AN IMPROVED HIGH SECURE COMMUNICATION USING AES WITH S.R AND M.C

[1]SHAIK MOBEENA, [2]T.BHAVANA
[1]M.tech-Scholar, Dept of ECE, Guntur Engineering College, Guntur, A.P, India.
[2]Assistant Professor, Dept of ECE, Guntur Engineering College, Guntur, A.P, India.

ABSTRACT: In recent years, Cyber Security has become rising issue where encryption is one of solution and has a significant role in protecting the data. Encryption algorithms are widely utilized in information security are asymmetric and symmetric. Now a day's Advanced Encryption Standard (AES) is one of symmetric encryption that most utilized often and the most secure encryption. S-boxes are used for the implementation of the multiplicative inverses and shared between encryption and decryption. However, AES encryption has slow computation. A novel method to improve AES algorithm with Shift Row and S-Box modification for Mix Column transformation is proposed in this paper.

Keywords: Advanced Encryption Standard (AES), Optimization; S box.

I. INTRODUCTION

Various techniques, such as cryptography, steganography, watermarking, and scrambling, have been developed for keeping data secure, private, and copyright protected. Cryptography is a necessary tool underlying virtually all networking and computer protection, traditionally utilized for military and espionage. Hence, there is a need for secure transactions in ecommerce, private networks, and secure messaging has moved encryption into the commercial realm. Advanced encryption standard (AES) was provided as Federal Information Processing Standards (FIPS) by National Institute of Standards and Technology (NIST) as a successor for data encryption standard (DES) algorithms.

A number of architectures for the VLSI implementation of AES Rijndael algorithm are reported in recent literature. It can be observed that some of these architectures are of low performance and some provide low throughput. Further, many of the architectures are not area efficient and can result in higher cost when implemented in silicon.

In this paper, a high throughput, high performance and area efficient architecture of VLSI for Rijndeal algorithm is proposed which is suitable for low cost silicon implementation. For high throughput in the data rates of encryption and decryption we optimize the proposed architecture by utilizing pipelining. The multiplication of Polynomial is introduced by utilizing XOR operation instead of utilizing multipliers for decreasing the complexity of hardware. In the proposed architecture both the modes of encryption and decryption utilizes common hardware resources, hence making the design is area efficient.

Selective utilization of combinational logic and look-up tables further reduces the architecture's memory optimization, area and performance. Our proposed architecture has a significant feature which is an effective solution of online (real-time) round key generation needing significantly less storage for buffering.

II. LITERATURE REVIEW

As a non linear component with confusion effect in block cipher algorithms, the s-box appears repeatedly in the round transformation and key expansion algorithm. It plays a key role in the security of the entire cipher algorithms. The anti-attack capability of various block ciphers is related to S-box cryptography characteristics of the entire algorithms. Therefore, it is a prerequisite of the design for analyzing S-box cryptography characteristics.

We know the number of S box used in the block ciphers is more, the nonlinearity of the cryptographic algorithms is higher, and the confusion of the cipher algorithm is stronger. In fact, the bigger S box, which is used in the hardware structure, the calculation, check list and storage are also required more time and space. In this case, the algorithm becomes very low efficient. So how to choose a good S-box, we need to consider the security of algorithm and the work efficiency of implementation. The traditional method generates the S box, which has a good index of cryptography. However, it is difficult to use pure logic hardware implementation consuming a large number of logic units. What's worse, the use of look-up table costs too much memory resource. Ideally, the method needs 16 clock cycles to complete the transformation, which is not conductive to the high-speed implementation of encryption system.
The new S box proposed by Wu Wenling et al. is implemented by a logic unit AND an OR gate. Although S box generates with fast speed, cryptography strength is not as strong as traditional S-box generation algorithm. After research, we propose an improved algorithm that is the dynamic S-box algorithm effectively compensating for indicators on cryptography. The improved algorithm can be not only satisfied with application for the advanced encryption algorithm AES, but also reduce hardware logic units and improve the speed of encryption and decryption.

We employ the new S box proposed by Wu Wenling et al. as S box of AES encryption system, and do some extension on the basis of the S box. Utilizing the S box to generate time-shift bits selection function can get 8 groups of S boxes. Meanwhile, exchange position of the input high 4 bits and low 4 bits, then we can get another new 8 groups of S boxes, whose cryptography property is not changed. Finally, we construct 16 groups of S boxes, between which exists linear relationship. So we realize extension from one set to many sets of S boxes. Dynamic S box is used for the key expansion module, byte conversion module and byte inverse transformation module, which improve the encryption strength.

