AN EFFICIENT ENCRYPTION IMPLEMENTATION USING AES ALGORITHM TECHNIQUES

[1]Dr Rajamohan Parthasarathy*, [2]Mr Seow Soon Loong, [3]Ms Preethy Ayyappan

[1]School of Information Technology, SEGi University
[2] School of Information Technology, SEGi University
[3] Faculty of Engineering and Built in Environment, SEGi University

* prajamohan@segi.edu.my

Abstract

The AES algorithm is a symmetric block cipher that can encrypt, (encipher), and decrypt, (decipher), information. Encryption converts data to an unintelligible form called cipher-text. Decryption of the cipher-text converts the data back into its original form, which is called plaintext. The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits. The National Institute of Standards and Technology, (NIST), solicited proposals for the Advanced Encryption Standard, (AES). The AES is a Federal Information Processing Standard, (FIPS), which is a cryptographic algorithm that is used to protect electronic data. Advanced Encryption Standard (AES), specifying an Advanced Encryption Algorithm to replace the Data Encryption standard (DES) the Expired in 1998. NIST has solicited candidate algorithms for inclusion in AES, resulting in fifteen official candidate algorithms of which Rijndael was chosen as the Advanced Encryption Standard. Some of these implementations are optimized for speed, some for area, some for configurability, and some for low-power applications. This is carried out in the Cadence Tool with NC simvision software.

Keywords: Advanced Encryption Standard (AES), Reversible logic, Exclusive-OR (XOR) operation, Data Encryption standard (DES).

1. Introduction:

To protect the data transmission over insecure channels two types of cryptographic systems are used: Symmetric and Asymmetric cryptosystems. Symmetric cryptosystems such as Data Encryption Standard (DES) [1], 3 DES, and Advanced Encryption Standard (AES) [4], uses an identical key for the sender and receiver; both to encrypt the message text and decrypt the cipher

text. Asymmetric cryptosystems such as Rivest-Shamir Adleman (RSA) & Elliptic Curve Cryptosystem (ECC) uses different keys for encryption and decryption. Symmetric cryptosystem is more suitable to encrypt large amount of data with high speed. To replace the old Data Encryption Standard, in Sept 12 of 1997, the National Institute of Standard Technology (NIST) required proposals to what was called Advanced Encryption Standard (AES).

Many algorithms were presented originally with researches from 12 different nations. Fifteen algorithms were selected to the Round one. Next five were chosen to the Round two. Five algorithms finalized by NIST are MARS, RC6, RIJNDAEL [2], SERPENT and TWOFISH [3]. On October 2nd 2000, NIST [4] has announced the Rijndael algorithm is the best in security, performance, efficiency, implement ability, & flexibility. The Rijndael algorithm was developed by Joan Daemen of Proton World International and Vincent Rijmen of Katholieke University at Leuven. AES encryption is an efficient scheme for both hardware and software implementation. As compare to software implementation, hardware implementation provides greater physical security and higher speed. Hardware implementation is useful in wireless security like military communication and mobile telephony where there is a greater emphasis on the speed of communication. Most of the work has been presented on hardware implementation of AES using FPGA [5]-[8]. This paper presents efficient hardware architecture design & implementation of AES using software tool and describes performance testing of Rijndael algorithm.

1.1 AES Algorithm

An encryption algorithm converts a plain text message into cipher text message which can be recovered only by authorized receiver using a decryption technique. The AES-Rijndael algorithm [4] is an iterative private key symmetric block cipher. The input and output for the AES algorithm each consist of sequences of 128 bits (block length). Hence Nb = Block length/32 = 4.The Cipher Key for the AES algorithm is a sequence of 128,. In this implementation we set the key length to 128. Hence Nk = Key length/32 = 4. 2.1

1.2 Encryption Process:

The Encryption and decryption process consists of a number of different transformations applied consecutively over the data block bits, in a fixed number of iterations, called rounds. The number of rounds depends on the length of the key used for the encryption process. For key length of 128 bits, the number of iteration required are 10. (Nr = 10). As shown in Fig. 1, each

of the first Nr-1 rounds consists of 4 transformations: SubBytes(), ShiftRows(), MixColumns() & AddRoundKey().[3]-[8]

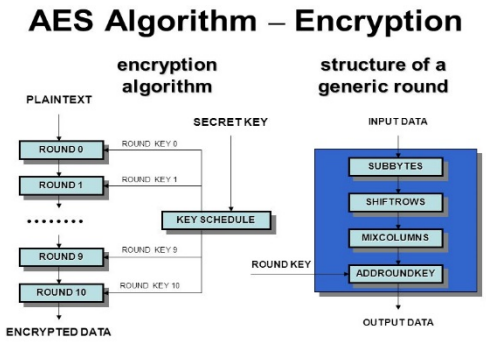

Figure: 1 – AES Algorithm Encryption

Table: 1 – AES Algorithm Encryption Generic Rounds

Number of rounds (Nr)	128-bit Data	192-bit Data	256-bit Data
128-bit Key	10	12	14
192-bit Key	12	12	14
256-bit Key	14	14	14

