International Journal of Gender, Science and Technology

..ISSN: 2040-0748

UGC Care Group I Journal

Vol-12 Issue-01 June 2023

DESIGN AND IMPLEMENTATION OF AES ENCRYPTION-DECRYPTION USING ROM SUBMODULES AND EXCLUSION OF SHIFTROWS WITH 45NM

Dr.Y.Murali Mohan Babu

Professor, ECE Department
Chadalawada Ramanamma Engineering College, Tirupati, A.P, India
kisnamohanece@gmail.com

ABSTRACT: Cryptography is an essential process to achieve the purpose of secure communication, which is always the top priority for a user. Data saving requires information security. Many applications, such as biometric data-based recognition systems and health monitoring, require short-term data security. The Advanced Encryption Standard (AES) is recognized as a specific for electronic information encryption. This standard is used in both software and hardware and has become the most frequently accepted encryption method. AES provides such great resistance to both direct and differential cryptanalysis. The plan and execution of the AES encryption-decoding and the reduction of shift columns with 45nm using ROM sub-modules are covered in this method. Another advantage is that the architecture with ROM-based key development modules is claimed to perform better than the conventional registers and end-of-shift pushes that combine two stages of the calculation. This encryptor relocates the encryption process sub-steps before implementing an effective merging for them. This approach is executed with S-BOX inner pipelining and is divided among the primary round and key development units. Implementation (AED) using ROM sub-modules and ES using 45nm model is very efficient than other models it terms of number of nodes, delay, power consumption and Area.

KEYWORDS: AES Encryption, Shiftrows, ROM sub-modules, cryptography, Low power and area.

I. INTRODUCTION

As Internet of Things (IoTs) become more prevalent and utilized, it has become important to maintain their security and reliability [1].

IoTs that have been compromised have the potential to compromise the privacy and integrity of both the device and the server systems that support them [2]. The use of cryptographic systems is crucial for complete security. It is utilized to protect not only the sent data but also the system itself. Algorithmically secure cryptosystems are frequently utilized [3]. In general, when properly designed, hardware implementations of encryption for common security protocols are not only more energy efficient but also more difficult to attack than their software counterpart. However, if the cipher is not implemented carefully, it can leak information through side channel attacks, compromising the theoretical strength of the security protocol.

The study of cryptography allows for the confidentiality of communication through an unsafe channel. The two fundamental steps in a cryptographic operation are the encryption of plaintext to create a cipher-text and the decryption of the cipher-text to reveal the plaintext [4]. Symmetric algorithms are used for encryption when the sender and recipient share the same key, whereas asymmetric or public key cryptography is used when the sender and recipient utilize distinct keys [5]. Different techniques, including bank cards, wireless

phones, e-commerce, pay-TV, etc., use the cryptographic method for authentication. The access control in many systems, including carlock systems, escalators, metro trains, etc., also needs encryption and decryption.

A cryptographic process is utilized for confirmation in various applications, for example, bank cards, remote phones, web based business, pay-TV, etc[6]. In 2001 by the National Institute of Standards and Technology (NIST) the Advanced Encryption Standard (AES) was created. This symmetric block cipher is designed to take the place of DES as the accepted standard for a variety of applications [7]. The size of the key affects how many cipher rounds are used in AES. For 128-, 192-, or 256-bit keys, it is equivalent to 10, 12, or 14 respectively.

Although AES can be implemented in software, its hardware implementation adds additional actual security to standard execution. There are various ways to implement hardware, including ultra-high, medium, and low bandwidth architectures, as well as speed trade-offs. The SubBytes, ShiftRows, MixColumns, and AddRoundKeys transformations make up the private key encryption method AES. Data is encrypted using AES in 128-bit blocks. The AddRoundKeys step creates a 128-bit round key to XOR with the 128-bit data even though it can accept keys of 192, 256, or any combination of the three sizes. The number of rotations is 10, 12, and 14 for 128-bit, 192-bit, and 256-bit keys, respectively [8].

Each round comprises of the accompanying four stages for encryption:

• SubBytes (SB): To every byte of the ongoing 8-bit block to a non-direct 8-digit S-box a byte-wise transformation that applies.

