AES Cardless Automatic Teller Machine (ATM) Biometric Security System Design Using FPGA Implementation

To cite this article: Nabihah Ahmad et al 2016 IOP Conf. Ser.: Mater. Sci. Eng. 160 012113

View the article online for updates and enhancements.

Related content
- Mathematical and information maintenance of biometric systems
  Z Boriev, S Sokolov, A Nyrkov et al.
- Method of preliminary localization of the iris in biometric access control systems
  N Minacova and I Petrov
- Software and hardware user authentication methods in the information and control systems based on biometrics
  Z Boriev, A Nyrkov, S Sokolov et al.
AES Cardless Automatic Teller Machine (ATM) Biometric Security System Design Using FPGA Implementation

Nabihah Ahmad, A. Aminurdin M. Rifen, Mohd Helmy Abd Wahab

Nano Simulation Group (NanoSIM), Department of Electronic Engineering, Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Batu Pahat, Parit Raja, Johor

Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Batu Pahat, Parit Raja, Johor

nabihah@uthm.edu.my

Abstract. Automated Teller Machine (ATM) is an electronic banking outlet that allows bank customers to complete a banking transactions without the aid of any bank official or teller. Several problems are associated with the use of ATM card such card cloning, card damaging, card expiring, cost of issuance and maintenance and accessing customer account by third parties. The aim of this project is to give a freedom to the user by changing the card to biometric security system to access the bank account using Advanced Encryption Standard (AES) algorithm. The project is implemented using Field Programmable Gate Array (FPGA) DE2-115 board with Cyclone IV device, fingerprint scanner, and Multi-Touch Liquid Crystal Display (LCD) Second Edition (MTL2) using Very High Speed Integrated Circuit Hardware (VHSIC) Description Language (VHDL). This project used 128-bits AES for recommend the device with the throughput around 19.016Gbps and utilized around 520 slices. This design offers a secure banking transaction with a low rea and high performance and very suited for restricted space environments for small amounts of RAM or ROM where either encryption or decryption is performed.

1. Introduction
On 1997 National Institute of Standard and Technology (NIST) have announced Advanced Encryption Standard (AES) as more protective and secure data encryption replacing the Data Encryption Standard (DES) [1]. AES is cryptographic algorithm was first used for encryption data by the United State of America. AES is easy to implement on software and hardware. Communication and data transfer in the present days invariably necessitate the use of encryption. It is used in military and government’s secret communication, and protecting many kinds of civilian systems such as Internet e-commerce, mobile networks, Automatic Teller Machine (ATM) transactions, copy protection (especially protection against reverse Engineering and Software piracy), and many more. Data encryption is achieved by following a systematic algorithm called encryption algorithm. An encryption algorithm provides confidentiality, authentication, integrity and non-repudiation. Confidentiality is the requirement that information is kept secret from people who are not authorized to access it. Authentication is the certainty that the message indeed originates from the purported sender [2].

ATM is an electronic telecommunications device that enables the customers of a financial institution to perform financial transactions, particularly cash withdrawal without the need for a human cashier, clerk or bank teller. On modern ATMs, customer is identified by inserting a plastic ATM card
with a chip that contains a unique card number and some security information such an expiration card. Authentication of the customer provided by entering a personal identification number (PIN). In recent (ranging from fraud, stealing, other) of ATM usage has been attributed to use of ATM such as card cloning, card damaging, card expiring, maintenance, and accessing customer account by third parties. All these can be a bygone issue if cardless ATM can be designed and implemented in future. A biometric security system is used to enhance the security of the ATM transaction.

The rest of the paper is organized as follows: section II describes the system design of this project. In section III is concerned with explanation of the encryption process. Section IV provides the FPGA implementation of encryption system, while section V describes the results. Finally section 6 concludes this paper.

2. System Design
Cardless AES system design consists of Multi Touch Screen Display (MTL), Finger Print Scanner and keypad, AES algorithm. Figure 1 shows the block diagram of the AES cardless ATM. This system consists of multi touch screen (MTL) used as a display for the user to enter identification number, fingerprint scanner with keypad and fingerprint scanner as an input to ATM which implemented AES algorithm. For example, the user has to select the desired language before run the hardware. Users enter the identification number adhered by the fingerprint scanner.

