A Comparative Study of AES Encryption Decryption

Gurpreet Kaur[1], Nishi Madaan[2]

[1]Student, M.Tech CSE, Computer Science, DAV University
Jalandhar, Punjab, India
[2]Assistant Professor, Computer Science, DAV University
Jalandhar, Punjab, India

Abstract: ***AES (Advanced Encryption Standard) is an effective encryption algorithm in applications like internet to provide cyber security and also in smart cards. In November 2001, the National Institute of Standard and Technology (NIST) of the United States choose the Rijndael algorithm as the suitable Advanced Encryption Standard (AES) to replace the Data Encryption Standard (DES) algorithm. AES is used in almost all network based applications to ensure security. The core computation of AES, which is performed on data blocks of 128 bits, is iterated for several rounds, depending on the key size. The strength of AES is proportional to the number of rounds applied. The number of rounds is fixed to 10, 12 and 14 for a key size of 128,192 and 256 bits respectively. The AES functional calculations include four transformation stages, which are Sub Bytes, Shift Rows, MixColoumns and AddRoundKey. In this paper, we have critically analyzed various AES techniques and also have covered AES overview such that it's Structure, Transformation Functions and Key Expansion.***

Keywords: AES, DES, Encryption Algorithm, Rijndael, FPGA, Throughput.

1. Introduction

Cryptography is the study of methods to transform information from its original comprehensible form into a scrambled incomprehensible form, such that its content can only be disclosed to some qualified persons. Cryptography ensures secure communications through confidentiality, integrity, authenticity and non-repudiation. Cryptography has evolved over the years from Julius Cesar's cipher, which simply shifts the letters of the words a fixed number of times, to the sophisticated RSA algorithm, which was invented by Ronald L.Rivest, Adi Shamir and Leonard M.Adleman and the elegant AES (Advanced Encryption Standard) cipher. Cryptographic algorithms used by nowadays cryptosystems fall into two main categories: Symmetric- key algorithms and asymmetric- key algorithms. Symmetric-key ciphers use the same key for encryption and decryption or the key used for decryption is computationally easy to compute given the key used for encryption. Cryptography using Symmetric ciphers are also can fall into two categories: block ciphers and stream ciphers. In contrast to block ciphers, which operate on a block of bits of a predefined length? Most popular block ciphers are DES, IDEA and AES and most popular stream cipher is RC6. Using symmetric- key cryptography, two parties who want to communicate confidentially must have access to the private key. In contrast with symmetric- key, the key used during encryption is distinct from that used during decryption in asymmetric-key algorithms. The encryption key is made public while the decryption key is kept secret. Within this scheme two parties can communicate securely as long as it is computationally herd to deduce the private key from the public one [6]. The large and growing number of internet and wireless communication user has led to an increasing demand of security measures and devices for protecting the user data transmitted over the insecure media [12]. A cryptographic algorithm is an essential part in network security [5]. National Institute of Standard and Technology (NIST) agreed on the block-based cipher system, data encryption standard (DES) in 1970. Since the DES cipher system was worn-out by violence attack methods, and then NIST announced the new DES algorithm, which is called the Triple DES (3DES). The 3DES uses the same algorithm as the DES and increases the difficulty of illegal breaks. But the 3DES algorithm exist two disadvantages. First, the 3DES requires three times more execution cycles than the DES, so the Execution efficiency of 3DES is not good enough. Second, both DES and 3DES only use a 64-bit length block data, the security and safety are not enough. Due to the two disadvantages of DES and 3DES, the NIST gave up the DES cipher systems and then asked for a new generation cipher system, which was called the advanced encryption standard(AES) in 1997[4]. In March 1999, the National Institute of Standard and Technology (NIST) organized the second Advanced Encryption Standard Candidate Conference (SAESCC) in Rome where a series of analyses of various algorithms were presented. This analysis includes evaluating not only their security capabilities, but also their performance, flexibility of implementation and other issues. Finally in October 2000, NIST announced the Rijndael as the winner algorithm for AES [13]. In November 2001, the AES was accepted as a standard of the Federal Information Processing Standards (FIPS). AES is regarded as the most reliable block cipher because there is no serious security flaws reported since it was released in 1999[3].

