VLSI ARCHITECTURE FOR ENCRYPTION AND WATERMARKING UNITS TOWARDS THE MAKING OF A SECURE CAMERA

O. B. Adamo Saraju P. Mohanty E. Kougianos M. Varanasi
oba0002@unt.edu smohanty@cse.unt.edu eliask@unt.edu varanasi@unt.edu
VLSI Design and CAD Laboratory, University of North Texas, P. O. Box 311366, Denton, TX 76203.

Abstract— **Considerable amount of research is directed at putting biometric data in conventional forms of identification such as passports. However, putting biometric data in passports makes the data vulnerable to theft, causing privacy related issues. To address such issues, we present a new approach and architecture in the framework of a digital camera, conceptualized as a "Secure Digital Camera (SDC)". The SDC uses watermarking and encryption processes for image security and authentication. The Rijndael AES algorithm and a DCT-based visible watermarking algorithm were chosen for implementation in our camera. The proposed architectures were modeled, simulated and synthesized in Xilinx ISE.**

I. INTRODUCTION

To enhance security, the Department of Homeland Security proposed inserting biometric data such as fingerprints, iris scans, signatures etc. in individual passports and visas. Biometrics is an important tool that can identify and crosscheck a person's identity [1]. However, such a vast database of biometric information makes an enticing target for hackers and terrorists. Counteracting unlawful attempts by protecting and preventing modification of biometric information creates an urgent need for development of protection mechanisms. An effective solution can be judicious use of watermarking and encryption together at the source end of the biometric process in hardware like digital camera or scanners etc. Watermarking is the process whereby a host image is embedded with data for the purpose of protection and authentication. On the other hand, encryption is the transformation of data into secret code with the purpose of protecting the secrecy of the data when sent through an insecure channel.

Several attempts have been made to develop the different units of a digital camera with watermarking capabilities, but few have dealt with the design of the entire camera. Only some of these attempts have also incorporated cryptography in the camera design. As a result, we present the design and architecture of a digital camera system that incorporates watermarking and encryption. The trustworthy camera, with the aim of restoring credibility to photographic images using encryption, is presented in [2]. A biometric authentication system for a secure camera is developed in [3], however, a VLSI architecture was not proposed. Authors in [4] presented a design for a CMOS APS imager incorporating circuits for a pseudo-random generator for invisible watermarking. The authors in [5] presented a VLSI architecture for implementing two digital watermarking schemes.

Industries have also produced cameras with watermarking capabilities; however these camera models were discontinued. For example, Epson released the PhotoPC 3000Z and 800Z model and Kodak also manufactured the DC-200 and DC-260 but were all discontinued [3]. In this paper, as a capstone to previous work, we introduce an architecture for a SDC with both watermarking and encryption capabilities for image security and authentication.

II. CONTRIBUTIONS OF THIS PAPER

We present a new concept of an SDC for image security, protection, and authentication. We propose that the SDC watermarks biometric data into the image of an individual taken by it using an invisible watermarking technique. The watermarking key is encrypted and then embedded visibly in the form of a barcode on the picture image. We therefore, present a hardware implementation of the Advanced Encryption Standard (AES) and a DCT based visible watermarking algorithm. The architecture for AES [6] is area optimal as the round key for each round is calculated on the fly, instead of calculating all round keys and storing them. The architecture that embeds the barcode is based on the visible watermarking algorithm in [7].

III. BIOMETRIC DATA PROTECTION USING THE PROPOSED SECURE DIGITAL CAMERA

In this section, we discuss the proposed SDC and its main components as shown in Fig. 1. The camera will invisibly watermark biometric information such as "iris image", "handwritten signature", "fingerprint" etc. into an individual's image, which is then added to the passport. The watermarking will be key-based and this key will be encrypted and then embedded as a visible watermark in the form of a barcode on the picture image. This unique concept of biometric data hiding is presented in Fig. 2. The robustness of the invisible watermark and the authenticity of the picture image will be based on the secret key. The biometric data cannot be accessed and extracted unless the secret key is known. At the same time, the secret key for the invisible watermarking process cannot be known unless it is decrypted. Hence, our design offers double protection to the biometric data embedded into the picture image. This approach will take care of the privacy issues pertaining to the owners of the biometric data.

