Asian Journal of Scientific Research

ISSN 1992-1454

science alert
http://www.scialert.net

ANSInet
an open access publisher
http://ansinet.com

OPEN ACCESS Asian Journal of Scientific Research

ISSN 1992-1454
DOI: 10.3923/ajsr.2017.88.96

Research Article
FPGA Implementation of Rapid Ciphering and High Throughput of Smart Card Memory Ciphering System

[1]Wira Firdaus Yaakob, [1]Jahariah Sampe and [2]Noorfazila Kamal

[1]Institute of Microengineering and Nanoelectronics (IMEN), National University of Malaysia, 43600 Bangi, Selangor, Malaysia
[2]Department of Electrical, Electronic and System Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia (UKM), 43600 Bangi, Selangor, Malaysia

Abstract

Background: The advances of attack methods on the smart card now-a-days are getting more serious. It has encouraged researchers to put more effort in enhancing the data memory ciphering system in smart card memory management processing unit. **Materials and Methods:** In this study, there are three major units that constructs the system: Advanced Encryption Standard (AES) cipher block, Random Number Generator (RNG) key generation and scrambler/descrambler. This system is developed in the Memory Management Processing Unit (MMPU) of the smart card. By having the AES cipher unit, the plaintext from the Central Processing Unit (CPU) is encrypted or decrypted using a random key that is generated by the RNG key generation unit. The encrypted data also called as ciphertext is scrambled with the data from the scrambler/descrambler unit before being written into the memory during the write mode. Meanwhile during the read mode, the secured data from the data memory is descrambled with the data from the scrambler/descrambler unitinto the ciphertext. For memory types that allow for data reading only, e.g., ROM typically storing executable code, the process will be one way only i.e., descrambling and decryption. User Personal Identification Number (PIN) is utilized in the scrambling and descrambling processes. This prototype system is implemented in the Field Programmable Gate Array (FPGA) Xilinx's Zynq-7000 XC7020-1-CLG484. **Results:** The system is managed to complete the process within a a single cycle CPU that is about 40 nsec with 12002 Look-Up Table (LUT) slices, 3146 slice registers, a maximum frequency of 70.98 MHz and maximum combinational path delay of 0.471 nsec. The key finding of this study is that the system is capable to achieve throughput of 9085 (Mbits sec^{-1}) and 40 nsec ciphering time that are the best compared to the previous study. **Conclusion:** The proposed system is able to provide a secured data memory ciphering system for smart card with low resources, fast ciphering time and high throughput in the ARM-based FPGA Xilinx Zynq-7000 prototyping. The smart card is used in many applications including national identification (ID), financial security and health insurance.

Key words: Smart card, ARM-based FPGA, LUT, AES, RNG, ID

Received: November 12, 2016 Accepted: January 17, 2017 Published: March 15, 2017

Citation: Wira Firdaus Yaakob, Jahariah Sampe and Noorfazila Kamal, 2017. FPGA implementation of rapid ciphering and high throughput of smart card memory ciphering system. Asian J. Sci. Res., 10: 88-96.

Corresponding Authors: Wira Firdaus Yaakob and Jahariah Sampe, Institute of Microengineering and Nanoelectronics (IMEN), National University of Malaysia, 43600 Bangi, Selangor, Malaysia Tel: +603 89118156

Competing Interest: The authors have declared that no competing interest exists.

Data Availability: All relevant data are within the paper and its supporting information files.

INTRODUCTION

Smart card is very significant in modern life. There are so many applications that benefits from it such as national identification, medical history and money transactions. However, due to the precious information that is stored or processed in it, many attackers have put a lot of efforts to access the data[1]. Therefore, the security system in the smart card especially in the Memory Management Processing Unit (MMPU) has to be enhanced using state-of-the-art cryptographic algorithms and technologies and yet in an economic cost in order to secure the data and the whole smart card system. The MMPU is the unit that interfaces to smart card system to the memories.

The smart card architecture as illustrated in Fig. 1 consists of a CPU, internal and external memories and peripherals of analog, digital and mixed-signal design. The analog designs that are developed for the smart card security system at the physical level are RNG, voltage and light sensor. A Phase-Locked Loop (PLL) and voltage regulator are the mixed-signal design in the system, while UART, timer, Watch Dog Timer (WDT), Memory Management Processing Unit (MMPU) are the logic peripheral examples in the design. The memories that are used in the smart card are 64 kb Read Only Memory (ROM), 6 kb read/write Electrically Erasble Programmable Read Only Memory (EEPROM) and 256 bytes scratchpad Internal Random Access Memory (IRAM). Fast read/write access in the smart carrd system is achieved by having a 4 kb external RAM (XRAM). The IS07816-3 with UART modules is used as the communication protocol to communicate with smart card readeron a very low level[2].

