Sindh Univ. Res. Jour. (Sci. Ser.) Vol.45 (4):683-688 (2013)

SINDH UNIVERSITY RESEARCH JOURNAL (SCIENCE SERIES)

Simulation Based Modified Solution of Aes and Encryption Key Establishment

S. M. A. QADRI*, H. NAWAZ++, M. Y. KOONDHAR **, M.Y. HAB***

Department of Computer Science, Sindh Madressatul Islam University, Karachi

Abstract: AES (Advanced encryption standard) is the most widely used encryption standard around the world. Data encrypted by AES is strong enough to decrypt by any other means. Although the AES encryption is one of the best encryption till now. The main issue of AES raised in this paper are 1) Key development and securing issue. 2) Universally known algorithm. To overcome these issues of AES there is a need of some modification in standard AES, for this the modification described in this paper not only provide the efficient solution of these main issues but it also resolves the issue in such a generic way that it can be applicable to all different size and different modes of AES algorithm. One more important thing is all the modification implemented in AES algorithm till now is a part of that modified algorithm which leaves the gate open for any cryptanalysis to extract the source code of modified algorithm and all the modification steps are known by that cryptanalysis, the modification on standard AES implemented in this paper is through the use of unique technique in which the modified part of the paper is processed outside the actual AES algorithm , the dongle key concept is introduced in this paper which is responsible for the generation the secondary key, The hardware implementation of dongle key not only protects the modified AES algorithm but it also insures the authentication of the user.

Keywords: *AES, KEY, Encryption, Modification*

1. INTRODUCTION

The AES (Advanced Encryption Standard) was introduced by Joan Daemen & Vincent Rjimen. The United States government used AES as official Encryption standard. The NIST (National Institute of Standards and Technology) U.S. FIPS PUB 197 proclaimed the AES (FIPS, 2001). As the use of internet is increasing day by day it demands strong encryption technique for securing information (Hua *et al* 2009; Bilski *et al* 2011) AES has many features which provide strong level of security and flexible architecture with high computing, Normally AES block cipher used large no of bits to offer very high security (Zhou *et al.*, 2009; Pham *et al* 2012). Advance encryption standard is strong Encryption tool till now but it still required Modifications because the cryptanalysis is designing more and more efficient techniques to attacks on AES every day (Kamali *et al* 2010) No of different techniques are developed to acquire the information from the encrypted data in which differential power analysis (Chen *et al* 2010) and Brute force attack are power full attack used to recover the encrypted data, as according to Moor's law the processing speed of chips will double every eighteen months, this increase in processing power providing sufficient strength to brute force attack and the recovery of secret key process is becoming efficient day by day. In this paper "Simulation based Modified solution of AES and Encryption Key Establishment" the security improvement of AES against the brute force attack is discussed and method of improving AES weaknesses is defined in such a generic way that it can be applicable on every key size and every modes of AES. To insure the correctness of the Modification introduced in this paper the AES software has been designed also by using C # and all the modification of this paper is also implemented on this software, the result from this software show that the output of modified software is much more different than ordinary AES output which insure that it is impossible for the ordinary AES to decrypt the data Encrypt by Modified software and it is also impossible for any brute force attacker to recover the keys as the modification are applied between the round of the AES.

2. BACK GROUND

AES is using symmetric cipher which processes 128-bit blocks of data. AES supports 128 bits key size, 192 bits key size and 256 bits key size and having 10, 12 & 14 iterations, (Gurkaynak, 2006). There are two operation, key expansion and round transformation or conversion process performed by the AES encryption. The process key expansion expand 128 bit user key 10 times 128 bit round key that 128 bit sited on 4x4 matrix. User key 1 each one round mix the data by help of Round key, that round key generated by the encryption key. The states that is called 4x4 matrix of bytes which maintained by the cipher. By help of those states the operation is performed. Primarily, state is packed with the input data block and encryption key

