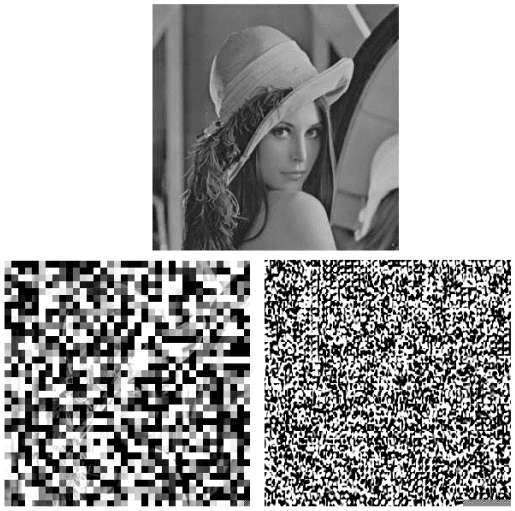

Fig. 10. a. Original Lena Image b. First Method c. Second Method

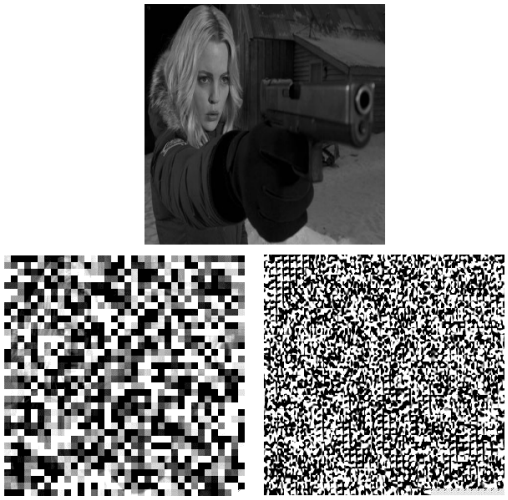

Fig. 11. a. Original Melissa Image b. First Method c. Second Method

In Table 1 and in Table 2 some information about Lena and Melissa images can be seen. In Table 1, original bitmap size of Lena and Melissa were given. Also, implementation of only JPEG blocks (without encryption) were examined and it was called as JPEG image size in the same table. Peak Signal to Noise Ratio (PSNR) and Root Mean Squared Error (RMSE) results were given in Table 2. Moreover, in the last rows of Table 2 for Lena and Melissa, compressed-encrypted image sizes were given. In the second method, result image's size is bigger than the first method since in the second method more pixels are encrypted than the first method. Therefore, decrease of zero pixel values in the zigzag vectors are the reason of increasing in the size. These values show that proposed methods are successful. PSNR values should be low for good encryption. Maximum frequency of hardware is 87.290 MHz. In Table 3 device utilization of hardware was given.

Table 1. Size Information

	Bitmap Image Size	***JPEG Image Size***
LENA	83 KB	8 KB
MELISSA	83 KB	6 KB

Table 2. Lena and Melissa Implementation Results

LENA	***First Method***	***Second Method***
PSNR	7.02 dB	6.18 dB
RMSE	136.54	152.27
Result Image Size	9 KB	22 KB
MELISSA	***First Method***	***Second Method***
PSNR	6.56 dB	5.27 dB
RMSE	141.32	156.022
Result Image Size	7 KB	21 KB

Table 3. Device Utilization

	Used	*Available*
Number of Slice Registers	1456	301440
Number of Slice LUTs	2130	150720
Number of Bonded IOBs	167	600
Number of DSP48E1s	9	768

6. Conclusion

In this paper, we implemented image compression-encryption hardware on a FPGA. JPEG was selected as a compression standard and TEA was used as an encryption algorithm. Two methods were determined and applied. First method is encryption of all DC coefficients. Second method is encryption of all DC coefficients and all first five AC coefficients of 8x8 DCT matrices. All blocks were implemented as a hardware on a FPGA by using Verilog HDL. In the results, we obtained successful compression-encryption and the results given in tables. Results show that first method is successful in terms of compression ratio but second method is better in terms of encryption of images. These two methods are the novelty of this work.