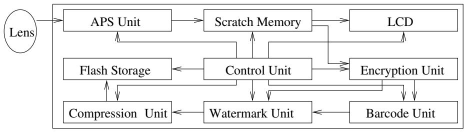


Fig. 1. Block Diagram of Proposed Secure Digital Camera (SDC)

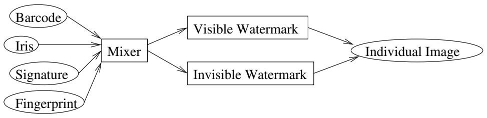


Fig. 2. Storing Biometric Information as Watermarks

To extract the biometric data, the encrypted key needed is obtained from the picture image by scanning the barcode. The encrypted key is decrypted and then used to extract the biometric information from the picture image. It is assumed that the extraction process is performed offline. The extracted biometric data is compared with the individual's data for verification and authentication.

IV. Algorithms Selected for Integration

In this section we discuss the AES and visible watermarking algorithms whose architectures are developed.

A. *Rijndael AES Algorithm*

Rijndael AES algorithm is a key-iterated block cipher where the round key needed to produce cipher text from plain text is derived from an initial key [8]. The keys derived from the initial key are repeatedly applied for the transformation of the plain text. The algorithm is a linear transformation cipher that is based on S-boxes. The algorithm supports fixed block and variable key length of 128, 192 or 256 bits [6] [9]. Data are handled as bytes in Rijndael in which each byte forms an element in a polynomial representation of Galois Field $GF(2^8)$ [10].

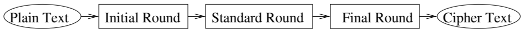


Fig. 3. Block Diagram of Rijndael AES Algorithm

The encryption process can be broadly divided into initial, standard and final round phases as shown in Fig. 3. The initial round phase involves the initial key and data addition. The standard round is made up of four operations: (i) byte substitution, (ii) row shifting, (iii) column mixing, and (iv) addition of round keys. These four operations are needed to complete one round. The number of rounds carried out depends on the key length. The AES with 128 bits key and block length consists of 10 rounds along with an initial addition of the round key. The final round has the same operations as the standard round except for the column mixing operation.

B. *DCT-Based Visible Watermark Algorithm*

During the first phase of the DCT based visible watermarking algorithm [7], the host image (passport photo) and the watermark image (barcode image) are divided into 8×8 blocks. The DCT coefficients of the 8×8 blocks of the host I and watermark images W are then calculated. The DCT coefficient of block n is represented by $c_{ij}(n)$, where n represents the position of block in image I. The mean gray value μ_n for each block of the original image is calculated using Eqn. (1), where $c_{00}(n)$ represents the DC coefficient of block n:

$$\mu_n = c_{00}(n). \tag{1}$$

The gray value of the normalized mean $\mu_n^{'}$ of block n is calculated using Eqn. (2), where c_{00max} represents the maximum value of $c_{00}(n)$:

$$\mu_n^{'} = \frac{c_{00}(n)}{c_{00max}}. \tag{2}$$

The maximum value of normalized mean of the host image I is calculated with Eqn. (3):

$$\mu_n^{'} = \frac{1}{N}\sum_{n=1}^{N} c_{00}(n). \tag{3}$$

The variance $\sigma_n^{'}$ of the AC-DCT coefficients of n-th block is calculated using Eqn. (4), where μ_{nAC} is the mean.

$$\sigma_n^{'} = \frac{1}{64}\sum_{n=1}^{N}\sum_{n=1}^{N}(c_{00}(n) - \mu_{nAC})^2 \tag{4}$$

If α_n and β_n are the scaling factors for n-th block, then the DCT coefficient of the watermark image (barcode) w_{ij} is combined together block wise with the host image (passport image) to obtain the watermarked image (barcode marked passport photo) using:

$$c_{ij} = \alpha_n c_{ij} + \beta_n w_{ij} \tag{5}$$

V. Architectural Implementations

In this section we discuss the architectures developed for AES encryption and visible watermarking algorithms.

A. *Architecture for Encryption Unit*

A high level view of the encryption unit architecture is presented in Fig. 4. Our implementation supports 128 bits of data and key length. The initial round module is carried out by XORing the 128 bit plain text with the 128 bit input key. The plain text input and the key input are retrieved from the input register. The output from the initial round is then passed through a 4 to 1 multiplexer to the register for temporary storage, after which it is then passed to the round module. A round key is generated for each round by the key schedule. The output is iterated back into the round module through the MUX. The round module is executed nine times. The control module takes care of the sequence of operations of the encryption unit.

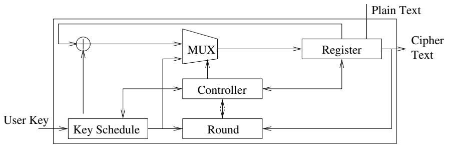


Fig. 4. High level architecture of Encryption Unit

1) Round Module: As shown in Fig. 5, the round module is made up of the Bytesub, Shiftrow, Mixcolumn, and Addroundkey submodules. The Mixcolumn submodule is not used in the final round.