The security system is implemented for logic circuit level and is developed in MMPU. The analog security circuit parts such as light and voltage sensors is implemented later in the Application Specific Integrated Circuit (ASIC) design implementation. The system consists of three major blocks, those are AES cipher, 128 bits RNG key generator and scrambler/descrambler. The AES cipher is used to encrypt the plaintext from CPU with the random key that is provided by the RNG key generator. The RNG unit is prototyped in FPGA using the logic pseudo RNG (PRNG) but then in ASIC implementation, the unit is replaced by a mixed-signal RNG IP. The ciphertext i.e., the encrypted data is then mixed with the scrambled data before being written into memory. On the reverse side, during the read mode, the data is read from the memory and is descrambled into ciphertext before decrypted into plaintext by the AES cipher. The scrambler/descrambler utilizes user PIN for scrambling and descrambling processes.

The system is prototyped in FPGA[3] to verify its functionality in real-time before being proceed to ASIC implementation[4]. Besides functionality, the hardware implementation results are compared with the previous study in terms of ciphering time and throughput. Smart card chip design industry is very competitive. Thus, rapid smart card design prototyping is very crucial. The fastest design time-to-market with a very low cost and best performance is very much required[5]. The other importance of the smart card design prototyping on the FPGA is for co-development hardware and software. The design can be customized at the early stage during development process. Hardware development consists of logic peripherals, memory, analog and logic interfaces. Software development comprises of

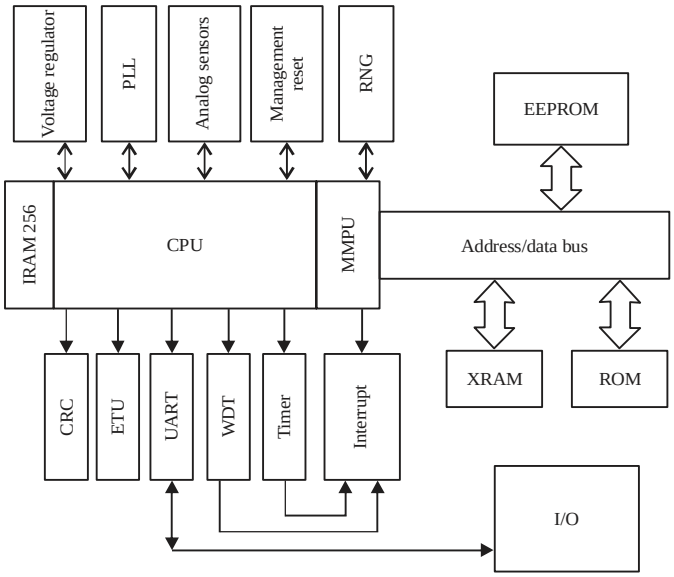

Fig. 1: Overall 8 bit CPU based smart card architecture

firmware and Operating System (OS). At last but not least, since the FPGA Zynq-7000 device is produced using deep submicron technology, therefore the ASIC implementation at the later stage using 180 nm CMOS technology will be much easier.

MATERIALS AND METHODS

Proposed smart card memory ciphering system: The complete processing cycle of the system as shown in Fig. 2 has been reduced from more than 20 clock cycles to a a single cycle CPU only that is approximately 40 nsec or 25 MHz. In the previous design, the 20 clock cycles are due to the encryption and decryption process of the AES-128 cipher and the remaining clock cycles are due to the scrambling and descrambling process[6].

AES-128 cipher unit: The AES-128 cipher is based on the AES Rijndael algorithm and is compliant to Federal Information Processing Standard Publications (FIPS PUB) 197. The cipher is a symmetric block cipher that can process plaintext blocks of 128 bits. Rijndael is a key-iterated block cipher, meaning that the initial block of plaintext and cipher key traverses through multiple rounds of transformation before generating the output[7]. A state is the intermediate result at the end of each round. There are 9 rounds of transformation for this 128 bits key AES. Each round consists of multiple operations like byte substitution, shift row, mix column and key addition. Byte substitution is the operation that runs independently on each byte of the state using a substitution table that is called as S-box. Shift row's operation runs by shifting bytes of the last three rows of the state cyclically while the first row is not shifted. Mix column's operation is the transformation process that runs on the state column-by-column. Each column is treated as a four-term polynomial. The final operation for each round that is key addition, operates by adding the round key to the state by a simple bitwise XOR. Other than key size 128 bits that is used in this study, Rijndael supports also for 192 and 256 bits key to encrypt plaintext blocks that are 192 and 256 bits, respectively.