• ShiftRows (SR): An immediate action that turns on all lines in a continuous network to the left.

• MixColumns (MC): One more quick move prompted a 4 x 4 cross segment. Each piece of the information network is replicated by the MixColumns structure in GF(28).

• AddRoundKey (AK): The direct XOR activity between the data cross section and the subkey of the ceaseless round.

Each round comprises of the accompanying four stages for decryption:

(a)InverseShiftColumns
(b) InverseSubstituteBytes
(c) AddRound keys
(d) InverseBlendSegments.

The output from the first two phases are XORing with the four words from the Addround keys key schedule in the third step. Take into consideration that the replacement and shifting procedures are carried out in a different order during a decryption round than they are during an encryption round. The remaining of the sentence is organized as follows. Section II discusses the literature review, and Section III describes the illustrated design and implementation of AES encryption-decryption using ROM sub-modules. The implementation is then described in Section IV. Finally, in Section V, draw conclusions.

II. LITERATURE SURVEY

D.H. Bui, E. Beigne, S. Bacles-Min, D. Puschini, and X. T. Tran et al. [9] Explains how to optimize hardware for a low-power,

high-speed AES architecture. To reduce area cost, the authors first utilized AES 32-datapath. Then, eight S-Box were used to increase throughput.
In order to save energy, a clock gating approach was lastly implemented in data storage registers. On ST FDSOI 28nm technology, the test-chip was validated. With a throughput of 28Mb/s, an operating frequency of 10MHz, and an energy consumption of less than 1pJ/b, it was able to achieve a power consumption of less than 20W for all significant configurations.

V. Hoang, V. Dao, and C. Pham et. al. [10] Explains the use of clock port technology to combine the best architectures to produce an ultra-low-power AES cryptographic core. The 65nm silicon used for this AES encryption core's construction uses SOTB (Silicon On Thin Buried Oxide) technology. Implementations claim that utilizing two S-boxes results in a minimal power consumption of 0.4 W/MHz and a reduction in the number of required clock cycles. Furthermore, the single SBox AES cryptographic core that is offered consumes only 2.4 kg of hardware resources (KGE).

W. Zhao, Y. Ha, and M. Alioto et al. [11] To save power, investigated portable AES accelerators. The quantity of accessible S-boxes in a lightweight AES scheme determines the number of encryption cycles. This AES architecture was implemented on a 65 nm test chip with a bandwidth of 376 kbps and energy of 0.83 pJ/bit at 0.32 V.

B. Buhrow, K. Fristz and E. Daniel et. al. [12] The AES-GCM Galois/Counter mode (Advanced Encryption Standard) was the basis for a new parallelization strategy. On a highly segmented network, this method can be used to create a scalable streaming core that can process multiple individually coded packets per clock cycle. It demonstrates how to utilize the technology to achieve throughput of 482 Gbps with a single Xilinx Ultra Scalable FPGA and more than 800 Gbps across multiple FPGA systems. Because the architecture does not require core-to-core communication, multi-FPGA systems are possible.

M. Mozaffari-Kermani and A. Reyhani-Masoleh et. al. [13] a simple AES concurrent error detection method is provided. The synthetic field is divided into S-box blocks and inverse S-box blocks using the proposed method, and the estimated parity of these blocks is found. An extensive search of all available complex fields found the optimal solution for an error detection scheme based on minimal parity. In addition, the total error coverage for 16 S-boxes (each inverse S-box) is nearly 100 percent, as demonstrated by simulations that introduce errors into one S-box (each inverse S-box).Finally, it has been demonstrated that the derivative best-fit synthetic field-based applied IC and field-programmable gate array implementations of the fault detection framework have lower hardware and time complexity than their counterparts.

L. Henzen and W. Fichtner et. al. [14] On reconfigurable hardware devices, offered a useful design methodology for implementing GCM with AES for authenticated encryption. Showed how to use 4 AES cores and 4 binary field multipliers to achieve 100 Gbps on an FPGA. Four pipeline steps are added to the GF(2128) multiplication to shorten the GHASH process critical path. The most recent GCM architecture, which runs at 119 Gbps on Xilinx Virtex-5 devices, is constructed from 44 layers.

