S-Box Design Analysis and Parameter Variation in AES Algorithm

Rashi Kohli
Dept of Computer Science
Amity University, Uttar Pradesh

Divya Sharma
Dept of Computer Science
Amity University, Uttar Pradesh

Manoj Kr. Baliyan
Dept of Computer Science,
Amity University, Uttar Pradesh

ABSTRACT

In this paper fundamental scrutiny of AES algorithm with the non serviceable aspects i.e. elevated performance, high throughput, and area efficiency is offered. A virtual analysis of DES, 3DES and AES is shown. This paper will present the theoretical analysis of parameter variations in the generation of the S-BOX. The Rijndael cipher, premeditated by Joan Daemen and Vincent Rijmen, is particular as authorized Advance Encryption standard (AES) and it is well apt for hardware exercise.

General Terms

Security, Parameter Analysis of cryptographic algorithms, Cryptography

Keywords

AES, DES, 3DES, Cryptography, S-Box, Multi-encryption

1. INTRODUCTION

To maintain data secure, secret and copyright protected from assorted hackers and unauthorised admittance and users, numerous techniques have been built up such as cryptography, steganography. Cryptography is indispensable mainstay in the world of networking; conventionally use in armed and surveillance purposes. Cryptography provides the necessary mechanism to provide confidentiality, accountability and accuracy in network communication and other related fields.

Advanced Encryption Standard (AES) was issued as Federal Information Processing Standards by national institute (FIPS) by National Institute of Standards and technology (NIST).This paper uses 128 bit key generated by key scheduling algorithm. Analysis of Rijndael Algorithm passes through the 4 layers consists of

i. ByteSub Transformation (S-Box Creation),
ii. ShiftRow Transformation,
iii. MixColumn Transformation,
iv. AddRound Key

This paper focus on detailed survey about the non linearity of the S-Box which is an important component of AES, which uses process of affine mapping and Inv-affine mapping for encryption and decryption respectively for elevated performance, high throughput and area efficiency. AES architecture presented uses Polynomial multiplication using XOR transformations as a substitute of multipliers to diminish the hardware complexity. In the proposed architecture both encryption and decryption rounds are performed on the same hardware resources, correspondingly building design area efficient.

A look up table is calculated using Verilog hardware description language. An appropriate use of look up table in S-Box creation can be very effective in analysis of AES algorithm. S-Box is very important component layer in the scrutiny leading to security of cipher as it depends on the non-linearity factor. [3] [6] [9] since the symmetric cipher drives faster than asymmetric cipher so they are dominant in the same field. S-box is the only element that injects the nonlinearity in the cipher making it more powerful. The following table shows the no of keys according to the block size. [8][2].

Table1: AES Categorization

KeySize (words/bytes/bits)	4/16/128	6/24/192	8/32/256
Number of Rounds	10	12	14
Expanded key size (words/byte)	44/176	52/208	60/240

2. RIJNDAEL ALGORITHM

Rijndael proposed AES algorithm which is an iterative block cipher that chains a variable data block and a variable key length of 128,192, or 256 bits, this means that AES works by recurring the similar definite steps multiple times. One of the design criteria for AES algorithm is that it can be implemented in hardware and software both which was not likely to be there in case of DES i.e. DES can only be implemented in hardware .AES works on the bytes i.e. that impart simplicity towards implementation. The challenge is that computers are enormously deterministic, but what is requisite of a good and strong key i.e. unpredictability and uncertainty. An initial key is expanded to generate the round keys, each of size equal to block length. Each round of algorithm receives a new round key from the key scheduling algorithm. At the end of each round the intermediate cipher result is called state in the AES proposal. Each round engross an addition or bitwise EXOR of the plaintext plus the key, so the original key must be expanded into a number of Round Keys and this algorithmic approach is called as key scheduling algorithm. The key used is expanded into sub keys, for each operation round, this process of sub keys generation is called as key expansion unit.