The AES Rijndael cipher's algorithm as shown in Fig. 3 can perform very well in both hardware and software with a very low memory requirements that helps to avoid high cost design development due to the large memory requirement and slow processing performance. The key or plaintext block size decides the number of rounds. The inverse cipher is called as decryption. During decryption process, the encryption key schedules' form is the same as encryption but the sequence of the transformations is done in reverse order. The AES Rijndael is also adopted in the FIPS standard that is documented by National Institute of Stndards and Technology (NIST). The cipher has proven reliable due to its high computational complexity.

RNG key generator unit: The unit utilizes 128 bits of RNG data as the seed for cipher key generation. In the FPGA implementation, the RNG is prototyped as pseudo RNG (PRNG) using a Linear Feedback Shift Register (LFSR) with a fixed seed. However, in the ASIC implementation later, the PRNG unit is replaced with a mixed-signal design of a True RNG (TRNG) IP for the seed generation. The TRNG IP is a physical source for randomness that ensures the maximum guessing effort depending on the key length[8]. For security reasons, the generated key that is used for complementary encryption and decryption operations is randomly generated for every transactions.

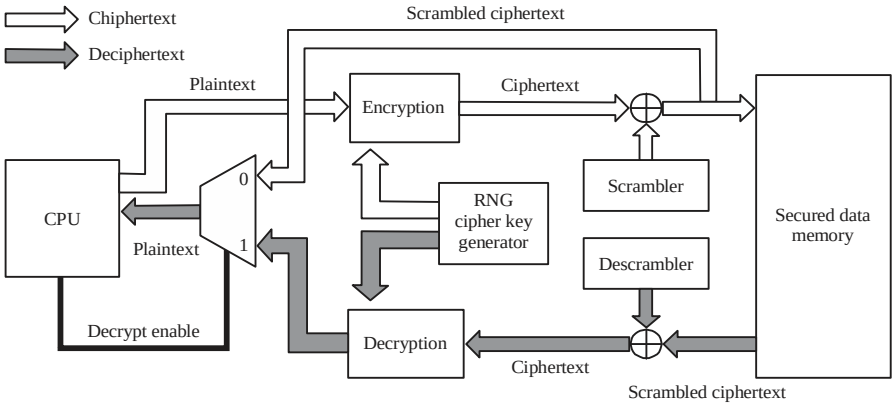

Fig. 2: Proposed memory ciphering system for smart card

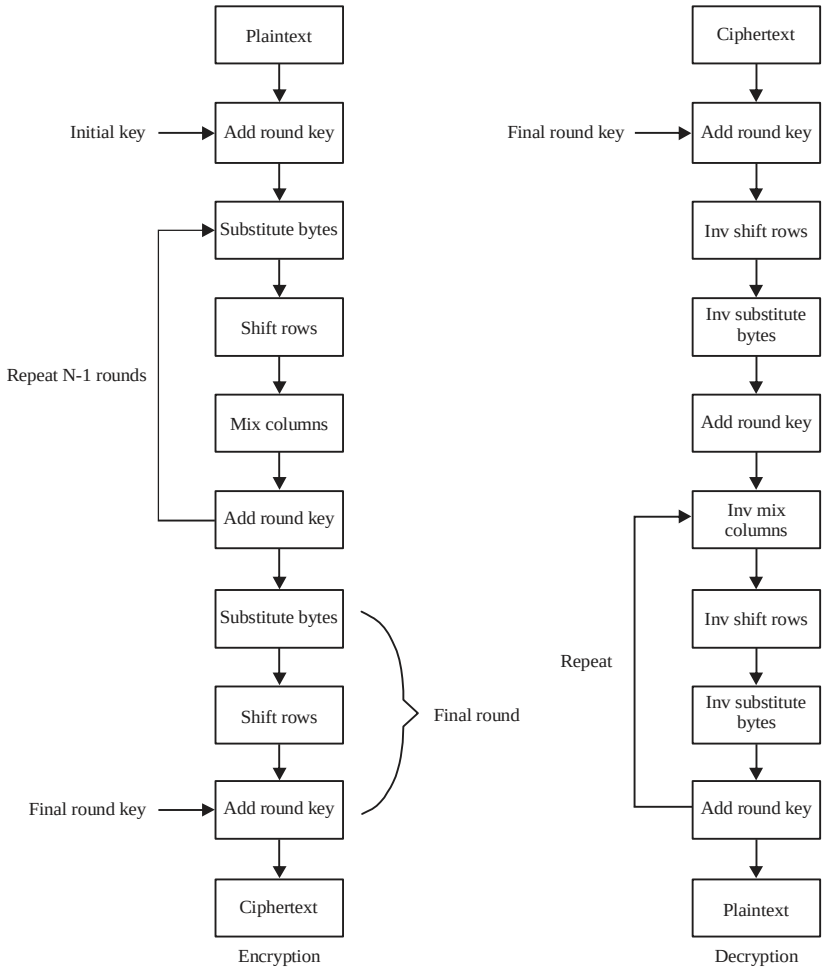

Fig. 3: AES-128 Rijndael algorithm

Scrambler/descrambler unit: The unit generates 128 bits scramble data that is mixed with the ciphertext in order to make it more difficult for the attacker to recognize the data from outside. The process of scrambling and descrambling requires user PIN input in order to provide more secured access. Without the correct PIN keys, the secured data from the memory cannot be descrambled into ciphertext and results the data cannot be accessed.

