Design of High Security System Using MAES without using S Box Technique

Mr. Jen Abhilash[1], Parvathi T[2], Chandra Sravanthi[3] , Pushpa Karuna B[4], Vijay Lakshmi U[5]

Swarnandhra College of Engineering and Technology, Narsapur, Andhra Pradesh

Abstract—

The increasing requirement for securing data communication in computer networks has led to the development of numerous cryptography algorithms. The Advanced Encryption Standard (AES) is a computer security standard issued by the National Institute of Standards and Technology (NIST) intended for protecting electronic data. The Federal Information Processing Standards (FIPS) Publication 197 contains its specification. The AES cryptography algorithm can be used to encrypt/decrypt blocks of 128 bits and is capable of using cipher keys of 128 bits wide (AES128). This study proposes the hardware implementation of the AES128 encryption algorithm. A unique feature of the proposed pipelined design is that the round keys, which are consumed during different iterations of encryption, are generated in parallel with the encryption process. This lowers the delay associated with each round of encryption and reduces the overall encryption delay of a plaintext block. This leads to an increase in the message encryption throughput.

Key words

MAES encryption algorithm, data communication security, computer networks, cryptography algorithm.

1. INTRODUCTION

AES is short for Advanced Encryption Standard and is a United States encryption standard defined in Federal Information Processing Standard (FIPS) 192, published in November 2001. AES is the most recent of the four current algorithms approved for federal us in the United States. One should not compare AES with RSA, another standard algorithm, as RSA is a different category of algorithm. Bulk encryption of information itself is seldom performed with RSA. RSA is used to transfer other encryption keys for use by AES for example, and for digital signatures. AES is a symmetric encryption algorithm processing data in block of 128 bits. Under the influence of a key, a 128-bit block is encrypted by transforming it in a unique way into a new block of the same size. AES is symmetric since the same key is used for encryption and the reverse transformation, decryption. The only secret necessary to keep for security is the key. AES may configured to use different key-lengths, the standard defines 3 lengths and the resulting algorithms are named AES-128, AES-192 and AES-256 respectively to indicate the length in bits of the key. Each additional bit in the key effectively doubles the strength of the algorithm, when defined as the time necessary for an attacker to stage a brute force attack, i.e. an exhaustive search of all possible key combinations in order to find the right one.

AES may, as all algorithms, be used in different ways to perform encryption. Different methods are suitable for different situations. It is vital that the correct method is applied in the correct manner for each and every situation, or the result may well be insecure even if AES as such is secure. Encryption with AES is based on a secret key with 128, 192 or 256 bits. It is as critically vital to use good and strong keys as it is to apply AES properly. Creating one good and strong keys is a surprisingly difficult problem and requires careful design when done with a computer. The challenge is that computers are notoriously deterministic, but what is required of a good and strong key is the opposite – unpredictability and randomness. Keys derived into a fixed length suitable for the encryption algorithm from passwords or pass phrases typed by a human will seldom correspond to 128 bits much less 256. To even approach 128--bit equivalence in a pass phrase, at least 10 typical passwords of the kind frequently used in day- to-day work are needed. Weak keys can be somewhat strengthened by special techniques by adding computationally intensive steps which increase the amount of computation necessary to break it. Security is not an absolute;

it's a relation between time and cost [6].

2. ADVANCED ENCRYPTION STANDARD

Advanced Encryption Standard (AES) is a specification for the encryption of electronic data. The algorithm described by AES is a symmetric-key algorithm, meaning the same key is used for both encrypting and decrypting the data.

2.1 Description of the cipher:

AES is based on a design principle known as a substitution-permutation network. It is fast in both software and hardware. AES[6] has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits, whereas Rijndael[7] can be specified with block and key sizes in any multiple of 32 bits, with a minimum of 128 bits. The blocksize has a maximum of 256 bits, but the keysize has no theoretical maximum.

AES operates on a 4×4 column-major order matrix of bytes, termed the state (versions of Rijndael with a larger block size have additional columns in the state). Most AES calculations are done in a special finite field.

The AES cipher is specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of ciphertext. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform ciphertext back into the original plaintext using the same encryption key.