AES main computation is performed on a fixed block size of 128 bits with a key size of 128, 192 or 256 bits. This core computation is iterated for many rounds. The number of rounds depends on the key size. It is set to 10, 12 or 14 for the cited key sizes respectively. The resistance of AES against breaking attacks depends entirely on the number of rounds used. The best known attacks are on 7 rounds for 128-bit keys, 8 rounds for 192-bit keys, and 9 rounds for 256-bit keys [2]. Rijndael is a cipher that offers a good "combination of security, performance, efficiency, implement ability and flexibility"[1]. AES

implementations are needed for many applications such as PDA (Personal Digital Assistance), Wireless devices, embedded applications, HDTV (High Definition TV) and Video Conferencing [14].

2. AES Algorithm

AES is a block cipher algorithm with a block size of 128 bits.

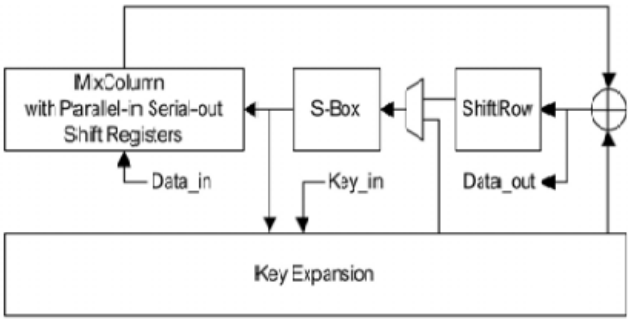

Figure1: Block diagram of the proposed AES encryption core architecture

The key size of AES can be independently specified to 128,192 or 256 bits. So called AES128, AES-192 or AES-256 respectively and accordingly there are 10, 12 or 14 iteration rounds to be performed for the encryption or decryption of a block [3][8]. The architecture operating on the intermediate result calls State, which is array of bytes. The array has four rows and four columns for the AES-128.

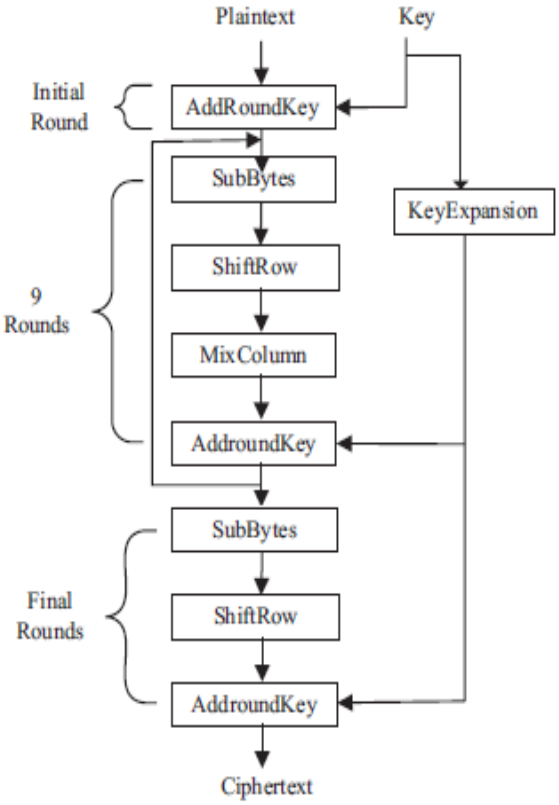

Figure 2: The AES-128 encryption algorithm

The details of these operations are as follow:

- **Sub Bytes phase:** Sub Bytes operation performs on each byte of the state using S-box which contains o permutation of all possible 256 8-bit values, which is a nonlinear substitution. It is the only non-linear transformation. The S-box is gained by a multiplicative inverse over GF (28) and an affine transform. However, the Sub Bytes operation is required for both encryption and key expansion. Its implementation has a direct impact on the overall throughput.

- **ShifRows phase:** ShifRow operation is the cyclic shifting of each row of the state to the left. The shifting numbers are depended on the number of the row. The top row is not shifted and the last three rows are cyclically shifted over 1, 2, and 3 bytes, respectively.