III.PROPOSED SYSTEM

The Advanced Encryption Standard (AES) is a symmetric block cipher which is utilized by the U.S. government for protecting information which is classified and is performed in software and hardware throughout the world for sensitive data encryption.

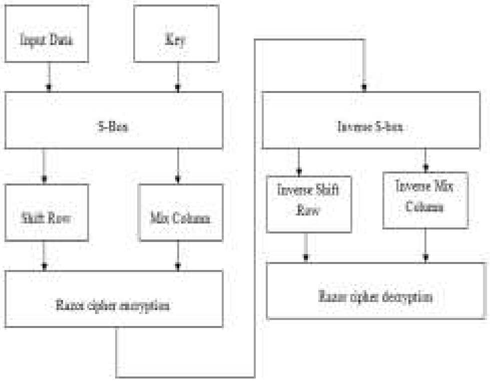

FIG. 1 PROPOSED BLOCK DIAGRAM

AES is actually, consists of 3 block ciphers, AES-128, AES-192 and AES-256.

Each cipher encrypts and decrypts the data by utilizing cryptographic keys of 128 bits, 192 bits and 256 bits, respectively in blocks of 128 bits. 14 rounds for 256-bit keys, 12 rounds for 192-bit keys and 10 rounds for 128-bit keys are there in advanced encryption standard.

A. AES Encryption

To implement the AES encryption algorithm, we proceed exactly the same way as for the key expansion, i.e, we initially implement the basic helper functions and then move up to the main loop. The functions considered as parameter a state, which is, as already explained, a rectangular 4x4 array of bytes. The shift Rows function iterates over all the rows and then call shift Row with the correct offset. Shift Row does nothing yet for shifting a 4-byte array by the given offset.

This is the part that involves the round Key which was generated during each iteration. Here simply XOR each byte of the key to the respective byte of the state. The Mix Columns implementation was carried out by initial one would generate a column and then call mix Column, which would then apply to the matrix multiplication.

One AES round is which has to apply all four operations consecutively on the state. All we have to do is take the state, the Expanded Key and the number of rounds as parameters and then call the operations one after the other. Finally, all we have to do is put it all together. Our parameters are the input plaintext, the key of size key Size and the output. First, we calculate the number of rounds depends on the key Size and then the expanded KeySize based on the number of rounds. Then we have to map the 16 byte input plaintext in the correct order to the 4x4 byte state expand the key by utilizing our key schedule, encrypt the state by utilizing our main AES body and finally un-map the state in the correct order to gain the 16 byte output cipher text.

B. AES Decryption

For the AES Decryption, the key schedule stays as similar, the only operations we have to develop are the inversed shift Rows and inverse mix Columns. As you can see, they are nearly identical to their encryption except that the rotation this time is to the right and that we use the inversed S-Box for the substitution. As for the operation of inversed mix Columns, the only difference is the multiplication matrix is different. Finally, the only thing left to do is to keep it all together in one inversed main algorithm. Please note that we use our expanded key backwards, starting with the last 16 bytes and then moving towards the start. The functional verification was carried out for all the test cases and hence the RTL modeling is taken to the synthesis process using the Xilinx tool.

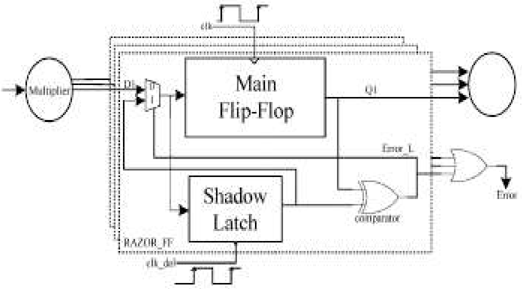

FIG 3. RAZOR FLIP-FLOPS

A Razor flip-flop contains flip-flop, XOR gate, mux and shadow latch. By utilizing a delayed clock signal, execution result is slower than the normal clock signal is catches by the shadow latch and the flip-flop catches the execution result for the combination circuit by utilizing a normal clock signal. The current operation path delay is exceeds the cycle period, and an incorrect result catches by the main flip-flop if the latched bit of the shadow latch is variant from that of the main flip-flop.

The error signal is set to 1 by the Razor flip-flop for notifying the system to re execute the operation if any errors occur and notify occurred an error in AHL circuit. For detecting whether an operation is examined to be a one-cycle pattern which can complete in a cycle by utilizing Razor flip-flops. If not the operation is re-executed with two cycles. While the execution is high cost, due to the re-execution frequency and overall cost is low.