RESULTS

Simulation performance of the smart card security system: Figure 4 shows the complete process of the proposed ciphering system takes only a single cycle CPU that is approximately 40 nsec or 25 MHz. The complete process cycle starts from the plaintext encryption and ends with the decryption of the ciphertext. The signal kld_cipher is used as a flag to enable the encryption process and the signal 'done' is used to notify the encryption process is completed.

Figure 5 shows the decryption result of ciphertext. From Fig. 5, the decrypted ciphertext is similar with the original plaintext that is in hexadecimal value "01". The decrypted data and the plaintext value is highlighted with the circle. The signal 'done2' is a flag to notify the decryption process has been completed. The 'encrypted_mem' signal is the encrypted data or ciphertext that goes into the scrambler/descrambler unit.

Hardware implementation performance of the security system: The FPGA resource summary result of the smart card design implementation on Xilinx's Zynq-7000 XC7020-1-CLG484 is shown in Fig. 6. The slice registers that are utilized in the design is only 2% (3148) of the available slice registers and the slice LUTs is 23% (12620).

The timing summary report that is shown in Fig. 7 indicates the minimum period is 14.088 nsec that is 70.982 MHz maximum frequency and the maximum combinational path delay is 0.471 nsec.

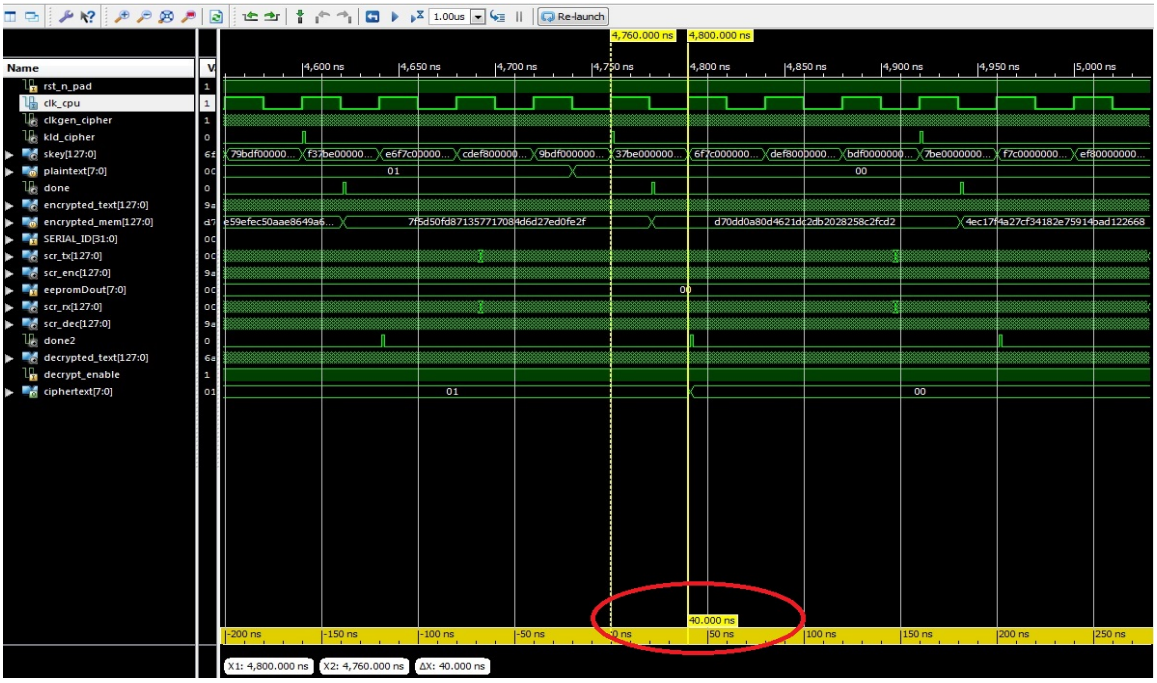

Fig. 4: Total ciphering time is a a single cycle CPU i.e., 40 nsec

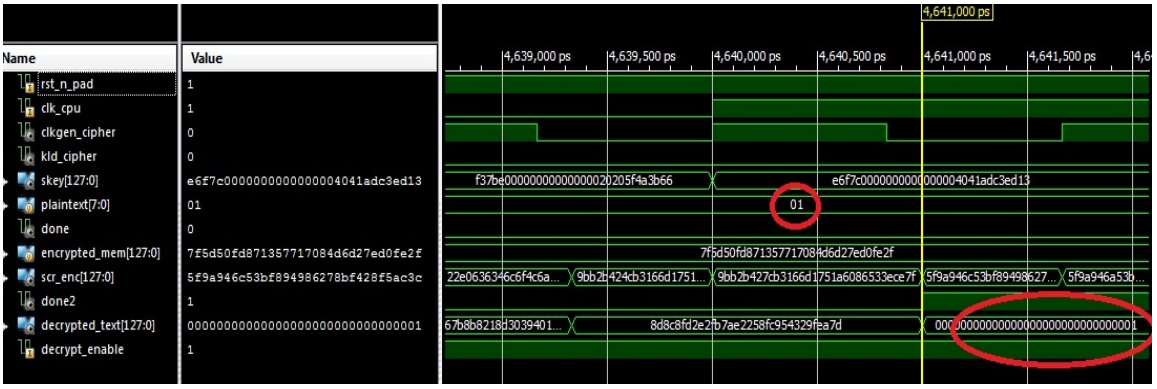

Fig. 5: Secured data is successfully decrypted to its plaintext value i.e., Hex 01

Table 1: Comparison between this study and previous studies[10-12]

Parameters	Bouesse *et al.*[10]	Bouesse *et al.*[11]	Zhang *et al.*[12]	Proposed study
Ciphering time (nsec)	910	595	294	40
Throughput (Mbits sec^{-1})	141	215	435	9085

In order to choose the best hardware performance of AES-128 based ciphering system for smart card, three previous different AES-128 ciphering systems as in Fig. 8 have been compared. Each of the ciphering systems' completion time and design throughput are measured and tabulated in Table 1. Equation 1 is the throughput calculation formula that has been used[9] in Table 1:

$$\text{Throughput} = \text{No. of output bits} \times \text{maximum frequency} \quad (1)$$

The results from Table 1 shows that the proposed ciphering system in this study has achieved the fastest ciphering time and the highest throughput followed by the parallel key expansion AES, optimized S-box Quasi Delay Insensitive (QDI) AES and conventional QDI AES.