Figure 2 shows the plaintext and cipher key as an input represent by keypad and fingerprint scanner, while ciphertext as an output display in MTL 2.

3. AES Algorithm
There are four transformations in the AES algorithm which is SubBytes, ShiftRows, MixColumns, and AddRoundKey used for encrypt identification number and finger scan. The encryption operation is illustrated in Figure 3. It contains a number of transformations applied sequentially on a data block in a fixed number of rounds (Nr). This Nr depends on the length of the encryption key.
SubBytes is a non-linear substitution of bytes that operates independently on each byte of the state using a substitution table (S-Box). The state is four rows of bytes that the internal operations of AES are performed on it. S-box in the SubBytes component consumes much of the total power in AES. There are various techniques to implement the S-box to satisfy criteria such as power, speed and delay for different applications either using 8 x 8 lookup tables (LUTs)[30] or using combinatorial logic directly from its arithmetic operations [31][32].

Shift Rows transformation makes the bytes in the last three rows of the state to be cyclically left shifted by 1, 2, and bytes for the 2\textsuperscript{nd}, 3\textsuperscript{rd}, and 4\textsuperscript{th} rows (the first row is not shifted). It proceeds as follows:

\[ S'_{r,c} = S_r, (c + \text{shift}(r, Nb)) \mod Nb \text{ for } 0 \leq r < 4 \text{ and } 0 \leq c < Nb \]

where Nb is number of bytes in each row of the state array, which is block length divided by 32. For Nb=4, shift(1,4) =1, shift (2,4) =2, shift (3,4) =3. ShiftRows operations are implemented by using thirty-two (32) 8-bit registers, or straightforward implementation, which requires a simple wiring between two 128-bit registers and consumes a large area in AES cipher. It is also unsuitable for small data path architecture.

MixColumn transformation is based on Galois field (GF) multiplication. Each byte of a column is replaced with another value that is a function of all four bytes in the given column. As a result of the MixColumn () transformation, the four bytes in a column are replaced by the following four bytes:

\[
\begin{bmatrix}
S_0, c' \\
S_1, c' \\
S_2, c' \\
S_3, c'
\end{bmatrix} = \begin{bmatrix}
&S_0, c \\
&S_1, c \\
&S_2, c \\
&S_3, c \\
\end{bmatrix} \bigoplus W_{i+c}
\]

where I round no. *Nb; O< c <Nb; Wi are the key generated words. The initial round key addition occurs at round 0 before the first application of the round function (l<S round < Nr).

A round key is an Nk words array obtained as follows: each byte of the previous round key is XORed with a constant that depends on the current round, and the result of the S-Box lookup for Wi, to constitute the next round key. The first round key is the original user key. The Nb, Nk and Nr for 128 bit AES is 4, 4 and 10, respectively.
4. Hardware Interface
The components involved in this project are: Fingerprint scanner, Multi touch screen, Keyboard. Following subsection describes each of the hardware implementation.

4.1. Fingerprint Scanner
A fingerprint scanner is used in the AES system where the module involved with the heavy lifting behind reading and identifying the fingerprints with an on-board optical sensor and 32-bit CPU. The fingerprint scanner could store the different fingerprints and the database of prints downloaded from the unit and distributed to other modules. User could also retrieved the image of a fingerprint and even pull raw images from the optical sensor.

Module used is model GT-511C1R which is more economical version of the GT-511 which has a less memory capacity (compared to the GT-511C3). The module could only store up to 20 different fingerprints but is capable of 360° fingerprint recognition and download and upload templates using serial interface.

The module is small and easy to mount using two mounting tabs on the side of the sensor. The on-board JST-SH connector has four signals: Vcc, GND, Tx, Rx. Simply connect the module to the computer using an FTDI Breakout and start the software to read fingerprints. Figure 4 shows the fingerprint scanner that used in AES system.