- **Mix Column phase:** Mix Column operation performs on the state column by column, treating each column as a four-term polynomial overGF (28). The operation of '_' is XOR operation modulo 2 and the '_' is a multiplication of polynomials modulo an irreducible polynomial m(x) _ x8 _ x4 _ x3 _ x _1.

- **AddRoundKey phase:** AddRoundKey is only a simple logical XOR of the state using a round key which is produced by the key expansion operation.

- **Key expansion phase:** The key expansion operation generates a key schedule. It generates a total of 11 round-key of 16 bytes. Each of four consecutive bytes form a word denoted i w, taking into account that the first round-key is the initial key. To generate every i w (except 0 3 w _ w) the routine uses the previous i_1 w XOR i_4 w (except imod 4 _ 0).To get the i w , when the imod 4 _ 0,the operation has four stages, Rot Word, Sub Word, XOR Rcon[i / 4] ,XOR i_4 w .The function RotWord takes a word[0 1 2 3 a ,a ,a ,a] as input, performs a cyclic permutation, and returns the word [1 2 3 0 a ,a ,a ,a]. Sub Word is a function that takes a four byte input word and applies the S-box to each of the four bytes to produce an output word [8].

3. Literature Review

In [1], Author presents Rijndael cipher system architecture and discusses the design trade-off among the various design choices to determine the optimum architecture and algorithm of the entire cipher system and functional units. A complete AES system can be divided into three major blocks: Key Expand, Control and EnDecrypt. Use of techniques such as unrolling and pipelining allows exploring the design space, tailoring the performance and area requirements. In order to explore the advantage of sub-pipelining further the Sub Bytes/ InvSubBytes is implemented by combinational logic to avoid the unbreakable delay of LUT's (Look-up tables) in the traditional designs. Using the proposed architecture, a fully sub-pipelined AES core with both inner and outer round pipelining and a two-sub stages in each round unit realized using Virtex-E devices can achieve a throughput of 30.88 Gbps at 241.313 MHz and 4626 CLB slices with 160

BRAM's in non-feedback modes, which is faster and more efficient.

In [2], Author proposed a novel hardware implementation of AES 128. The architecture allows one to perform the core computation of the algorithm in a pipelined manner. The throughput of the cryptographic hardware is more than 2Gbits per seconds. The pipelined execution of the AES algorithm allows an increase of the number of rounds without much loss of efficiency. Increasing the number of rounds applied, improves the resistance of the AES algorithm to cryptanalysis attacks.

In [3], Author examines the applicability of using a pipelined S-box in compact AES hardware implementations. A new VLSI architecture design for AES implementation is proposed to accommodate a four stage pipelined S-box. A new architecture design for compact hardware implementation of an AES encryption core is presented. The new design is featured with four stage pipelined S-box which is synthesized using synopsis design compiler version X-2005.09 under 0.18-um CMOS standard cell technology from TSMC through CMC Microsystems. The implementations results indicate that throughput increases 2.1 times and pipelined S-boxes are applicable to compact implementations of AES for the purpose of speed improvement.

In [4], Author develops FPGA –based high-throughput 128 bits AES cipher processor by using new high-speed and hardware sharing functional blocks. The AES functional blocks include Sub Bytes, Shift Rows, and Mix Columns and AddRoundKey transformations. The content-addressable memory (CAM) based scheme is used to realize the new proposed high-speed Sub Bytes block. The new hardware sharing architecture is applied to implement the proposed high-speed Mix Columns block. Next, the efficient low-cost AddRoundKey architecture is used for real-time key generations. The Xilinx ISE^{TM} 7.1 with XST^{TM} synthesizer is used as design tool. In proposed sequential AES design with both encryption and decryption, the operational frequency can reach 75.3 MHz and the throughput can be up to 0.876Gbits/s. both of the proposed sequential and full pipelined AES realizations achieve high throughput requirements and can be suitably used for the 802.11I CCMP applications.