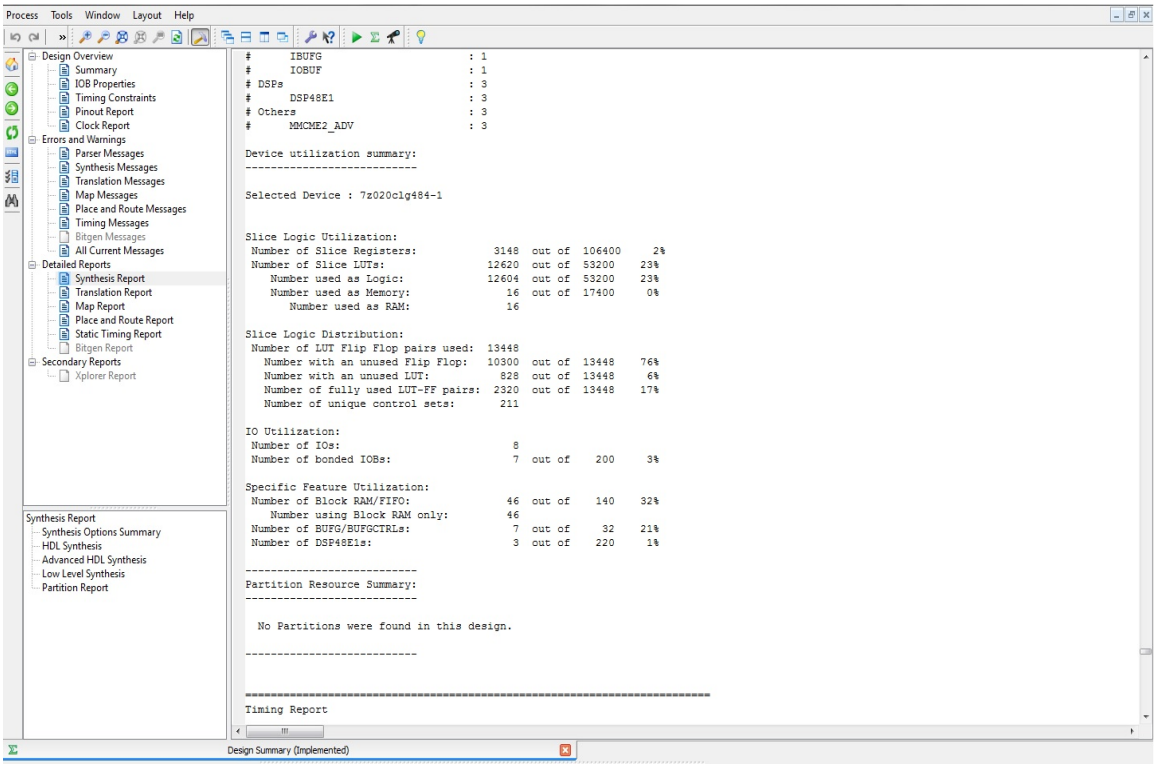

Fig. 6: Resource summary of the memory ciphering system

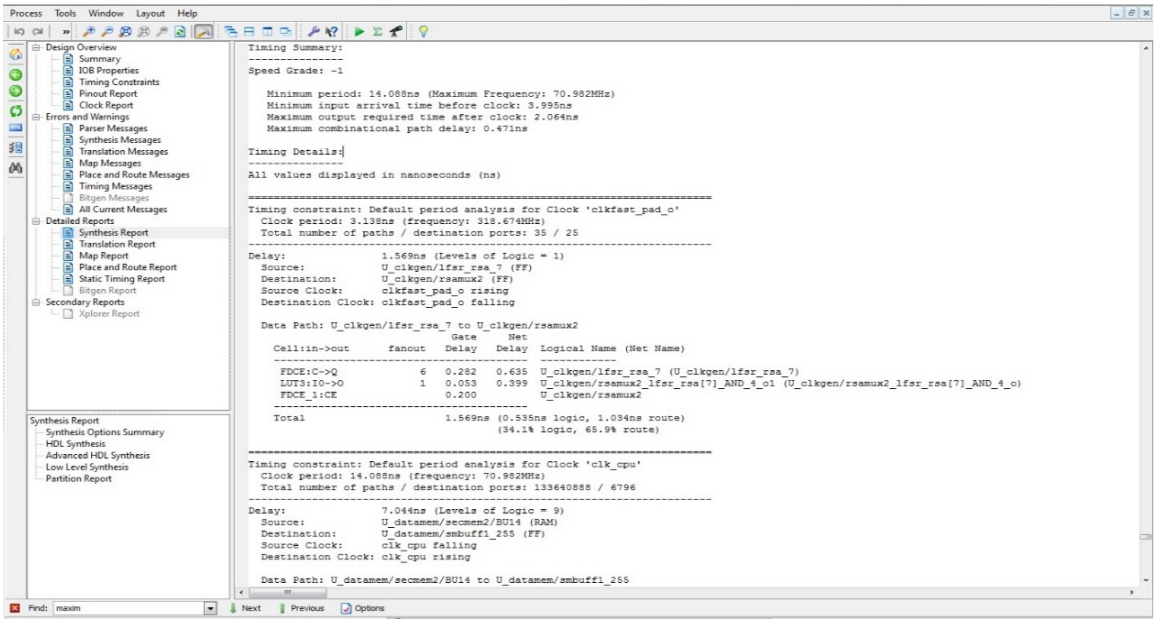

Fig. 7: Area and time report for memory ciphering system

DISCUSSION

The ciphering system for memory encryption is very crucial nowadays due to the capability for smart card memory protection[13]. Thus, this system is proposed in order to improve the existing ciphering system of a smart card for more security protection and hardware performance. The plaintext input from the smart card CPU is very easy to be recognized by an