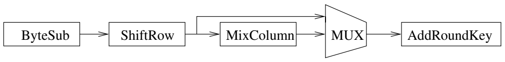


Fig. 5. Architecture of Round Module

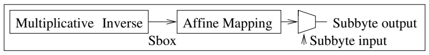


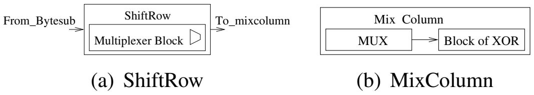
Fig. 6. Architecture of ByteSub Submodule

ByteSub: This is made up of multiplicative inverse in $GF(2^8)$ and affine mapping over $GF(2)$ transformations as shown in Fig. 6. The field $GF(2^8)$ is defined by finding a polynomial that is irreducible over $GF(2)$. The byte substitution operation is carried out in this paper with the aid of S-box. The architecture consists of 16 S-boxes working in parallel. An input byte is replaced by its corresponding value from the S-box.

ShiftRow: The architecture shown in Fig. 7(a) shifts the position of bytes in the states by offsets in cycle. The first Row is unshifted. The second row is shifted to the left once, the third row is shifted to the left twice, and the fourth is shifted thrice.

MixColumn: The elements of columns in a state are considered as coefficients of polynomial over Galois field $GF(2^8)$, where these elements are smaller than three. This polynomial is then multiplied by the fixed polynomial $\left(c(x) = (03)X^3 + (01)X^2 + (01)X + (02)\right)$ modulo $\left(X^4 + 1\right)$. The operation could be carried out using matrix operations. The column mixing step is basically a matrix multiplication in the Galois field, which is carried out using shift and XOR operations and the architecture is shown in Fig. 7(b).

AddRoundkey: The round key from the key generator is XORed with the block state obtained from the mixcolumn transformation as shown in Fig. 8.



(a) ShiftRow (b) MixColumn

Fig. 7. Architecture of ShiftRow and MixColumn

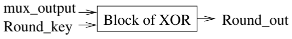


Fig. 8. Architecture of AddRoundkey Submodule

2) Key Schedule: The round key is obtained from the initial key through key expansion using the architecture in Fig. 9. If it is the first round in the standard round module, the multiplexer outputs the initial key for expansion. The module is able to generate subsequent round keys from the initial round keys through the register. As a result, round keys are generated at every round thereby reducing area requirements. The bytesub operation in the key schedule was implemented using S-Boxes. Round key computation is completed in one clock cycle.

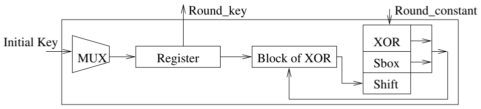


Fig. 9. Architecture of Key Schedule module

3) Controller module: The sequence of operation of the system is determined by the control module. Multiplexer select inputs and register load signals are provided by the control module. The controller was implemented as a Finite State Machine (FSM) with twelve states.

B. Architecture of Visible Watermarking Unit

The high level view of the watermarking architecture is shown in Fig. 10. The mean gray value of host image block is computed from the DCT coefficients as the DC value of the 8×8 DCT block coefficient. In addition, the Mean and variance of the AC-DCT coefficients for each block is computed using the perceptual analyzer and the architecture as shown in Fig. 11.

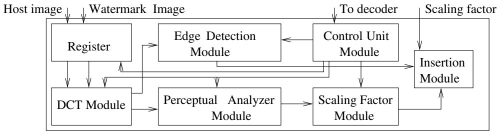


Fig. 10. High Level Architecture of Watermarking Unit

The scaling factor determines the ratio of host and watermark images in the final watermarked image and is calculated using the scaling and embedding factor module whose architecture is shown in Fig. 12.

The blocks that are at the edge of the image are determined by the edge detector block shown in Fig.

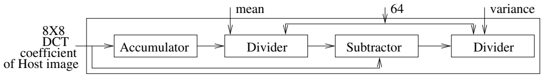


Fig. 11. Architecture of Perceptual Module

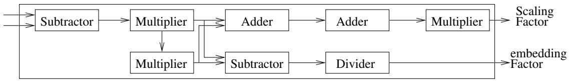


Fig. 12. Architecture of Scaling Module

13. The final watermarking process is carried out by the insertion unit that implements Eqn. (5).

VI. Prototyping and Simulation Results

The architectures were modeled using VHDL and the functional simulation was carried out using Modelsim XE III 6.0a tools. The VHDL code was compiled using Xilinx ISE 8.1i. The synthesis of the architectures was carried out for VIRTEX -II technology with xc2v500-6fg256 target devices. The timing simulation that was obtained with the aid of Modelsim is shown in Fig. 14. The cell usage is also presented in Table I which is all the logical cells that are basic elements of the technology. The minimum period is the timing path from a clock to another clock in the design. The encryption unit's rate based on simulation is determined to be $0.5 Gbps$. We performed exhaustive testing of the prototype units in the camera on several test images and two of them are shown in Fig. 15.