2.2 High-level description of the algorithm:

1. KeyExpansion—round keys are derived from the cipher key using Rijndael's key schedule

2. Initial Round

AddRoundKey—each byte of the state is combined with the round key using bitwise xor

3.Rounds

a) SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.

b) ShiftRows—a transposition step where each row of the state is shifted cyclically a certain number of steps.

c) MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.

d) AddRoundKey

4, Final Round (no MixColumns)

a) SubBytes,-In the SubBytes step, each byte in the matrix is updated using an 8-bit substitution box,the Rijndael S-box.

b) ShiftRows -The ShiftRows step operates on the rows of the state; it cyclically shifts the bytes in each row by a certain offset.

c) AddRoundKey

5. The MixColumns step:

In the MixColumns step, the four bytes of each column of the state are combined using an invertible linear transformation.

In the AddRoundKey step, the subkey is combined with the state. For each round, a subkey is derived from the main key using Rijndael's key schedule; each subkey is the same size as the state.

The subkey is added by combining each byte of the state with the corresponding byte of the subkey using bitwise XOR.

3. Area-Optimized AES-128

3.1 Brief Description of Rijndael Algorithm

Rijndael algorithm consists of encryption, decryption and key schedule algorithm. The main operations of the encryption algorithm among the three parts of Rijndael algorithm include: bytes substitution (SubBytes), the row shift (ShiftRows), column mixing (MixColumns), and the round key adding (AddRoundKey). It is shown as Fig. 1.

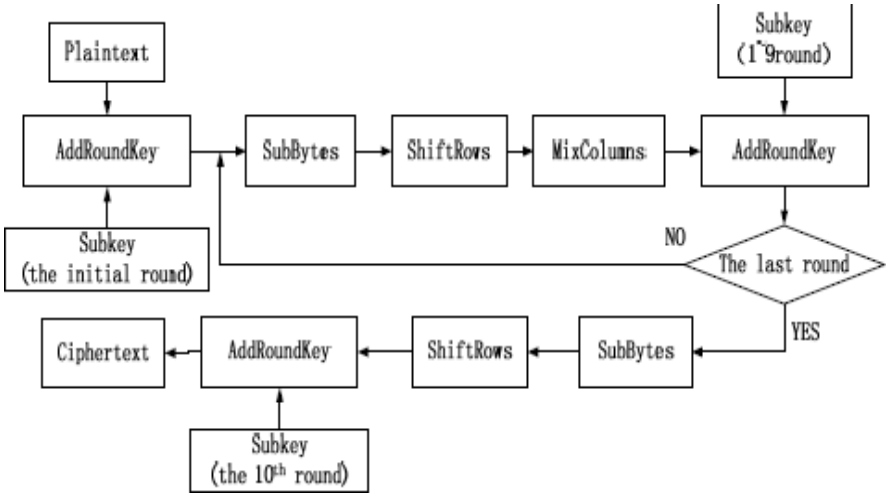

Fig. 1. The structure of Rijndael encryption algorithm

Encryption algorithm processes Nr+1 rounds of transformation of the plaintext for the ciphertext. The value of Nr in AES algorithm whose packet length is 128 bits should be 10, 12, or 14 respectively, corresponding to the key length of 128,192,256 bits. In this paper, only the (AES-128) encryption scheme with 128-bit keys is considered.

3.2 The Design of Improved AES-128 Encryption Algorithm

1) Two main processes of AES encryption algorithm:

The AES encryption algorithm can be divided into two parts, the key schedule and round transformation. Key schedule consists of two modules: key expansion and round key selection. Key expansion means mapping Nk bits initial key to the so-called expanded key, while the round key selection selectes Nb bits of round key from the expanded key module.

Round Transformation involves four modules by ByteSubstitution, ByteRotation, MixColumn and AddRoundKey.

2) Key points for the design

In the AES-128, the data in the main process mentioned above is mapped to a 4×4 two-dimensional matrix. The matrix is also called

a_{00}	a_{01}	a_{02}	a_{03}
a_{10}	a_{11}	a_{12}	a_{13}
a_{20}	a_{21}	a_{22}	a_{23}
a_{30}	a_{31}	a_{32}	a_{33}

state matrix, which is shown as Fig.2.

Fig. 2. The state matrix

In the four transformation modules of round transformation, the ByteRotation, MixColumn and AddRoundKey are all linear transformations except the ByteSub.