C. Tokunaga and D. Blaauw et. al. [15] A block of switching capacitors is used to isolate switching activity and protect the AES engine by balancing the current drawn from the cryptographic core. While the non-sensitive blocks of the crypto engine are powered directly from the power source, the sensitive blocks of the engine are powered through a series of capacitors. The power supply supplies each capacitor, which is then separated while the encoder core is provided. The idea that the capacitor must be drained to a specified voltage before being charged in order to balance the amount of charge coming from the external power source is important.

III. AES ENCRYPTION-DECRYPTION

In this approach, AES Encryption-Decryption (AED) design and 45nm implementation with ROM sub-modules and Exclusion of Shiftrows (ES) are presented. There are several architectural changes made that improve the overall effectiveness of the cryptosystem while consuming less space and power. The device uses the Tannertool board to implement the proposed AES architecture for encryption and decryption. This method uses the Shift row operation to exclude rows. In general, AES performs the following transformations: SubBytes, MixColumns, ShiftRows, and the AddRoundKey.

A Shift Row can be excluded by calling the required shifted element from the data matrix instead of calling elements sequentially. As a result, Sub-Byte and Shift Row are combined into a single step. The AES algorithm's decreased step can be a justification and a means of lowering the overall hardware's space and power requirements. As a result, the design advances to a new level, increasing the system's overall effectiveness.

The round key that is generated is kept in the ROM module, not in a register. The initial input key is one of 11 round keys created by the key expansion procedure, each with 128 bits. Registers may be used to store information. A single bit register requires more space in this system than a single bit ROM does (Read Only Memory). 10x128-bit round keys are therefore stored in ROM in the proposed architecture. The ROM round key storage module is depicted in Figure 3. To create 10 round keys, 40 ROM sub-modules are used. Eight bits are spread over four locations in each ROM.

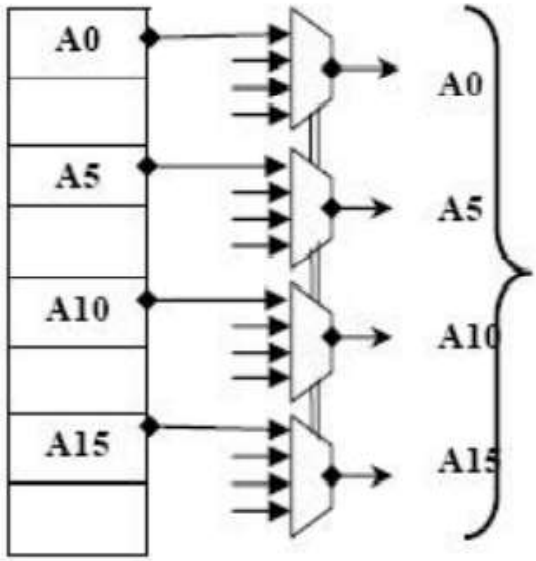

Fig. 1: SHIFTED ELEMENTS CALLING THROUGH MUX

In just one clock cycle, the RoundKey generator generates a key with four bytes. The AddRoundKeys module transmits round keys in four clock cycles and requires a total of 16 bytes. 40 ROM modules contain the round keys that the key generator produces. The ROMs provide 128-bit keys for the

encryption of the next block of data in four clock cycles.

The module that performs the MixColumn transformation is either effective or performs the following. The following operation is carried out using the four 8-bit input data X1, X2, X3, and X4.
OutByte=X01*02+ X2*03+ X3*01+ X4*01

This is easily accomplished with simple shifting and addition. Four of these modules were utilized to calculate four bytes in one clock cycle.