AES is taken by most them as research paper to find out the correct architecture to implement on the hardware. Speed and resource requirement are the mainly considered to select the architecture for the application. In this area and speed considered but other important parameter is delay. There are number of operation methods were there for FIPS-197. One of the methods is ECB. Additional flexibility to attack can be obtained by using one of the modes. Next Output Feed Back (OFB) mode appropriately such mode, also restricts the efficiency of pipelining. FPGA [6] role is improving from prototyping level to main stream production level. This drastic change is due to the high pressure to decrease design cost, less time to market and risk. Due to growth in technology, results in various version of FPGA by the leading manufacturer. In some of the application made the user to move from FPGA to ASIC (application specific integrated circuits), it gives low cost for user[3].

1.3 Inputs, outputs and the state

The plaintext input and cipher text output for the AES algorithms are blocks of 128 bits. The cipher key input is a sequence of 128, 192 or 256 bits. In other words the length of the cipher key, Nk, is 4, 6 or 8 words which represent the number of columns in the cipher key. The AES algorithm is categorized into three versions based on the cipher key length. The number of rounds of encryption for each AES version depends on the cipher key size.

In the AES algorithm, the number of rounds is represented by Nr, where Nr = 10 when Nk = 4, Nr = 12 when Nk = 6, and Nr = 14 when Nk = 8. The following table.

Table: 2 – AES Algorithm Blocks Inputs, Outputs and State

	Block Size in Words N_b	Key Length in Word N_k	Number of Rounds N_r
AES 128 Bits Key	4	4	10
AES 192 Bits Key	4	6	12
AES 256 Bits Key	4	8	14

The table gives the description about the different key sizes AES algorithm. As the key size increases the security for the data increases.

The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits. Reversible logic circuits have attracted the attention of researchers in recent years for mainly two reasons. Firstly, Landauer showed that during logic computation, every bit of information loss generates KTln2 joules of heat energy, where K is the Boltzmann's constant and T is the absolute temperature of environment. And, according to Bennet, for theoretically zero energy dissipation, computations have to be reversible in nature. Secondly, quantum computations which are the basis of quantum computers are reversible in nature[7].

The input and output for the AES algorithm consists of sequences of 128 bits. These sequences are referred to as blocks and the numbers of bits they contain are referred to as their length. The Cipher Key for the AES algorithm is a sequence of 128,192 or 256 bits. Other input, output and Cipher Key lengths are not permitted by this standard. The bits within such sequences are numbered starting at zero and ending at one less than the sequence length, which is also termed the block length or key length. The number "i" attached to a bit is known as its index and will be in one of the ranges 0 <= i < 128, 0 < i < 192 or 0 < = i < 256 key length specified.[5]

2. Encryption Process

AES is an iterative rather than Feistel cipher. It is based on 'substitution–permutation network'. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).

Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix.

Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key. The schematic of AES Cipher structure is given in the following illustration as shown in Figure: 2 and Figure: 3.

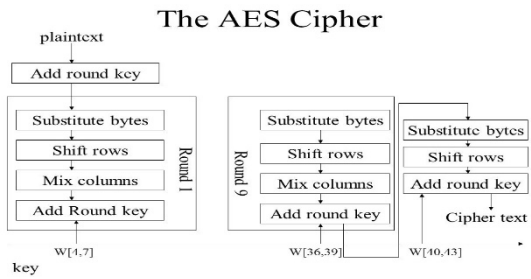

Figure: 2 – The Schematic AES Cipher Structure

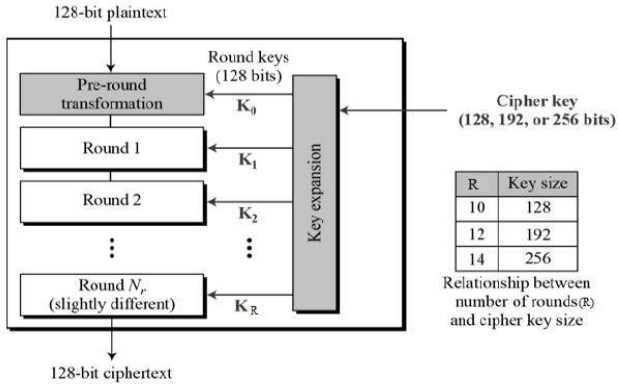

Figure: 3 – The AES Schematic AES Cipher Text Round Keys

2.1 Add Round Key

In the Addition of Round Key transformation Add Round Key, a Round Key is added to the State by a simple bitwise XOR operation each of the 16 bytes of the state is XORed against each of the 16 bytes of a portion of the expanded key for the current round. The Expanded Key bytes are never reused. So once the first 16 bytes are XORed against the first 16 bytes of the expanded key then the expanded key bytes 1-16 are never used again. The next time the Add Round Key function is called bytes. example: 11010100 XOR 11111111 = 00101011 (Hex 2B).