From analysis of the AES algorithm principle it is observed the following operations given below.

ByteSubstitution operation simply replaces the element of 128-bit input plaintext with the inverse element corresponding to the Galois field GF (28), whose smallest unit of operation is 8 bits/ group.

ByteRotation operation takes cyclic shift of the 128-bit state matrix, in which one row (32 bits) is taken as the smallest operand.

MixColumns operation takes multiplication and addition operations of the results of ByteRotation with the corresponding irreducible polynomial x8 + x4 + x3 + x + 1 in GF(28), whose minimum operating unit is 32 bits.

Addroundkey operation takes a simple XOR operation with 8-bit units.

The inputs of plaintext and initial key, intermediate inputs and outputs of round transformation, as well as the output of ciphertext in the AES algorithm are all stored in the state matrixes, which are processed in one byte or one word. Thus, in order to take operations at least bits, the original 128-bit data should be segmented. We design some external controllers in the new algorithm, so that the data transmission and processing can be implemented on each column of the state matrix

(32bit).That means the data should be packed and put into further operations.

Take the independent and reversible bytes substitution operation of S-box as example. Firstly, the state matrix is divided into four columns. And then byte replacement is achieved by the operation of look-up table shown as Fig. 3.

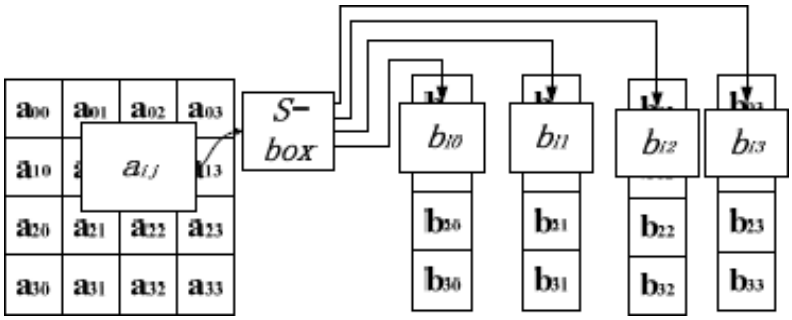

Fig. 3. Bytes segmentation and replacement processing

Therefore, the original 128-bit input of plaintext and key will be replaced with four consecutive 32-bit input sequences respectively. In order to decrease the output ports, four continuous 32-bit ciphertext sequences have taken place of the original 128-bit output by adding a clock controller. The 128- bit data in the round transformation is also split into four groups of 32-bit data before the operation of pipelining.

3) The Process of New algorithms

From the above analysis, it is found that the process of AES encryption can be mainly divided into two parts, key schedule and round transformation. The improved structure is also divided into these two major processes. The initial key will be sent to the two modules namely the Keyexpansion and Keyselection, while the plaintext is to be sent to the round transformation after the roundkey is selected. But the operand of data transmission is turned into a 32-bit unit. The new algorithm process is shown in Fig. 4.

The functions of various parts of the structure shown above are described as follow:

The initial round of encryption:

The four packets of consecutive 32-bit plaintext (128 bits) have been put into the corresponding registers. Meanwhile, another four packets of consecutive 32-bit initial key (128 bits) have been put into other registers by the control of the enable clock signal. Furthermore, this module should combine the plaintext and initial key by using the XOR operators.

Round Transformation in the intermediate steps:

A round transformation mainly realizes the function of SubBytes and MixColumns with 32-bit columns. Four packets of round transformation are processed independently. Then the results of MixColumns and the 32-bit keys sourced from Keyexpansion are combined by using XOR operators. Here, the round transformation is a module with 64 input ports (32- bit plaintext+32-bit key) and 32 output ports.

The function of SubByte is realized by Look-Up Table (LUT). It means that the operation is completed by the Find and Replace after all replacement units are stored in a memory (256×8bit = 1024 bit).

The implementation of MixColumn is mainly based on the mathematical analysis in the Galois field GF(28). Only the multiplication module and the 32-bit XOR module of each processing unit(one column) are needed to design, because the elements of the multiplication and addition in Galois field are commutative and associative. Then the function of MixColumn can be achieved. Fig.5 is a block diagram for the introduction of pipelining technology used in the round transformation.