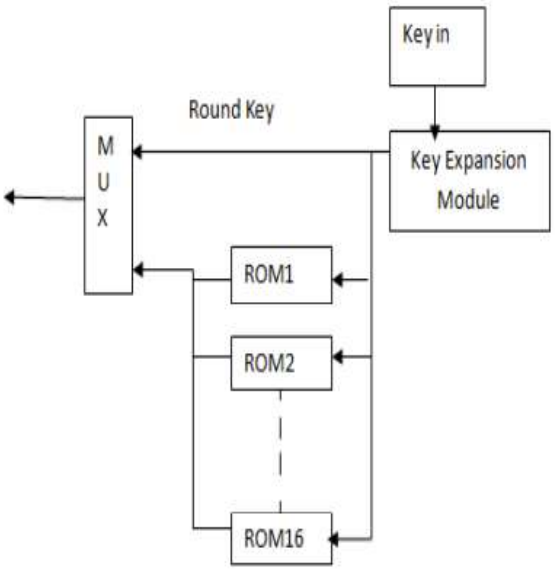

Fig. 2: ROUND KEY STORAGE IN ROM

IV. RESULTS AND DISCUSSION

The AES design is implemented in hardware using 45nm ASIC technology. Overall performance is equivalent to that of the traditional ROM-based key generation approach. Pipelining mode is used to implement the AES architecture. To complete each pipeline, 22 clock cycles are required. Sub Bytes transformation requires 16 clock cycles, whereas the S-box in ROM is used to access a single byte. To XOR four bytes with the key, which can be obtained in a single clock cycle from the ROM or directly from the key generator (in the case of the first block of data), the mixing module needs six clock cycles. 40 ROM modules hold the round keys that the key generator makes. ROMs are given in 4 clock cycles for block by block data encryption following a 128-digit key.

The waveform represents an AES encryption simulation using the suggested architecture. With 128 bits distributed across four inputs, each of which contains 32 bits of text and a cipher key, this demonstrates the effectiveness of the encryption process. The Tanner Tool programme may generate a device utilization report. Tanner S-Edit is a simple schematic capture and design entry environment. It provides the processing power required to manage the capture of your most sophisticated mixed-signal IC design. Tanner T-Spice, Analog FastSpice (AFS), and Eldo simulators are all firmly linked with S-Edit.

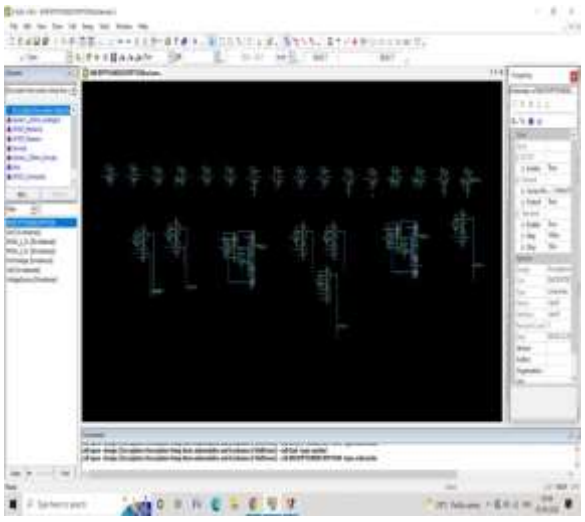

Fig. 3: SIMULATION ANALYSIS

Implementation (AED) using ROM sub-modules and ES using 45nm with the standard Differential Power Analysis (DPA)-resistant AES engine cell technology

library are compared in Table 1 in terms of delay, power consumption, and area. When compared to the standard DPA-resistant AES engine architecture, the proposed architecture achieves a delay improvement of approximately 0.9 ns. Additionally, the occupied power consumption of the proposed architecture is significantly lower than that of any other method, at 56.3 mWatta. It may be determined that the proposed architecture consumes less power than the other model even at higher clock frequencies. The area of described model is less (0.09 mm^2) than other model. Therefore from results it is clear that, implementation (AED) using ROM sub-modules and ES using 45nm model is very efficient than other models it terms of number of nodes, delay, power consumption and area.

Table 1: COMPARATIVE ANALYSIS

Parameters	DPA-resistant AES engine	(AED) using ROM submodules and ES
Number of nodes	22	14
Delay (ns)	26.2	12.1
Power consumption (mWatts)	154	56.3
Area (mm^2)	0.15	0.09

The graphical representation of above parameter as Delay, Power consumption and Area are shown in below Fig. 4, Fig. 5 and Fig. 6 respectively.