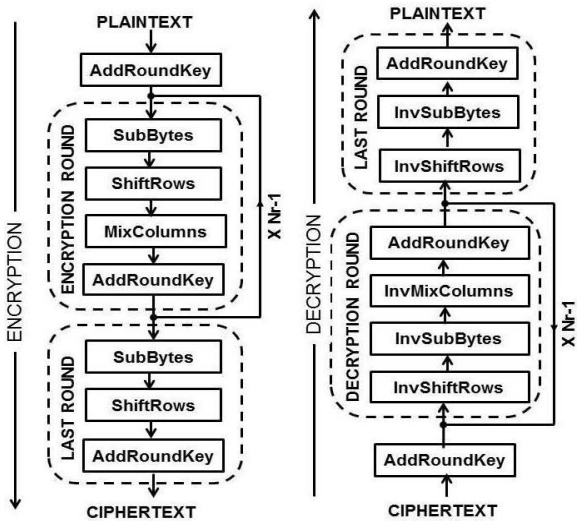

Figure 1: S-Box Components [2] [9]

3. S-BOX CALCULATIONS

S-Box is one of the most crucial keystones that will lead to the security at AES level as it is non linear, invertible transformation. S-box entries are computes using the multiplicative inverses in Galois Field GF (2^{8}). This multiplicative inverse uses the affine mapping concept for encryption part and inverse affine mapping concept for decryption part. S-Box is computed either by computing substitution or by looking the look up tables, in the analysis of AES. Using look up table adds non functionality of being fast and inexpensive in terms of power consumption but there is major drawback i.e. the size of the silicon is about 1,700 gates equivalents per table is 0.18 µ technology.

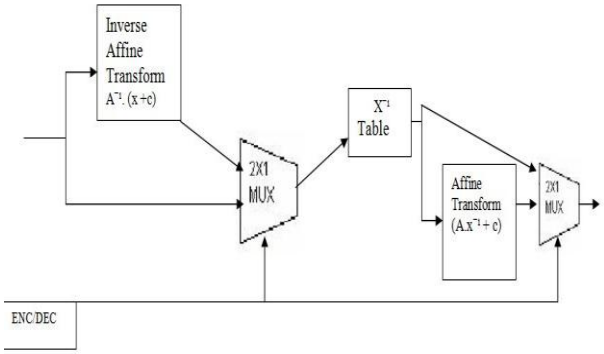

Figure 2: S-Box Computation

Affine mapping for encryption round will be calculated as valAffTrans[i] = val[i] ^ val[(i+4)%8] ^ val[(i+5)%8] ^ val[(i+6)%8]^ val[(i+7)%8] ^ c_i[i]; where c_i[i] is 01100011 is constant, leftmost bit is being MSB

Inverse Affine mapping for decryption round will be calculated as InvAffIndex[i] = index[(i+2)%8] ^ index[(i+5)%8] ^ index[(i+7)%8] ^ c_inv[i]; where c_inv[i] is 00000101 is constant, leftmost bit being MSB. [8]

3.1 Shift row layer

In this transformation the rows of the block state are shifted above diverse offsets. The sum of shifts is gritty by the block length. The Shift Rows transformation cyclically shifts the last three rows of the state by different offsets. The first row is left unaffected in this transformation. Each byte of the second row is shifted one position to the left. The third and fourth rows are shifted left by two and three positions, correspondingly. [7] [8] [9].

3.2 Mix column transformation layer

In this transformation each column of the block state is considered as a polynomial over *GF* (2^{8}). It is multiplied with a constant polynomial $C(x)$ or $D(x)$ over a Finite field in encryption or decryption, respectively. In hardware, the multiplication by the corresponding polynomial is done by XOR operations and multiplication of a block by X. This is implemented using a multiplexer; the control being the MSB is 1 or 0.

3.3 Add round key layer

In this transformation, the round key obtained from the key scheduler is XORed with the block state obtained from the Mix column transformation or shift row transformation based on the type of round being implemented, in the standard round, the round key is XORed with the output obtained from the Mix Column transformation. In the final round the round key is XORed with the output obtained from the Shift transformation. Add round key layer depends on the key scheduling algorithm. [8] [9].