In [5], Author presents an hardware implementation of the AES block cipher with Virtex II Pro FPGA using 0.13 um and 90 nm processor technology utilizing a high-speed parallel pipelined architecture increasing throughout for AES encryption algorithm. The chip contains the same ten units, and each unit can execute one round of the algorithm. Using external pipelined design, ten rounds of the algorithm are executed in parallel in a chip. Furthermore, using internal pipelining and key exchange pipelining, our implementation operating at 233 MHz achieves a throughput of 29.77 Gbps in encryption.

In [6], Author proposed a novel pipelined hardware implementation of AES-128 that can be used for both encryption and decryption and also shows that if the required number of rounds must increase to defeat attackers, the proposed implementation stays efficient. The architecture allows one to perform the core computations of the algorithm in a pipelined manner. The pipelined encryption and decryption allows an increase of the number of rounds without much loss of efficiency. A unique hardware is used for encryption and decryption. The hardware proposed is massively parallel and executes the four main steps of the algorithm in a pipelined manner, which allows a reasonable throughput to a little more of 1Gbs.

In [7], Author presented an equivalent AES architecture of efficiency optimized sub-pipelined optimized sub-pipelined and balance the throughput and resource consumption. These two implementations of AES using the composite field algorithm and Block RAM have to realize the Sub Bytes/ InvSubBytes module respectively. Based on the composite field algorithm, the throughput of AES can greatly enhance the throughput while the throughput/area rate of 6.744Mbps/slice. The second design's throughput/area rate achieved 7.19 Mbps/slice and the maximum frequency is limited to the maximum frequency of the Block RAM's.

In [8], Author presented an implementation of the AES-128 cryptographic algorithm using outer-round only pipelined architecture. Using Virtex-II Pro FPGA xc2vp70, the pipelined implementation of AES-128 is presented, Registers are inserted between rounds. Implementation presents 11-stage pipeline and every stage can execute one round of the algorithm. Using this external pipeline architecture, 11 blocks of data can be disposed at the same time. Throughput of 34.7 Gbps in encryption at the 233 MHz frequency is achieved. Two kinds of Block RAM is exploited, one for round of transformation, the other for key expansion.

In [9], Author proposed FPGA-based high throughput 128 bits AES cipher processor. An equivalent pipelined AES architecture working on CTR mode to provide the highest throughput up to date through inserting some registers in appropriate points making the delay shortest, when implementing the byte transformations in one clock period. Xilinx Foundation ISE^{TM} 10.1 FPGA design tool is used in the synthesis of the design. The proposed equivalent pipelined AES architecture provides the throughput of 73.737 Gbps. Clock frequencies of 576.07 MHz and resource efficiency of 3.21 Mbps/LUT. The proposed design research higher throughput than the other designs up to date and its resource efficiency is also very high. The design has so high throughput that it can be used to the developing high speed access network.

In [10], Author presented efficient sub-pipelined architectures of the AES algorithm, with unrolling and pipelining to explore the design space to tailor the performance and area requirements. Offline key expansion is used in order to reduce memory requirements and save power. Inner- round pipelining and scheduling allow high frequencies to be achieved and efficient uses of resources. In order to explore the advantage of sub-pipelining further, the Sub Bytes/ InvSubBytes is implemented by combinational logic to avoid the unbreakable delay of LUT's in the traditional designs. The resulting implementation has moderate area demands in term of

CLB slices, low memory requirements and achieves throughputs in the range of 26 Gbps. Compared to other academic and commercial implementations, the presented design demonstrated the highest throughput and one of the smallest memory/area to performance ratios. Optimization approaches for the implementations supporting multiple key lengths and modes of operation require further study.

In [11], Author proposed simulated sub-pipelined ASIC design for AES in CMOS 180nm. The loop itself is sub-pipelined into stages by placing registers in locations where the signal propagation in the stages are comparable. In the optimization of the compact loop there are trade-offs between increasing hardware complexity, over consumption and throughput. In the new optimized structure, increasing the gate count by 25.8% resulted in more than 158% increase the data throughput.