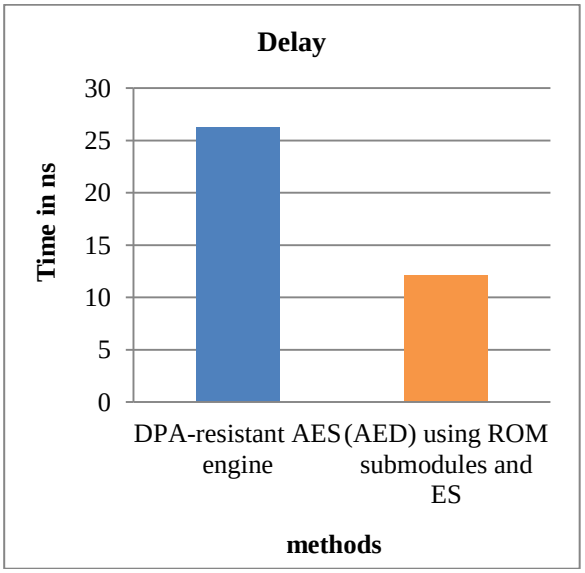

Fig. 4: PERFORMANCE ANALYSIS IN TERMS OF DELAY

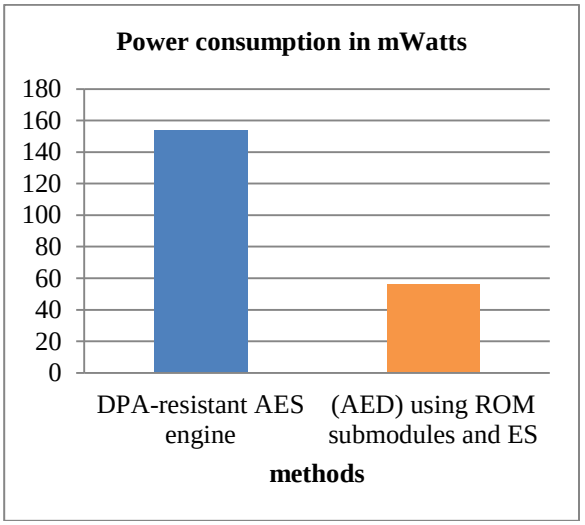

Fig. 5: PERFORMANCE ANALYSIS IN TERMS OF POWER CONSUMPTION

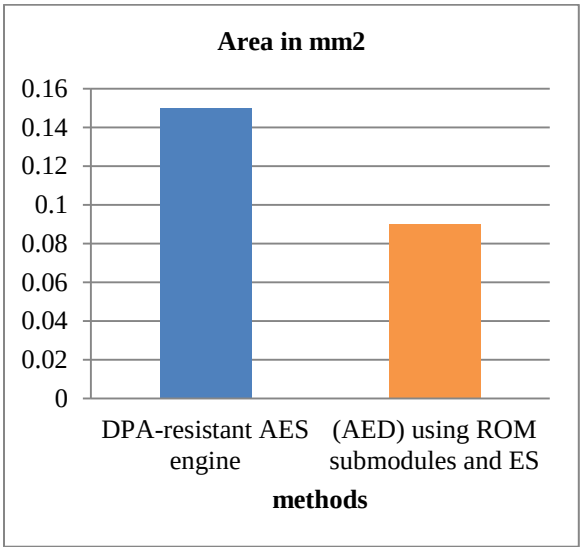

Fig. 6: PERFORMANCE ANALYSIS IN TERMS OF AREA

V. CONCLUSION

In this analysis, the design and implementation of AES Encryption-Decryption (AED) utilizing ROM sub-modules and Exclusion of ShiftRose (ES) with 45nm are presented. The S-BOX, which is shared by the primary round and key expansion units, is implemented by this architecture through internal pipelining. The shift row exception is handled with by calling the necessary shifted element from the data matrix. Sub-byte and shift row are combined into a single phase as a result of reduced. By using ROM rather than registers to store round keys, this method saves area and power. From results it is clear that, implementation (AED) using ROM sub-modules and ES using 45nm model is very efficient than other models it terms of number of nodes, delay, power consumption and Area.