2.2 Substitute Bytes

The bytes substitution transformation Byte sub (state) is a non-linear substitution of bytes that operates independently on each byte of the State using a substitution table (Sbox).[4-8]

2.3 Shift Rows

This block functions Circular byte shift in each

1. 1st row is unchanged
2. 2nd row does 1 byte circular shift to left
3. 3rd row does 2 byte circular shift to left
4. 4th row does 3 byte circular shift to left.

2.4 Mix column

This is perhaps the hardest step to both understand and explain. There are two parts to this step. The first will explain which parts of the state are multiplied against which parts of the

matrix. The second will explain how this multiplication is implemented over what's called a Galois Field.

2.5 Matrix Multiplication

The state is arranged into a 4 row table. The multiplication is performed one column at a time (4 bytes). Each value in the column is eventually multiplied against every value of the matrix (16 total multiplications). The results of these multiplications are XORed together to produce only 4 result bytes for the next state. Therefore 4 bytes input, 16 multiplications 12 XORs and 4 bytes output. The multiplication is performed one matrix row at a time against each value of a state column.

3. Encryption Intermediate Results

3.1 Substitute Bytes simulation

The waveforms generated by the 128-bit byte substitution transformation. The inputs are clock of 10ns time period, Active High reset, and128-bit state as a standard logic vector, whose output is128-bit S-box lookup table substitution operation is complete 1 clock cycles as shown in Figure: 4

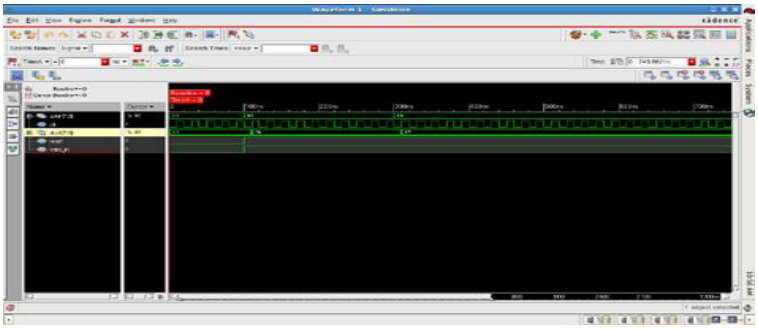

Figure: 4 – Substitute Bytes Simulation Waveform Transformation

3.2 Shift Rows simulation

The waveforms generated by the 128-bit shift row transformation. The inputs are clock of 10ns time period Active High reset, and 128-bit state as a standard logic vector. This operation is complete 1 clock cycles as shown in Figure: 5.

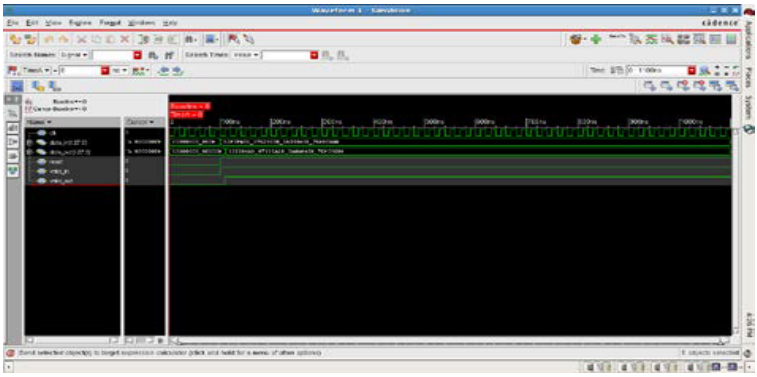

Figure: 5 – Shift Rows Simulation Waveform Transformation

3.3 Add Round Key simulation

The waveforms generated by the 128-bit Add Round Key operation. The inputs are clock of 10ns time period, 128-bit key and 128-bit state as a standard logic vector, whose output is the 128-bit which is EXOR operation of 128-bit state and 128-bit key. This operation is complete 10 clock cycles as shown in Figure: 6.