VII. Conclusions and Future Works

This paper presented a novel architecture and design of a secure digital camera that incorporates encryption and watermarking. The FPGA implementation of the encryption and watermarking unit of the SDC were presented. The design and implementation of the remaining components of the secure camera is being conducted as an on-going research in our laboratory. We plan to implement the barcode and compression units next. We also plan to optimize our design once the entire SDC is simulated.

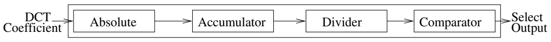
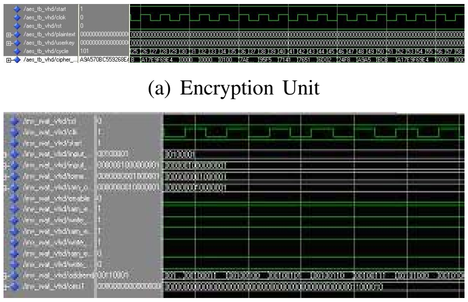


Fig. 13. Architecture of Edge Detector Module



(a) Encryption Unit

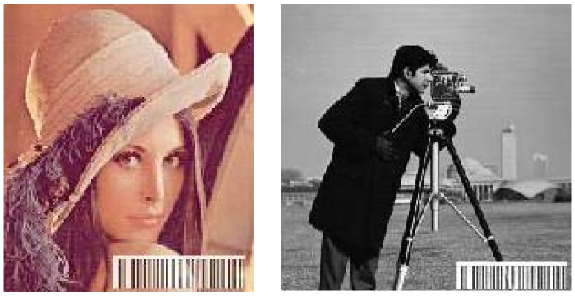
(b) Watermarking Unit

Fig. 14. Simulation of Encryption and Visible Watermarking Units

TABLE I

Encryption and Watermarking Unit Simulation Results

Parameters	Encryption Unit	Watermark Unit
Cells Usage (BELs)	6117	639
Maximum Frequency	217 MHz	96.318
Critical Path Delay	4.383ns	28.485ns
Minimum Time Period	4.969ns	10.318ns



(a) Lena

(b) Cameraman

Fig. 15. Experimental Results in Test Images

References

[1] "PC World Magazine, Monday January 17 2005," http://www.pcworld.com/news/article/0,aid,119324,00.asp.

[2] O. L. Friedman, "The Trustworthy Digital Camera: Restoring Credibility to the Photographic Image," *IEEE Transactions on Image Processing*, vol. 6, no. 4, pp. 905–910, November 1993.

[3] P. Blythe and J. Fridrich, "Secure Digital Camera," in *Proceedings of Digital Forensic Research Workshop (DFRWS)*, 2004.

[4] G. R. Nelson, G. A. Jullien, and O. Y. Pecht, "CMOS Image Sensor with watermarking Capabilities," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, 2005, pp. 5326–5329.

[5] S. P. Mohanty, N. Ranganathan, and R. K. Namballa, "VLSI Implementation of Visible Watermarking for a Secure Still Camera Design," in *Proceedings of International Conference of VLSI Design*, 2004, pp. 1063–1068.

[6] "NIST Federal Information Processing Standards Publication 197: Advanced Encryption 2001," http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[7] S. P. Mohanty, K. R. Ramakrishnan, and M. S. Kanakankalli, "A DCT Domain Visible Watermarking Techniques for Images," in *Proc. of IEEE International Conf on Multimedia and Expo*, 2000, pp. 1004–1009.

[8] J. Daemen and V. Rijmen, "The Design of Rijndael," in *Springer-Verlag*, 2002.

[9] N. M. Kosaraju, M. Varanasi, and S. P. Mohanty, "A High Performance VLSI Architecture for Advanced Encryption Standard(AES) Algorithm," in *Proc. of 19th IEEE International Conference on VLSI Design*, 2006, pp. 481–484.

[10] C. Chitu, D. Chien, I. Verbauwhede, and F. Chang, "A hardware implementation in FPGA of the Rijndael Algorithm," in *Proc. of 39th ACM/IEEE Design Automation Conference*, June 2002, pp. 399–404.