VI. REFERENCES

[1] Dawei Wei, Huansheng Ning, Feifei Shi, Yueliang Wan, Jiabo Xu, Shunkun Yang, Li Zhu, "Dataflow Management in the Internet of Things: Sensing, Control, and Security", Tsinghua Science and Technology, Volume: 26, Issue: 6, Year: 2021

[2] Mohammad Nuruzzaman Bhuiyan, Md Mahbubur Rahman, Md Masum Billahm, Dipanita Saha, "Internet of Things (IoT): A Review of Its Enabling Technologies in Healthcare Applications, Standards Protocols, Security, and Market Opportunities", IEEE Internet of Things Journal, Volume: 8, Issue: 13, Year: 2021

[3] Sorin Chiţu, Daniel Ciprian Vasile, Ionuţ Daniel Trămândan, Paul Svasta, "Key Expansion in Cryptographic Systems", 2020 IEEE 26th International Symposium for Design and Technology in Electronic Packaging (SIITME), Year: 2020

[4] Yi-Fan Tseng, Jheng-Jia Huang, "Cryptanalysis on Two Pairing-Free Ciphertext-Policy Attribute-Based Encryption Schemes", 2020 International Computer Symposium (ICS), Year: 2020

[5] Wei Ou, Ning Fang, Hequn Xian, Wei Guo, Feilu Hang, Linjiang Xie, Xuyue Tang, "Accelerating Public Key Cryptography in Android Platforms", 2019 5th International Conference on Big Data Computing and Communications (BIGCOM), Year: 2019

[6] Hossein Kouzehzar, Meisam Nesary Moghadam, Pooya Torkzadeh, "A High Data Rate Pipelined Architecture of AES Encryption/Decryption in Storage Area Networks", Electrical Engineering (ICEE), Iranian Conference on, Year: 2018

[7] Veronica Ernita Kristianti, Eri Prasetyo Wibowo, Atit Pertiwi, Hamzah Afandi, Busono Soerowirdjo, "Finding an Efficient FPGA Implementation of the DES Algorithm to Support the Processor Chip on Smartcard", 2018 2nd East Indonesia Conference on Computer and Information Technology (EIConCIT), Year: 2018

[8] Liting Yu, Dongrong Zhang, Liang Wu, Shuguo Xie, Donglin Su, Xiaoxiao Wang, "AES Design Improvements Towards Information Security Considering Scan Attack**",** 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), Year: 2018
[9] D.H. Bui, D. Puschini, S. Bacles-Min, E. Beigne, and X. T. Tran, "AES Datapath Optimization Strategies for Low-Power Low-Energy Multi security-Level Internetof-Things Applications," IEEE Transactions on Very Large Scale Integration (VLSI) systems, vol. 25, no. 12, pp. 3281–3290, Dec. 2017
[10] V. Hoang, V. Dao, and C. Pham, "Design of ultralow power AES encryption cores with silicon demonstration in SOTB CMOS process," Electronics Letters, vol. 53, no. 23, pp. 1512–1514, 2017
[11] W. Zhao, Y. Ha, and M. Alioto, "AES architectures for minimum-energy operation and silicon demonstration in 65nm with lowest energy per encryption," in 2015 IEEE International Symposium on Circuits and Systems (ISCAS), 2015, pp. 2349–2352
[12] B. Buhrow, K. Fristz and E. Daniel, "A highly parallel AES-GCM core for authenticated encryption of 400 Gb/s network protocols," in 2015 International Conference on ReConFigurable Computing and FPGAs (ReConFig), December 2015
[13] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "A Lightweight High-Performance Fault Detection Scheme for the Advanced Encryption Standard Using Composite Fields," iEEE Transactions on Very Large Seale integration (VLSI) Systems, Vol. 19 (I), pp. 85- 91, Jan. 2011
[14] L. Henzen and W. Fichtner, "FPGA parallel-pipelined AES-GCM core for 100G Ethernet applications," in 2010 Proceedings of ESSCIRC, Sept 2010
[15] C. Tokunaga and D. Blaauw, "Secure AES engine with a local switched-capacitor current equalizer," 2009 IEEE International Solid-State Circuits Conference - Digest of Technical Papers, San Francisco, CA, 2009, pp. 64-65