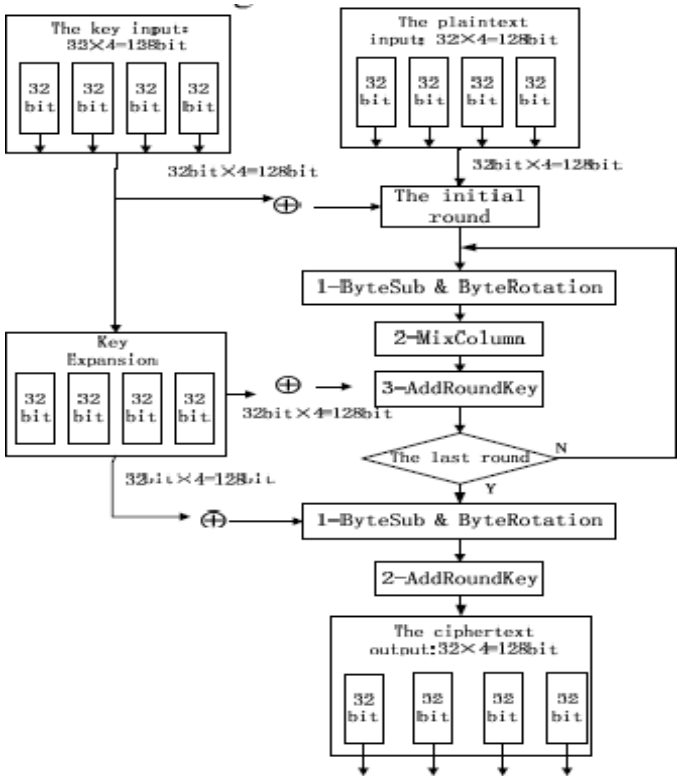

Figure 4. The new improved structure of AES algorithm

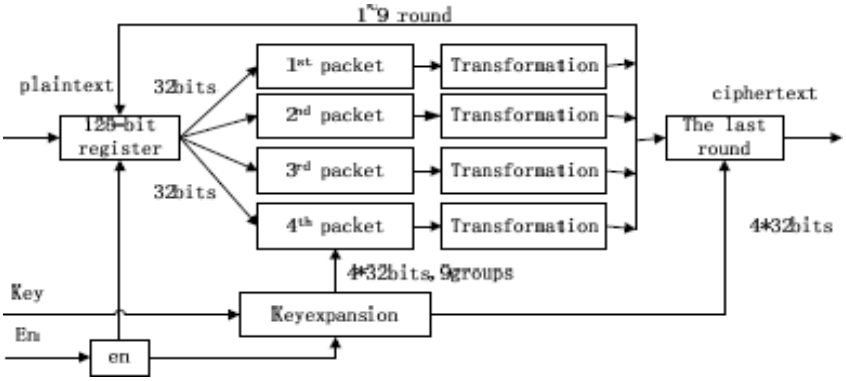

Figure 5. The round processing with pipeline technology

In the process of pipelining, the 128-bit data is divided into four consecutive 32-bit packets that take round transformation independently.

The operation of the above four groups of data can be realized in pipelining technology[4]. In brief, it can be described as follow: store the unprocessed data in the 128-bit register, and
control the clock for re-starting the 128-bit register to read the new data when the four groups' operations have been overcome. Thus the 128-bit round-operating unit has been transformed into four 32-bit round-operating elements. The internal pipelining processing should be implemented during the whole nine intermediate Round Transformations of the four packets before achieving the 128-bit ciphertext.

- The process of the last round
  The final round is a 128-bit processor. After nine rounds of operations included Shiftrows, SubByte and Mixcolumns, the 128-bit intermediate encrypted data will be used in XOR
  operation with the final expanded key(4*32bit), which is provided by the key expansion module. The output of final round in the processor is the desired 128-bit ciphertext. Similarly, the ciphertext is divided into four packets of 32- bit data by an external enable signal.

- Key expansion and Key extraction
  This module is implemented basically the same with the traditional way as another part of the AES encryption algorithm. The only difference lies on the mode of data transmission. The initial key and expanded keys are divided into four 32-bit data before being extracted.