In [12], author presented a high throughput, area efficient implementation of the AES algorithm. The complexity of the S-box is greatly reduced by a basis transformation from Gf (2^8) to GF (2^4). There is a 64% area reduction in S-box and about 50% total area reduction as compared with the previous LUT approaches. The pipelined AES chip provides a very high throughput while keeping the area small. In addition, the design can also perform key expansion on-line. With the standard on-chip bus interface, the AES cipher can be plugged into the system chip easily. Finally, testability has been stressed in the proposed design.

In [13], Author implements a fast fully pipelined architecture for the AES encryption method is implemented that is suitable for securing data exchange in real-time applications such as video encryption. A brief analysis of AES implementations and a testable unrolled pipelined implementation for AES is presented. The design is synthesized using a 0.35 um ASIC library, for which a delay less than 20 ns is extracted for each pipeline stage of the design. Therefore it can achieve a maximum throughput of 6 Gbps. With the added BIST architecture test coverage of about 98% is obtained.

In [14], author implements FPGA and multicore processor, parallelization to satisfy the timing constraint. So that better throughput has been achieved. However software techniques has wide acceptance because of its implementation flexibility. From the simulation results, author identifies that hardware techniques are efficient over the software techniques in examining the parameters such as execution time, frequency and throughput. Further, the performance can be increased by integrating Open MP with SIMD (Single Instruction Multiple Data).

In [15], Author presents a fully synchronous, memory-based, single-chip FPGA implementation of the recent AES Standard, Rijndael encryption algorithm. Design partition allowed for an iterative loop structure where the block ciphers was implemented using the Electronic Code Book (ECB) mode of operation. The encryption RTL design focuses on a memory-based bite-sized arithmetic pipeline structure that processes one round at a time. The dual-width encryption pipeline permits the incorporation of a 32-bit DSP core into the byte size data path while reducing latency issues associated with DSP cores having smaller bandwidths. The RTL design targeted the Xilinx Virtex-II Pro FPGA's. The high-speed design provides a throughput rate of 0.1 Gbps per data channel. As the ECB mode permits easy parallelization higher performance yield can be achieved without significantly larger chip pin-outs.

4. Critical Analysis

Table1. Comparison of Advanced Encryption Techniques

Reference Number	*Technology Used*	*Advantage*	*Throughput*
[1]	Virtex-E devices along with LUT and Block RAM.	High Throughput, Sub-pipelining.	30.88Gbps.
[2]	Four 128 registers are used.	Secure	More than 2Gbits/s.
[3]	Compiler version X-2005.09 under 0.18-um Cmos standard cell technology	Throughput increases 2.1 times, low area required.	2.1 times increase.
[4]	FPGA Xilinx ISE™ 7.1 with XST™ synthesizer.	Higher throughput than other sequential and full pipelined design.	0.876Gbits/s in proposed sequential AES design. 32Gbits/s in full pipelined AES design.
[5]	Virtex-II Pro FPGA device is used.	High speed.	29.77Gb/s.
[6]	128 registers are used.	Secure.	1Gbps.
[7]	Xilinx Virtex-4 xc4v1x100 device is used.	Low cost, high speed.	82.65Gbps.
[8]	Block RAM is used to store S-box.	Higher efficiency.	34.7Gbps.
[9]	Xilinx Foundation ISE™ 10.1 FPGA design tool is used.	Higher throughput, resource efficient.	73.737Gbps.
[10]	Virtex-E devices	Compact, memory less, high-speed hardware architectures, combined data path, resource sharing, logic optimization, increased speed.	26.64Gbps.
[11]	CMOS 180nm technology.	High speed, low power delay, throughput is increased more than 158%.	6Gbit/s.
[12]	0.25 Um	High throughput	2.977Gbps for

	CMOS technology is used.	rate, area efficient, complexity of S-box is reduced, perform key expansion on-line.	128-bit keys, 2.510Gbps for 192-bit keys and 2.169Gbps for 256-bit keys.
[13]	LUT, BIST (Built in self test) technique is used.	Higher speed, low area cost, low cost of S-boxes.	6Gbps.
[14]	Xilinx xc5vlx110t-1 device is used.	Best trade-off between parallel software and pipelined hardware execution, performance is increased.	31.25Gbps.
[15]	Xilinx Virtex-II Pro device is used.	High speed design, easy parallelization, higher performance.	0.1Gbs/data channel.