![Figure 4. Fingerprint Scanner](image)

4.2. MTL2
The new Terasic Multi-touch LCD Module Second Edition (MTL2) is an all-purpose capacitive touchscreen for FPGA applications and provides five points multi-touch gesture support. An Integrated Drive Electronic (IDE) cable with an IDE to GPIO (ITG) adapter is used to interface with various Terasic FPGA development boards through a 2x20 GPIO interface on the MTL2. Figure 5 shows the MTL2 hardware are used in AES system.

![Figure 5. Multi-Touch LCD](image)

4.3. PS/2 Keyboard
Keyboards consist of a large matrix of keys, all of which are monitored by an on-board processor called the keyboard encoder. The specific processor varies from keyboard-to-keyboard but they all basically do the same thing. Monitor which key are being pressed/released and send the appropriate data to the host. This processor takes care of all the debouncing and buffers any data in its 16-byte buffer, if needed. Motherboard contains a keyboard controller that is in charge of decoding all of the data received from the keyboard and informing the software of what is going on.
There are two different types of scan codes: make codes and break codes. A make code is sent when a key is pressed or held down. A break code is sent when a key is released. Every key is assigned its own unique make code and break code so the host can determine exactly what happened to which key by looking at a single scan code. The set of make and break codes for every key comprises a "scan code set". There are three standard scan code sets, named one, two, and three. All modern keyboards default to set two. Figure 6 shows the PS/2 keyboard hardware are used to implement in the system.

4.4. RS-232
Universal Asynchronous Receiver / Transmitter (UART) is one of the simplest forms of digital communication. UART allows users to send data serially, therefore only two wires are needed to send data to the computer, one for the data and a second one for ground. Information sent through UART is sent in packets which usually consist of one start bit, one stop bit, and a byte (8-bits) of data. For this project, data is sent from an NIOS II to a DE2-115. The process is done by writing VHDL code to transmit the data. The transmitter code is rather simple. The data that is being sent is shifted and assigned to the TxD output to send the data. A Baud rate of 9600 is being used here, which means 9600 bits are sent each second. To apply this, the clock inside the FPGA is divided as the FPGA has a 50MHz clock, by dividing it can create the equivalent of a 9600/second clock. It needed a shifter to shift each byte 9600 times per second, meaning need the equivalent of a 9.6kHz signal. The internal clock has a speed of 50MHz. The complete output and port numbers are shown in Figure 7. The port numbers are also marked on the RS232 header.

4.5. AES Hardware Implementation
The plaintext and cipher key will encrypt the data to access data and sent text to user from the algorithm will be successfully designed and implemented. Fingerprint Scanner as input for Cipher key for this project and also the Identification Number as input for Plaintext. Besides that, Multi-touch LCD as the display to give instruction to user and output display.

5. Results and Discussion
A complete VHDL code is written for encryption AES algorithm. The whole system is simulated in Quartus-II software. The simulation result verified the functionality of the system. The hardware implementation also carried out by implementing the LCD based signaling. The top module results is implemented on FPGA Cyclone IV. The encrypted message is sent to the DE2-115 and compared with the stored fingerprint image. The hardware also shows the proper functionality of the system.
5.1. AES Implementation

All sub blocks in the AES system were simulated using the Modelsim and were proven to proven correctly. The plaintext and cipher key used in this simulation (128-bit) are as follows:

Plaintext: 2b7e151628aed2a6abf7158809e4f3c
Cipher key: 6bc1bee22e409f9e93d7e117393172a

For graphical purposes, only the encryption process will be displayed in this chapter. Figure 8 illustrates the waveform simulation result for encryption process SubBytes transformation. This design is independent of the clock input signal and generated after the first plaintext and cipher key initial received. The output of SubBytes is:

After SubBytes: 090862bf628e3042c747feeda4a647

Figure 8. Waveform simulation for SubByte transformation

Waveform simulation for ShiftRow, Mixcolumn and AddRoundKey transformation is shown in Figure 9, where the 8-bit output from this sub-block after 10 clock cycles. Data output transformation is:

After ShiftRow: 09287f476f746abf2c46204da6e3ee
After Mixcolumn: 529f16c2978615ae01aae54ba1a2659
After AddRoundKey: 0bfabf406ee4d3042ca6b997a5c5816

Figure 9. Waveform simulation for others block transformation

After 10 rounds transformation the cipher key was executed as shown in Figure 10:

Cipher key: 3ad77bb40d7a3660a89ecaf32466ef97
The output was compare with AES Test Vector.