All of the above modules can be decomposed into basic operations of seeking and XOR if the AES algorithm is implemented on FPGA[1,2]. So the basic processing unit (look-uptable) of FPGA can be used. The operation of AddRoundKey is taken first in each round. When the plaintext and initial key are input, the encryption module starts running, and the expanded keys are stored into the registers at the same time. This implementation method is independent on a specific FPGA.

4. VLSI Systems

Very-large-scale integration (VLSI) is the process of creating integrated circuits by combining thousands of transistor-based circuits into a single chip. The ICs have three key advantages over digital circuits built from discrete components with respect to size, speed, reduced cost and low power consumption. An Application-Specific Integrated Circuit (ASIC) is an integrated circuit (IC) customized for a particular use, rather than intended for general-purpose use.

Field-programmable gate arrays (FPGA) are the modern-day technology for building a breadboard or prototype from standard parts; programmable logic blocks and programmable interconnects allow the same FPGA to be used in many different applications. For smaller designs and/or lower production volumes, FPGAs may be more cost effective than an ASIC design even in production.

5. Results & Discussion

Figure 6 and 7 depicts the basic and detailed RTL schematic whereas figure 8,9 and 10 shows the simulation results of AES rounds and key expansion.

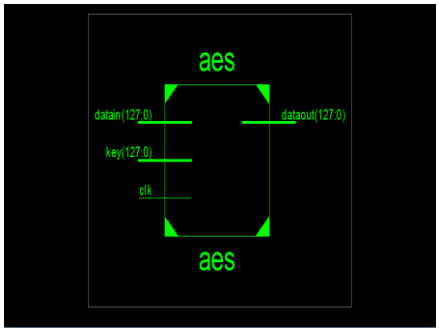

Fig.6. Basic RTL Schematic

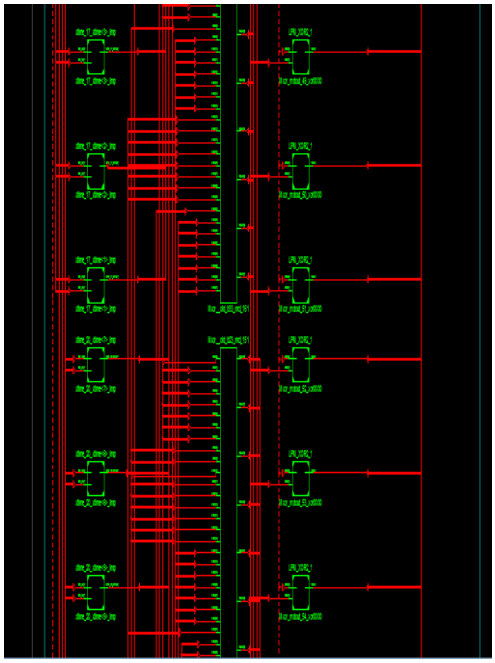

Fig. 7 Expanded RTL Schematic

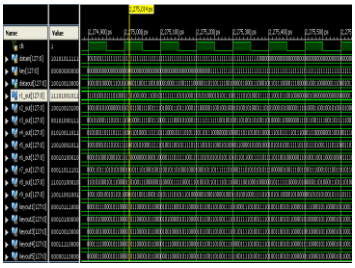

Fig.8 AES Simulation

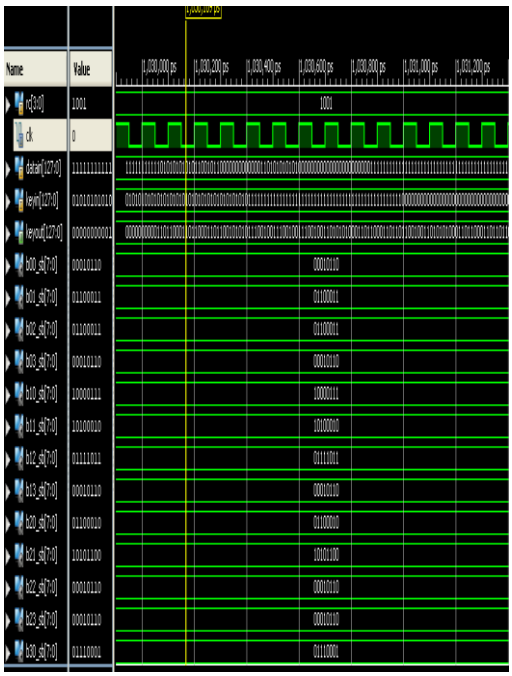

Fig.9 ROUNDS Simulation

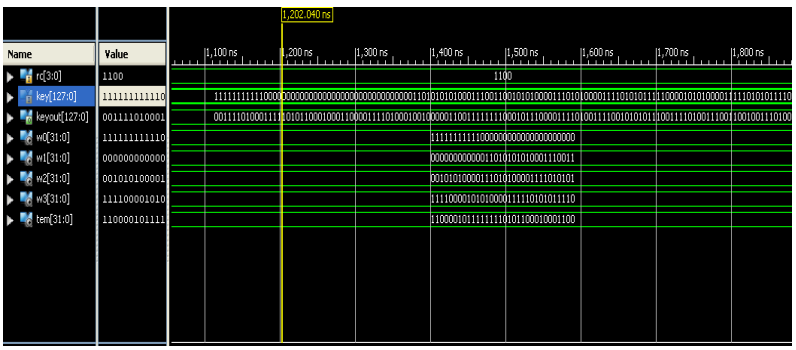

Fig. 10 Simulation for Key Expansion

CONCLUSION & FUTURE SCOPE

A FPGA implementation of area-optimized AES algorithm meeting the actual application is proposed here. After being coded with Verilog Hardware Description Language, the waveform simulation of the new algorithm was taken in the ModelSim SE PLUS 6.0 and Quartus 7.2 platform. Ultimately, a synthesis simulation of the new algorithm has been done.