5. Conclusion

In this work, we have presented an implementation of the AES-128 cryptographic algorithm using outer-round only pipelined architecture. A new architecture design for compact hardware implementation of an AES encryption core is presented. The new design is featured with a 4-stage pipelined S-box. The implementation results show that, compared with the previous smallest encryption-only AES hardware implementation, the new design uses the same amount of gates to achieve an increase of 2.1 times in throughput. The implementation results indicate that pipelined S-boxes are applicable to compact implementations of AES for the purpose of speed improvement.

References

[1] Nalini C, Nagaraj, Dr. Anandmohan P.V & Poornaiah D.V, V.D. Kulkarni, "An FPGA Based Performance Analysis of Pipelining and Unrolling of AES Algorithm", IEEE ADCOM2006,pp. 477-482.
[2] Nadia Nedjah, Luiza de Macedo Mourelle, Marco Paulo Cardoso, "A Compact Pipelined hardware Implementation of the AES-128 Cipher", IEEE ITNG2006,pp. 216-221.
[3] Cheng Wang and Howard M. Heys, "Using a Pipelined S-Box in Compact AES Hardware Implementations", IEEE NEWCAS2010, pp. 101-104.
[4] Chih-Peng Fan and Jun-Kui Hwang," Implementations of High Throughput Sequential and Fully pipelined AES Processors on FPGA,"IEEE ISPACS2007, pp-353-356.
[5] Deen Kotturi, Seong-Moo Yoo, and John Blizzard, "AES Crypto Chip Utilizing High-Speed Parallel Pipelined Architecture", IEEE ISCAS2005,pp. 4653-4656 vol.5.
[6] Nadia Nedjah, Luiza de Macedo Mourelle,"A Versatile Hardware for Advanced Encryption Standard", http://www.dynamicpublishers.com/JIAS/arcs/a6.pdf, Journal for Information Assurance and Security, Dynamic Publishers, 2006,pp. 51-58.
[7] Dong Chen, Gouchu Shou, Yihong Hu, Zhigang Guo, "Efficient Architecture and Implementations of AES", IEEE ICACTE2010, pp. V6-295-V6-298.
[8] Yulin Zhang, Xinggang Wang, "Pipelined Implementation of AES Encryption Based on FPGA", IEEE ICITIS 2010, pp. 170-173.
[9] Shanxin Qu Guochu Shou Yihong Hu Zhigang Guo Zongjue Qian," High Throughput, Pipelined Implementation of AES on FPGA", IEEE IEEC2009, pp.542-545.
[10] Nalini C. Iyer, Anandmohan P.V, Fellow IEEE, Poornaiah D.V, and V.D. Kulkarni, Member, IEEE, "High Throughput, low cost, Fully Pipelined Architecture for AES Cryto Chip", IEEE IC2006,pp.1-6.
[11] A.Alma aitah and Zine-Eddine Abid, "Area Efficient-High Throughput Sub-Pipelined Design of the AES in CMOS 180nm", IEEE IDT2010, pp. 31-36.
[12] Chih-Pin Su, Tsung-Fu Lin, Chih-Tsun Huang and Cheng-Wen Wu," A Highly Efficient AES Cipher Chip", IEEE ASP-DAC2003,pp.561-562.
[13] Mahdi Nazm-Bojnordi, Naser Sedaghati-Mokhtari and Seid Mehdi Fakhraie, "A Self-Testing Fully Pipelined Implementation for the Advanced Encryption Standard", IEEE ICM2005, pp. 260-263.
[14] J.Saira Banu, M. Vanitha, Dr. J.Vaideeswaran, Dr.S.Subha, "Loop Parallelization and Pipelining Implementation of AES Algorithm Using OpenMP and FPGA", IEEE ICE-CCN2013, pp. 481-485.
[15] Kenneth Stevens, Otmane Ait Mohamed, "Single-chip FPGA Implementation of a Pipelined, Memory-Based AES Rijndael Encryption Design", IEEE ECE2005,pp. 1296-1299