Figure 10. The Encryption Stage Inputs/Outputs

5.2 Performance Analysis

A throughput parameter of the AES system implementations have been analysed. Throughput denotes the speed of the encryption process. It can be defined as the number of encrypted or decrypted bits in a unit of time, or its average number of bits processed per second. Table 1 shows that the proposed AES-128 architecture required with 1% of the slices hardware resources of DE2-115. The increased
slices requirement will automatically affect the throughput per slice metric. The IOBs usage is approximately 44% because the input and output pins use bus widths. Nevertheless, this issue does not make any impact on the total throughput of the proposed architecture.

<table>
<thead>
<tr>
<th>FPGA</th>
<th>Min CLK</th>
<th>Max Freq.(MHz)</th>
<th>Latency</th>
<th>Throughput (Gbps)</th>
<th>Slices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclone IV</td>
<td>2.24</td>
<td>445.68</td>
<td>10</td>
<td>19/016</td>
<td>520</td>
</tr>
</tbody>
</table>

5.3 Hardware Implementation of AES System LCD Display
The LCD module has built-in fonts and can be used to display text by sending appropriate commands to the display controller. the 16x2 LCD is used to indicate the instruction to operate the AES system. Figure 11 shows the example instruction is display on 16x2 LCD module.

5.4 MTL2 Implementation
The GUI is used to implements a simple user interface. Before running the GUI, GUI should be vertically merged into 800x1440 interface and be stored in FLASH of the DE2-115 board in advance. In this GUI, the user can use the GUI by using single touch "move right" or "move left" gesture to select the previous or next step. Figure 12 shows the GUI implemented in the DE2-115 and display at MTL2. The GUI is programmed into DE2-115 using NIOS II Eclipse that can read C++ source Code.

5.5 Fingerprint Implementation
The fingerprint used 128x128 pixel images. While synthesizing the VHDL code, and it has used internal DE2-115 RAM by using NIOS II processor instead of on board memory, so the no. of LUTs used gone up to 90% from the initial 20% value. 128*128 size of image are transmitted to DE2-115 board via RS232 implemented on DE2-115 and display the pixel values on the LCD which are available on the DE2-115 board. Figure 13 shows the hardware implementation on the DE2-115 board.
5.6 Comparison with previous work

Table 2 show the comparison of this project with others researchers. The comparison is about the throughput and the slices. Implementation using Altera device achieved the lowest throughput compared when implemented in Xilinx device. The factor of difference maximum frequency, clock input on difference devices mitigate the throughput and slices. Compared to [7], even the throughput are the highest, but it required the biggest area of implementation. The proposed AES achieved a throughput of 19.016Gbps with 520 slices area utilization is acceptable when compared to [6] by using Altera device.

Table 2. Comparison of AES Implementation with difference devices.

<table>
<thead>
<tr>
<th>Author</th>
<th>Device</th>
<th>Latency</th>
<th>Throughput (Gbps)</th>
<th>Slices</th>
</tr>
</thead>
<tbody>
<tr>
<td>[3]</td>
<td>Xilinx Virtex5</td>
<td>10</td>
<td>22.054</td>
<td>273</td>
</tr>
<tr>
<td>[7]</td>
<td>Xilinx Spartan3e</td>
<td>10</td>
<td>60.043</td>
<td>2495</td>
</tr>
<tr>
<td>Proposed</td>
<td>Altera DE2-115</td>
<td>10</td>
<td>19.016</td>
<td>520</td>
</tr>
</tbody>
</table>

6. Conclusions

Biometric encryption system for ATM banking system using AES implementation is presented in this paper. AES is capable of safeguard against all known attacks. A number of design considerations have been taken in designing the ATM system. Proposed design implemented using FPGA offers a high performance and low area which very suited for restricted space environments. It has the highest throughput for feedback mode and second highest for non-feedback mode.

7. Acknowledgement

This work was financially supported by Universiti Tun Hussein Onn Malaysia (UTHM).

References