7. References

[1] Jain, A., "Image data compression: A review," *Proceedings of the IEEE* , vol.69, no.3, pp.349,389, March 1981.

[2] Neelamani, R.N.; de Queiroz, Ricardo; Fan, Z.; Dash, S.; Baraniuk, R.G., "JPEG compression history estimation for color images," *Image Processing, IEEE Transactions on* , vol.15, no.6, pp.1365,1378, June 2006

[3] B K ShreyamshaKumar, ; Chidamber R Patil. (2009). JPEG Image Encryption using Fuzzy PN Sequences. *Signal Image and Video Processing*, 2010, 1-9-9

[4] Kumar, S.K.N.; Kumar, H.S.S.; Panduranga, H.T., "Hardware software co-simulation of dual image encryption using Latin square image," *Computing, Communications and Networking Technologies (ICCCNT),2013 Fourth International Conference on* , vol., no., pp.1,5, 4-6 July 2013

[5] Jui-Cheng Yen; Jiun-In Guo, "A new image encryption algorithm and its VLSI architecture," *Signal Processing Systems, 1999. SiPS 99. 1999 IEEE Workshop on* , vol., no., pp.430,437, 1999

[6] El-Khamy, S.E.; Lotfy, M.A.; Ali, A.H., "A new color image encryption. technique utilizing fuzzy pseudo-random bit generator," *Radio Science Conference, 2005. NRSC 2005. Proceedings of the Twenty-Second National* , vol., no., pp.185,194, March 15-17, 2005

[7] Mitra A., Subba Rao Y. V. and Prasanna S. R. M, "A New Image Encryption Approach using Combinational Permutation Techniques", *International Journal of Computer Science*, Vol. 1, No. 2, pp. 127-131, 2006.

[8] Bourbakis N. and Alexopoulos C, "Picture Data Encryption using SCAN Pattern", *Pattern Recognition*, vol.25, no.6, pp.567-581, 1992.

[9] Chao-Shen Chen; Rong-Jian Chen, "Image Encryption and Decryption Using SCAN Methodology," *Parallel and Distributed Computing, Applications and Technologies, 2006. PDCAT '06. Seventh International Conference on* , vol., no., pp.61,66, Dec. 2006

[10] Pichler F. and Scharinger J, "Efficient Image Encryption based on Chaotic Maps", *Proceedings of 20th Workshop of the Austrian Association for Pattern Recognition*, pp.159-170, Leibnitz, Austria, preprint, 1996.

[11] Li S. and Zheng X, "Cryptanalysis of a Chaotic Image Encryption Method", *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS'02)*, Vol. 2, pp. 708-711, 2002.

[12] Socek D., Li S., Magliveras S. S. and Furht B, " Enhanced 1-D Chaotic Key-Based Algorithm for Image Encryption", *IEEE Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*, pp.406-408, Sep 2005.

[13] Gu G. and Han G.: "An Enhanced Chaos Based Image Encryption Algorithm", *IEEE Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC'06)*, Vol.1, pp.492-495, Sep 2006.

[14] Abdelhalim, M.B.; El-Mahallawy, M.; Ayyad, M.; Elhennawy, A., "Implementation of a modified lightweight cryptographic TEA algorithm in RFID system," *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for* , vol., no., pp.509,513, 11-14 Dec. 2011

[15] Bagbaba, A.C.; Ors, B., "Implementation of a secure Near Field Communication system on a FPGA," *Electrical and Electronics Engineering (ELECO), 2013 8th International Conference on* , vol., no., pp.621,625, 28-30 Nov. 2013

[16] Acharya, T., Tsai, P.-S. (2005). JPEG2000 Standard for Image Compression: Concepts, Algorithms and VLSI Architectures. Image Rochester NY.

[17] Agostini, L.V.; Silva, I.S.; Bampi, S., "Pipelined fast 2D DCT architecture for JPEG image compression," *Integrated Circuits and Systems Design, 2001, 14th Symposium on.* , vol., no., pp.226,231, 2001