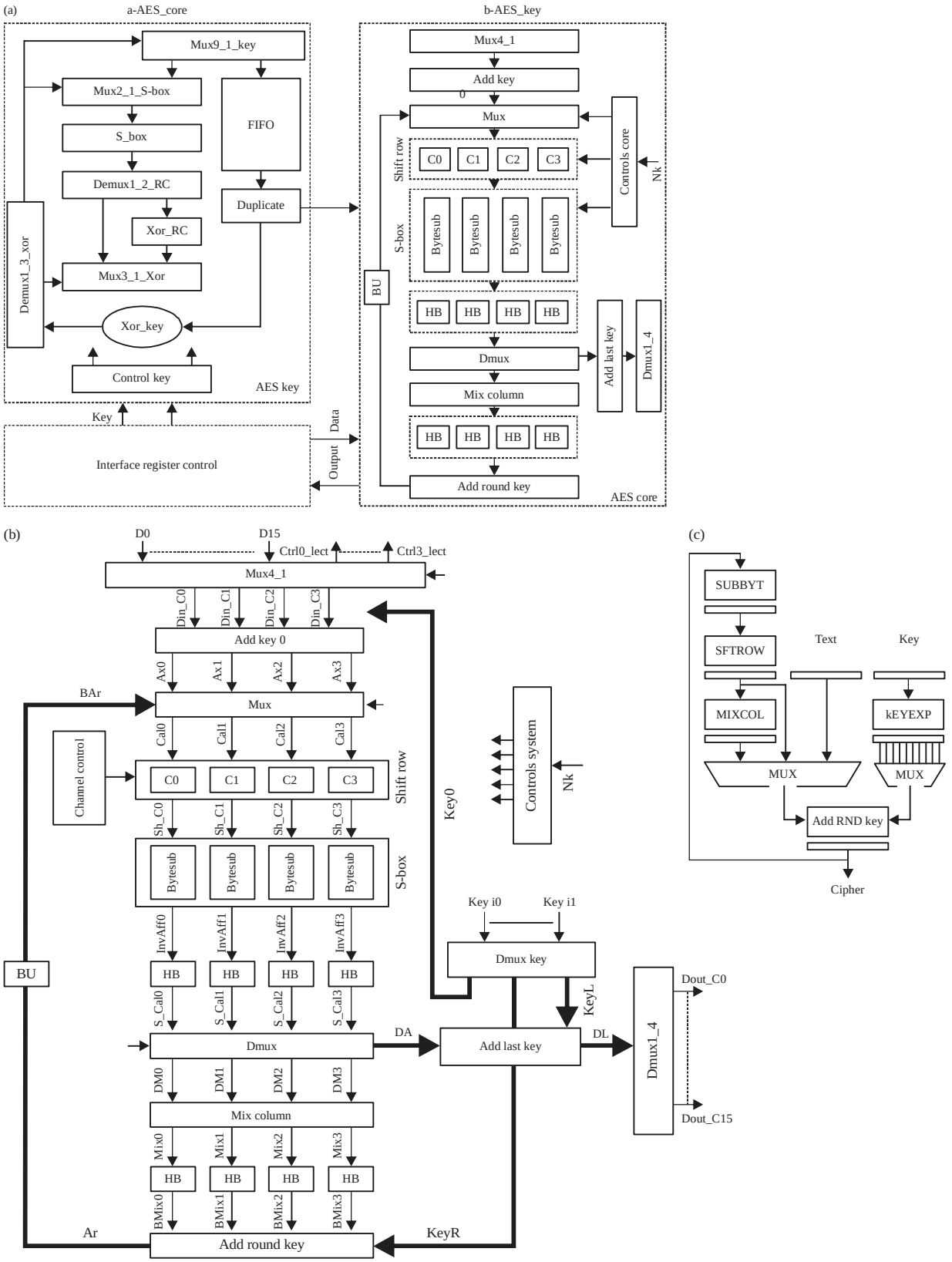

Fig. 8(a-c): Three different of previous ciphering system for smart card that have been compared (a) Conventional QDI AES, (b) Optimized S-box QDI AES and (c) Parallel key expansion AES

Fig. 9: Real-time testing of the ARM-based FPGA prototype smart card

attacker, thus, the FIPS-197 AES-128 has been utilized to encrypt the plaintext with the random key. The key is generated randomly for every transactions. The estimation of the number of keys that an attacker may attempt per second based on brute force attacks on AES-128 as follows:

- 1 personal computer results about 10^8 keys sec^{-1}
- 1 Graphical Processor Unit (GPU), 4×10^8 keys sec^{-1}
- 1 FPGA running with 200 MHz, 2×10^8 keys sec^{-1}
- 1 special device with about 2500 FPGA, 1.2×10^{11} keys sec^{-1}

The number of trial keys per second depend much on effectiveness of the cipher algorithm implementation[14]. However, the cipher algorithm is not enough. A very special machine can run more than one GPU at a time that will generate more trial keys. Hence, it is required to have the next logical level protection that is the scrambler/descrambler unit. Figure 5 shows the scrambled ciphertext i.e., scr_enc is generated for every half of the clock cipher cycle. The scrambling and descrambling process utilizes a user PIN key in order to generate the scrambled data. With a wrong PIN key, the scrambled ciphertext from the memory cannot be descrambled and decrypted.

Another crucial area for smart card design is the hardware implementation performance. It will not be competitive in the market if just having a very complex ciphering system but weak hardware performancein terms of ciphering time and throughput[15]. The issue is resolved with the proposed ciphering system that provides the right balance between the two major requirements as can be seen from the results. The real-time testing of the smart card on the Xilinx ARM-based Zynq-7000 FPGA is shown in Fig. 9. Figure 9 shows that the software is able to write and read data from the smart card memory that can be any format such as picture, text, audio and etc. The read and write process takes about less than a second. Therefore, the proposed memory ciphering system that is developed in the prototype smart card is working successfully.