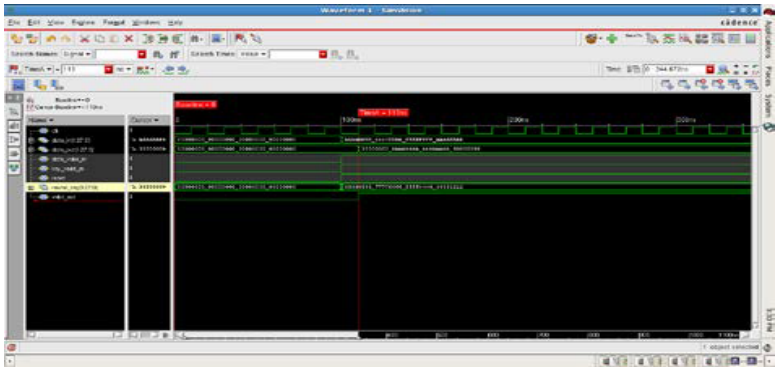

Figure: 6 – Add Round Key Simulation Waveform Transformation

3.4 Mix Columns

The waveforms generated by the 128-bit Mix Columns transformation. The inputs are clock of 10ns time period, Active High reset, and 128-bit state. This operation is complete 1 clock cycles as shown in Figure: 7.

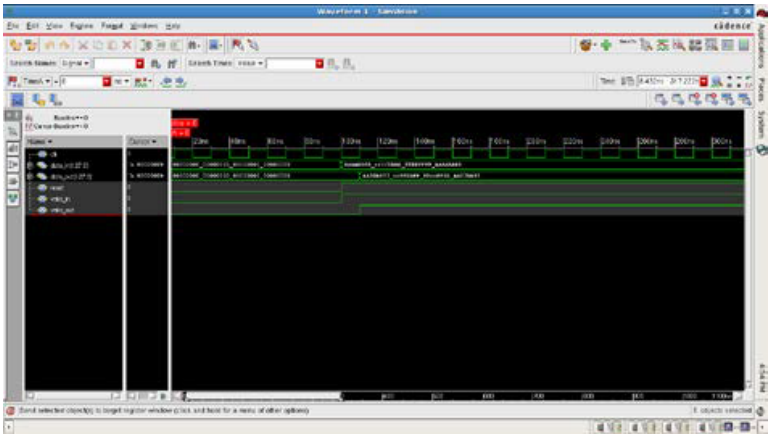

Figure: 7 – Mix Columns Simulation Waveform Transformation

4. Conclusion

The Rijndael algorithm was chosen as the new Advanced Encryption Standard (AES) for several reasons. The purpose was to create an algorithm that was resistant against known attacks, simple, and quick to code. In addition, the block size and key size can vary making the algorithm versatile. AES was originally designed for non-classified U.S. government information, but, due to its success, AES-256 is usable for top secret government information. As a result it can be stated that AES is not appropriate for being used in the strategic applications that have been classified.

5. Acknowledgement

The authors would like to thank SEGi University Management, SEGi Journal of Engineering & Technological Advances (JETA) Editorial Board, Research Innovation Management Centre (RIMC) SEGi University and School of Information Technology SEGi University.

References

[1] "Data Encryption Standard (DES)," National Technical Information Service VA 22161, 1999, National Institute of Standards and Technology (NIST)

[2] Maraghy M, Hesham S & Abd El Ghany M.A, “Real-time Efficient FPGA Implementation of AES Algorithm”, IEEE International SOC Conference Sept 2013.

[3] R. Drechsler, A. Finder, and R. Wille. Improving ESOP-based synthesis of reversible logic using evolutionary algorithms. In Proceedings of Intl. Conference on Applications of Evolutionary Computation (Part II),

[4] M.Sambasiva Reddy & Mr.Y.Amar Babu, “Evaluation Of Microblaze and Implementation Of AES Algorithm using Spartan-3E”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 2, July 2013.

[5] P. Kocher, J. Jaffe, and B. Jun. Differential power analysis, In Proceedings of Advances in Cryptology (CRYPTO ’99), LNCS Vol. 1666, pp: 388–397, 1999

[6] Hoang Trang and Nguyen Van Loi , “An Efficient FPGA Implementation of The Advanced Encryption Standard algorithm”,IEEE International Conference on Computing and Communication Technology, page 1 -4, Ho Chi Minh city, 2012.

[7] Kamali S.H, Shakerian R, Hedayati M and Rahmani M, “A new moodier version of Advanced Encryption Standard based algorithm for image encryption”, (ICEIE) International Conference on Electronics and Information Engineering, volume 1, Aug 2010.

[8] Ahmad N, Hasan R and Jubadi W.M, “Design of AES Sbox using combinational logic optimization”, IEEE Symposium on Industrial Electronics & Applications, Oct 2010.,