IV. RESULTS

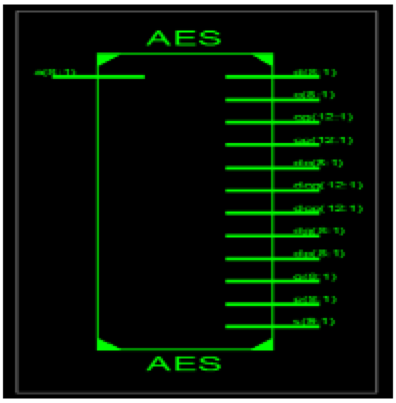

Fig 4. RTL SCHEMATIC

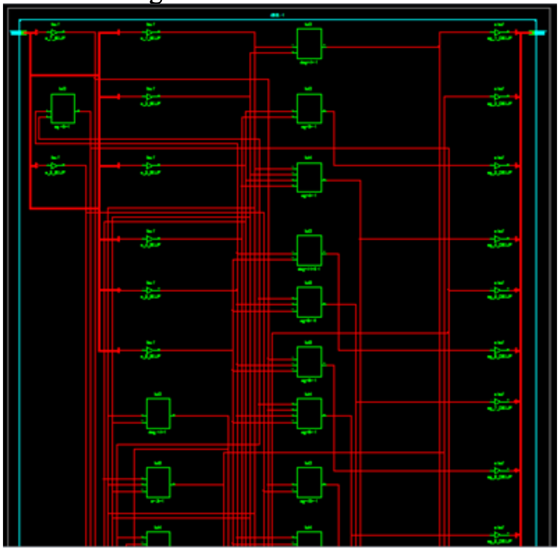

Fig 4. TECHNOLOGY SCHEMATIC

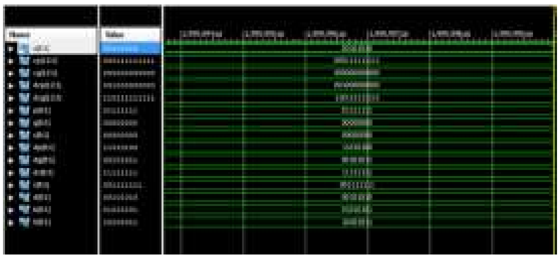

Fig 5. OUTPUT WAVEFORM

V. CONCLUSION

We have presented VLSI architecture for the Rijndael AES algorithm which executes both the encryption and decryption. S-boxes are used for the implementation of the multiplicative inverses and shared between encryption and decryption. The round keys are required for each round of the implementation which is generated in real-time. The forward and reverse key scheduling is developed and hence allowing area minimization which is efficient. Encryption algorithm is being utilized by military and government for a secure communication. The purpose of encryption is for hiding the data from unauthorized usage. We proposed a method for employing the crypto processor run in integration with a General Purpose Processor. We have proposed a AES algorithm which is a pipeline version that can encrypt data.

VI. REFERENCES

[1] S. P. Mohanty, K. R. Ramakrishnan, and M. S. Kankanhalli, “A DCT Domain Visible Watermarking Technique for Images,” in *Proc of the IEEE International Conf on Multimedia and Expo*, 2000, pp. 1029–1032.

[2] M. S. Kankanhalli and T. T. Guan, “Compressed-Domain Scrambler / Descrambler for Digital Video,” *IEEE Transactions on Consumer Electronics*, vol. 48, no. 2, pp. 356–365, May 2002.

[3] B. M. Macq and J. J. Quisquater, "Cryptography for Digital TV Broadcasting," *Proceedings of the IEEE*, vol. 83, no. 6, pp. 944–957, Jun 1995.

[4] H. Kuo and I. Verbauwhede, "Architectural Optimization for a 1.82 Gbits/sec VLSI Implementation of the AES Rijndael Algorithm," in *Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems*, 2001, vol. 2162, pp. 51–64.

[5] M. McLoone and J. V. McCanny, "Rijndael FPGA Implementation Utilizing Look-up Tables," in *Proceedings of the IEEE Workshop on Signal Processing Systems*, 2001, pp. 349–360.

[6] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A Compact Rijndael Hardware Architecture with S-Box Optimization," in *Proceedings of Advances in Cryptology - ASIACRYPT 2001*, 2001, pp. 171–184.

[7] S. Mangard, M. Aigner, and S. Dominikus, "A Highly Regular and Scalable AES Hardware Architecture," *IEEE Transactions on Computers*, vol. 52, no. 4, pp. 483–491, April 2003.