The result shows that the design with the pipelining technology and special data transmission mode can optimize the chip area effectively. Meanwhile, this design reduces power consumption to some extent, for the power consumption is directly related to the chip area. Therefore the encryption device implemented in this method can meet some practical applications. As the S-box is implemented by look-up-table in this design,

the chip area and power can still be optimized. So the future work should focus on the implementation mode of S-box. Mathematics in Galois field (28) can accomplish the bytes substitution of the AES algorithm, which could be another idea of further research.

REFERENCES

[1] *Y. Jun, D. Jun, L. Na and G. Yixiong, "FPGA-Based Design and Implementation of Reduced AES Algorithm," 2010 International Conference on Challenges in Environmental Science and Computer Engineering, Wuhan, China, 2010, pp. 67-70, doi: 10.1109/CESCE.2010.123*

[2] *M. Deshpande, M. S. Deshpande and D. N. Kayatanavar, "FPGA implementation of AES encryption and decryption," 2009 International Conference on Control, Automation, Communication and Energy Conservation, Perundurai, India, 2009, pp. 1-6.*

[3] *S. Hiremath and M. S. Suma, "Advanced Encryption Standard Implemented on FPGA," 2009 Second International Conference on Computer and Electrical Engineering, Dubai, United Arab Emirates, 2009, pp. 656-660, doi: 10.1109/ICCEE.2009.231.*

[4] *Abdel-hafeez.S.,Sawalmeh.A. and Bataineh.S.,"High Performance AES Design using Pipelining Structure over GF(28)" IEEE Inter Conf.Signal Proc and Com.,vol.24-27, pp.716-719,Nov. 2007.*

[5] *M. R. M. Rizk and M. Morsy, "Optimized Area and Optimized Speed Hardware Implementations of AES on FPGA," 2007 2nd International Design and Test Workshop, Cairo, Egypt, 2007, pp. 207-217, doi: 10.1109/IDT.2007.4437462.*

[6] *M. Liberatori, F. Otero, J. C. Bonadero and J. Castineira, "AES-128 Cipher. High Speed, Low Cost FPGA Implementation," 2007 3rd Southern Conference on Programmable Logic, Mar del Plata, Argentina, 2007, pp. 195-198, doi: 10.1109/SPL.2007.371748.*

[7] *M. B. Abdelhalim, H. K. Aslan and H. Farouk, "A design for an FPGA-based implementation of Rijndael cipher," 2005 International Conference on Information and Communication Technology, Cairo, 2005, pp. 897-912, doi: 10.1109/ITICT.2005.1609675.*