CONCLUSION

A comparison among four types of AES-128 ciphering system for smart card have been carried out. There are conventional QDI AES, optimized Sbox QDI AES, parallel key expansion AES and a single cycle CPU synchronous AES. Result shows that the proposed ciphering system in this study that is based on a single cycle CPUAES achieves the best performance in terms of ciphering time and throughput. The system achieves 20 times higher and 7 times faster for the throughput and ciphering time, respectively than the previous ciphering system. The ciphering system process is not solely on the AES-128 encryption and decryption but includes also the scrambling and descrambling based on the user PIN key. The total smart card resource in the FPGA including the memory ciphering system is very low, that is about 3148 (2%)

and 12620 (23%) for slice registers and LUTs, respectively. Therefore, the proposed system is not only providing more protection for the data in memory, but also utilizing very low gate count that helps to get a competitive cost with a very high performance.

SIGNIFICANCE STATEMENTS

The advances of attack methods on the smart card nowadays are getting more serious. Therefore, this study proposes a secured data memory ciphering system for securing information transaction between a host and the internal smart card memory. The system has been implemented with low resources, fast ciphering time and high throughput in the ARM-based FPGA Xilinx Zynq-7000 prototyping. The proposed system has been compared with other previous studies in terms of ciphering time and throughput. The proposed system in the smart card can be used in many applications including national identification (ID), financial security and health insurance.

ACKNOWLEDGMENT

This study is funded by Ministry of Education Malaysia under grant FRGS/2/2014/TK03/UKM/02/1 and GUP-2015-021.

REFERENCES

1. Ege, B., E.B. Kavun and T. Yalcin, 2011. Memory encryption for smart cards. Proceedings of the International Conference on Smart Card Research and Advanced Applications, September 14-16, 2011, Leuven, Belgium, pp: 199-216.
2. Yaakob, W.F.H., H.H. Manab and S.N.M. Adzmi, 2014. Smart card chip design implementation on ARM processor-based FPGA. Proceedings of the IEEE 3rd Global Conference on Onsumer Electronics, October 7-10, 2014, Tokyo, Japan, pp: 294-297.
3. Sampe, J. and M. Othman, 2008. Hardware implementation of higher throughput anti-collision algorithm for radio frequency identification system. Am. J. Eng. Applied Sci., 1: 136-140.
4. Sampe, J. and M. Othman, 2008. Fast detection anti-collision algorithm for RFID system implemented on-chip. J. Applied Sci., 8: 1315-1319.
5. Mohammed, L.A., A.R. Ramli, V. Prakash and M.B. Daud, 2004. Smart card technology: Past, present and future. Int. J. Comput. Internet Manag., 12: 12-22.
6. Jyrwa, B. and R. Paily, 2009. An area-throughput efficient FPGA implementation of the block cipher AES algorithm. Proceedings of the International Conference on Advances in Computing, Control and Telecommunication Technologies, December 28-29, 2009, Trivandrum, India, pp: 328-332.
7. Tonde, A.R. and A.P. Dhande, 2014. Review paper on FPGA based implementation of Aadvanced Encryption Standard (AES) algorithm. Int. J. Adv. Res. Comput. Commun. Eng., 3: 4878-4880.
8. BSI., 2013. Methodology for cryptographic rating of memory encryption schemes used in smartcards and similar devices. Version 1.0, October 31, 2013, Bundesamt fur Sicherheit in der Informationstechnik (BSI), Bonn, Germany.
9. Soltani, A. and S. Sharifian, 2015. An ultra-high throughput and fully pipelined implementation of AES algorithm on FPGA. J. Microprocessors Microsyst., 39: 480-493.
10. Bouesse, G.F., M. Renaudin, A. Witon and F. Germain, 2005. A clock-less low-voltage AES crypto-processor. Proceedings of the 31st European Solid-State Circuits Conference, September 12-16, 2005, Grenoble, France, pp: 403-406.
11. Bouesse, F., M. Renaudin and F. Germain, 2004. Asynchronous AES crypto-processor including secured and optimized blocks. J. Integr. Circ. Syst., 1: 5-13.
12. Zhang, Q., J. Cao, D. Yu, X. Cao, X. Zhang, Y. Ye and B. Chen, 2015. A low-energy high-throughput asynchronous AES for secure smart cards. Proceedings of the IEEE International Conference on Electron Devices and Solid-State Circuits, June 1-4, 2015, Singapore, pp: 487-490.
13. Gilmont, T., J.D. Legat and J.J. Quisquater, 1999. Enhancing security in the memory management unit. Proceedings of the 25th EUROMICRO Conference, Vol. 1, September 8-10, 1999, Milan, Italy, pp: 449-456.
14. Savari, M. and M. Montazerolzohour, 2012. All about encryption in smart card. Proceedings of the International Conference on Cyber Security, Cyber Warfare and Digital Forensic, June 26-28, 2012, Kuala Lumpur, Malaysia, pp: 54-59.
15. Kaur, S. and R. Vig, 2007. Efficient implementation of AES algorithm in FPGA device. Proceedings of the International Conference on Computational Intelligence and Multimedia Applications, Volume 2, December 13-15, 2007, Sivakasi, India, pp: 179-187.