++Corresponding Author: Haque Nawaz Lashari, E-mail: hnlashari@smiu.edu.pk, Cell# +92-333-7588408
*Department Networks & Telecommunication, MAJU, Karachi
** Information Technology Centre, Sindh Agriculture University, Tandojam,
***IMCS Sindh University Jamshoro

than XOR. The matrix last column the first row exchanged by last row of column then that column substituted from sbox value than XOR with first column of user key and round constant column. Regular rounds consist of procedures known as sub bytesshift rows, mix columns & add round key (Gurkaynak, 2006)

Sub Bytes: The Sub Bytes transformation is a *n*on linear substitution operation that works on bytes. each byte of the input state is replaced using the same substitution function (called S-Box) (Ahmad *et al* 2010; Rachh *et al* 2009). The S-Box is defined as the multiplicative inverse (Naziri *et al* 2008) in the Galois field GF (2^8) with the irreducible polynomial m(x) = x 8 + x 4 + x 3 + x + 1 followed by an affine transformation. The Inv Sub Bytes transformation, which is needed for decryption, is the inverse of the affine transformation followed by the same inversion as in the Sub Bytes transformation.

(2) Shift Rows: The Shift Rows transformation rotates each row of the input state to the left, whereby the offset of the rotation corresponds to the row number. The Inv Shift Rows of this transformation is computed by performing the corresponding rotations to the right (FIPS, 2001).

(3) Mix Columns: The Mix Columns transfers map of each column of input state to a new column in output state. Every one input column considered as a polynomial above GF (2^8) & that multiplied with constant polynomial a(x) = (03) x3 + (01) x2 + (01) x + (02). Coefficient of a(x) is elements of GF (2^8) also & in this equation represented by hexadecimal values. The Inverse Mix Columns is the conversion of multiplication of each column with a(x) = (0B) x3 + (0D) x2 + (09) x + (0E) (Mohamed , 2010).

(4) Add Round Key: Add Round Key conversion is self inverting. That maps a 128 bit input state to 128 bit output state by XOR the input state by a 128-bit round key (Thongkhome, 2011 ; Lee, 2010).

3. MATERIALS AND METHODS

These three major issues of AES are addressed in this paper and the solution of these issues are provided in such a generic way that it is applicable for all Key size and all types of mode use in AES. The solutions are provided in two different steps.

Step No 1 Key Establishment Step

The key is considered as the primary key some compression algorithm or XOR operation is applied on the bytes of that key which compress the 16 bytes key in 1 byte. The value of this byte is considered as the substituted value of AES SBOX an inversely extract the value of this substituted byte from AES SBOX.

Now this value is referred to the inverse SBOX and 16 byte key is extracted from there. This key is called the Secondary Key which is the XOR with primary key to extract Encryption Key.

Step No 2 Modification of AES

In this step the three steps of AES Sub Bytes, Shift Row and Mix Colum are compressed in to two steps Shift Row and combined Sub Bytes and Mix Columns step (Hossain, 2010) . These two most computational steps Sub Bytes and Mix Column are combined using customized lookup table, so instead of using two most computational required steps the proposed algorithm uses only one step with considerable less computational requirement which provides the efficient implementation of algorithm Further more there is also step variation done on proposed algorithm in which instead of doing the Shift Row step the combined Sub Byte and Mix Column step is done first than Shift Row step is executed these variation change the data in such a way that it becomes impossible by any other AES software to decrypt the proposed modified algorithm data. The Modification is shown in F**igure 1.**