4. KEY GENERATION ALGORITHM

1. The initial key is expanded and the generated round keys are stored in four 32-bit registers.
2. Both the forward and reverse key scheduling is done in the same device.
3. The *Byte Sub* required in the key expansion unit is implemented using the S-Boxes.
4. Four S-Boxes are needed for a 128-bit key and 128-bit data block implemented using 8 × 256 ROM cells.
5. Multiplexers are used as a control signal to distinguish between the initial key and the round key (obtained from the initial key using a key expansion unit).
6. The least significant 32 bits of the 128-bit key is cyclically shifted to the left by a byte, implemented using combinational logic.
7. The resulting word after the left shift operation is sent through the S-boxes and the affine mapping operation in order to perform Byte Sub
8. The key resulting from the *Byte Sub* is XORed with the Round Constant (RCON).
9. In this architecture, the round constant is generated using the combinational logic. The round constant should be symmetric with the round key being generated [5] [7] [8] [9]

5. COMPARATIVE ANALYSIS

Whether to choose DES, 3DES, AES depends on user needs and task. DES was developed in 1977 and it was carefully designed to work better in hardware than software. DES performs lots of bit manipulation in substitution and permutation. Advance Encryption Standard (AES) was

developed to take into account safety measures, software & hardware recital, suitability in restricted-space milieu and resistance to power testing.[4]

Table 2: Comparison DES, 3DES &AES [4]

Distinguishing Parameters	DES (Data Encryption Standard)	3DES (Triple Data Encryption Standard)	AES(Advance Encryption Standard)
Key Size	56 bits	112 or 168 bits (depending on keys used)	128,192 or 256 bits
Time of algorithm	Symmetric	Symmetric	Symmetric
Speed	Low	Moderate	High
Resource consumption	High	Moderate	Low
Security	Proven Inadequate	Still Insecure	Secure
Timeline	1977	Standardized 1978	Official standard Since 2001
Block Size	64 bits	64bits	128,192 or 256 bits
Time to Crack(assume to check all possible keys at 50 billion keys per second)	For 64 bit key 400 days	For 112 bit key 800 days	For 128 bit key 5×10^{21} years

6. PROPOSED PARAMETER VARIATIONS

Parameter variations are categorized on the basis of Die-to-Die (D2D) and With-In-Die (WID) variations. D2D applies directly on the functional blocks or on a software chip as a whole, affecting the internal connections which use the multipliers and XOR multiplications used in creation of S-Box to impart the non-linearity. Inter chip or Inter block variations are caused by systematic effects affecting the internal functions, typically the affine and Inv-affine mapping used. Conversely a WID variation consists of random and systematic components that induce different electrical characteristics. WID variations can occur because of some intrinsic factors such as statistical deviations of clock frequency during encryption and decryption affecting the throughput. In D2D device-to-device correlation factor is 1 and in case of WID correlation factor is 0. [1]

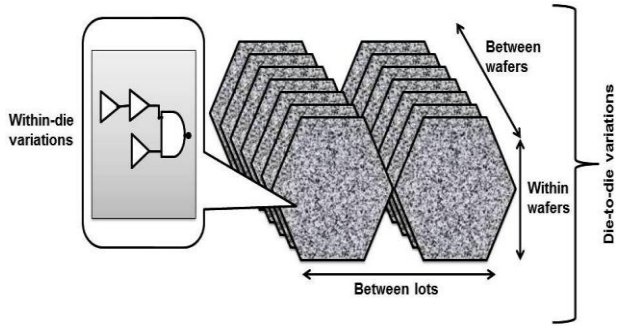

Figure 3: Parameter Variation [1]

The main problem of arithmetic implementations is the increased power dissipation. To solve this problem of power dissipation we need a mechanism for S-BOX creation that uses the Galileo's field .This means that power dissipation and cost must be the design constraints which need to be focused to provide effective low cost solution for the same.

To achieve this we can implement Conventional S-BOX by using re-ordering of components, introducing avalanche effect mechanism, by reducing the driving strength as fan out, fan in, and insertion of registers occupying the maximum area. The first parameter for achieving low power dissipation can be implemented if inputs are supplied continuously using look up table for the S-box and all the transitions in all the inputs of all the phases of encryption and decryption occur at the same time although it increases the complexity level but simultaneously affects the power dissipation. This is an ideal theoretical scenario in which power is consumed solely for the calculation of multiplexers and XOR transformation used in the S-BOX and this is referred as "Zero-delay" model. [1]

7. FURTHER ANALYSIS

This part of analysis states that the analysis of Rijndael algorithm can be mapped into the prototype chip using the four layers described in this paper, but it should take into an account.

7.1 Implementation analysis:

The design units that are described must be evident with respect to their functionality. As described earlier S-Box injects non-linearity which is the important part as per security is concern. So while designing the S-box, careful strategies must be employed as it uses the pipelined architecture.

7.2 Memory optimization:

Seeing as the devise is based on one clock cycle for apiece encryption round, the reminiscence modules had to be duplicated .For example, in the Byte Sub, the S-boxes need to be duplicated 16 times. Consequently, the alternative of memory architecture is very decisive. Since the entire table entries are fixed in the standard, the usage of ROM is

preferred. The architecture requires several small ROM modules instead of one large module, since each lookup will only be based on a maximum of 8-bit address, which translates to 256 entries. We employed the multiplicative inverse function using the look-up table of size 8 × 256. We have a total of 20 copies of the S-boxes in our design; 16 of them in encryption module and 4 in the key scheduling module. [5] [8] [9].

8. CONCLUSION AND FUTURE SCOPE

In this paper firstly the in depth survey of S-Box analysis has been done which imparts non-linearity to the system in accordance with that the parameter variations which use die-to die concept and within-in-die concept is used over here. Secondly the comparative analysis of DES, 3DES, AES is shown specifically in terms of speed, resource consumption& security. This paper describes how to minimize the power dissipation and how to map the S-box into prototype chip. High security is major concerned today ,security applications are rising at its peak, In every nook and corner whether it is network or it can be implemented in smart cards as well. Here S-box report non linearity to the system making it more secure. To increase the security to protect the data & other related entities many techniques have been proposed by the researchers using their empirical knowledge. This paper highlights the secure components of S-box which is using XOR operations instead of polynomial multiplication still there is a limitation of complex look up tables that were used in the creation of S-box, we can use other algorithms as IDEA, BLOWFISH, RSA, RC6, XTEA & other light weight algorithms.

Yet another possibility to increase security is multi-encryption technique using the above algorithms since the evolution of encryption of encryption in the field of cryptography may provide better security, and increase in throughput than single encryption techniques.

9. REFERENCES

[1] Yusuf Leblebici, Milos Stanisavljevic Saif-ur RehmanVariability "Tolerant Design and Optimization of an Advanced Encryption Standard Core in State-of-the-Art 65nm Technology" August 20,2010

[2] Liberatori monica, otero Fernando, Bonadero J.C, Castineira, " AES_128 cipher, high speed, low cost FPGA implementation" , [1-4244-0606-4/07]

[3] AL-Qudah Ibrahim mohammad, DOA, "New S-box design of advance Encryption standard (AES)" Computer Science, Ministry of High Education, Ryiahd Saudi Arabia.

[4] Hamdan.O.Alanazi, B.B.Zaidan, A.A.Zaidan, Hamid A.Jalab, M.Shabbir and Y. Al-Nabhani, "New Comparative Study between DES, 3DES and AES "[ISSN 2151-9617].

[5] The Design of Rijndael, Springer-Verlag, 2002.

[6] A. J. Elbirt, W. Yip, B. Chetwynd, and Christof Paar, ÒAn FPGA Implementation and Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists,Ó in Proceedings of the Third Advanced Encryption Standard (AES) Candidate Conference, 2000, pp. 13Ð27.

[7] Panu Hämäläinen,Timo Alho,Marko Hännikäinen, and Timo D. Hämäläinen Tampere "Design and Implementation of Low-area and Low-power " University of Technology / Institute of Digital and Computer Systems P. O. Box 553, FI-33101, Tampere, Finland

[8] M. Kosaraju, varanasi murali, P.Mohanty saraju , "A high performance VLSI Architecture for Advance Encryption Standard (AES) algorithm", 1063-9667/06 (IEEE)

[9] George N. Selimis∗, Athanasios P. Kakarountas, Apostolos P. Fournaris, Athanasios Milidonis, and Odysseas Koufopavlou, Department of Electrical and Computer Engineering, University of Patras, Greece, 26110"A Low Power Design for Sbox Cryptographic Primitive of Advanced Encryption Standard for Mobile End-Users" Journal of Low Power Electronics Vol.3, 1–10, 2007