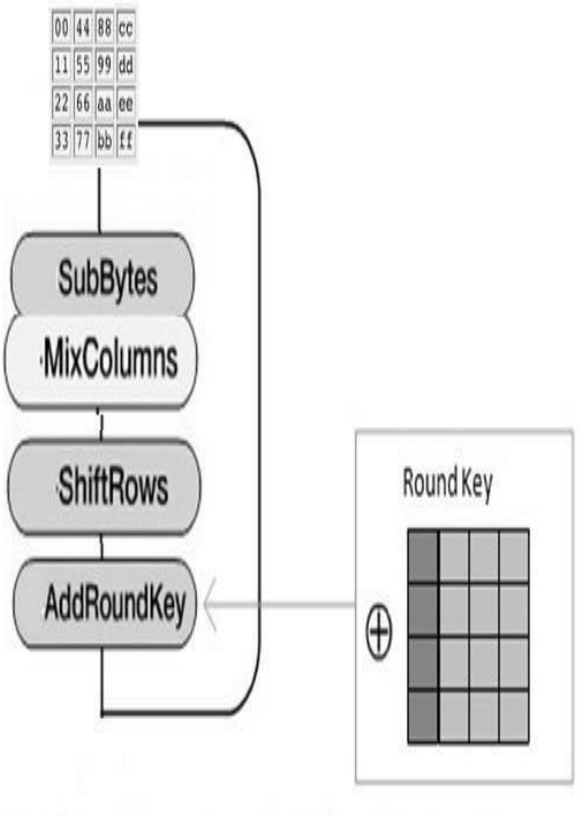


Figure 1 Modified AES

4. RESULTS AND DISCUSSION

4.1 C # as Simulation Tool

The C# programming language is a language which is known as an intermediate language that is used by programmers to create executable programs. C# is most powerful but complex C++ (called "see plus plus") by means of the ease of use. It is user friendly but extra verbose than Visual Basic. The Visual Basic .NET manifestation is roughly similar with C# in many respects. C# is under the umbrella of .NET language, C# has introduced many new features. The .cs is the extension of C# (C Sharp) program

4.2 A Modern Object-Oriented Language

C# language endeavors to balance the requirement for the consistency and the efficiency. The object-oriented language like Smalltalk, according to point of view that "Everything is an object." The instance type integer is an object according to the approach it is the advantage. On the other hand, the disadvantage is inefficiency. To evade that overhead, additional languages like Java pick to separate the type system interested in primitives and every thing else, most important to less overhead, however, to split between user-defined & primitive types. C# balancing the 2 conflicting point of views through providing a unified system type by which all types as well as primitive type is derived from common base type, although analogously consenting for performance optimization that permit primitive type & user defined type are treated as raw memory, through nominal overhead and better efficiency.

4.3 Develop Robust & Durable Software

Primarily, C# is a language type safe, connotation that programs are prevented from accessing the objects in unsuitable customs. All codes and data are linked by a type, all objects have a linked type, and although operations are defined with the linked type that can be performed on an object. Type safety remove complete type of errors in C and C++ programs which stemming from invalid casts, like bad pointer arithmetic & even malicious code F**igure 1-9.**.

4.4 Simulator used for Modified AES

The C Sharpe program is used to simulate the modified AES encryption because C Sharp is the most robust and durable programming tool available now a days, it also provides better security option as compared to any other language tool. The form developed for the simulation of modified AES has fixed Key sized of 128 bits, four textbox and 6 buttons which are named as:

Text boxes

a) Key b). Plain Text c) Cipher Text d) Deciphered Text

Buttons

a) Encipher b). Decipher c). Super Encipher d). Super Decipher e). Spr File Encipher f) Spr File Decipher

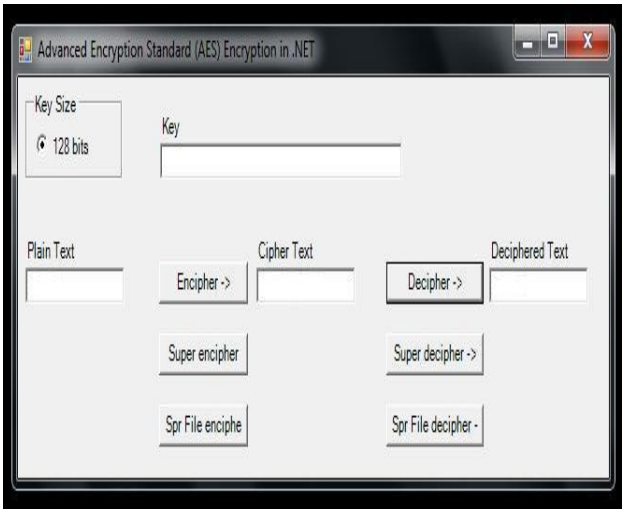


Figure 2 Modified AES simulator

4.4 Simulation Text Boxes

Key Text Box:	this text box is used to enter the user key during the process of encryption and decryption.
Plain Text Box:	This text box is used to enter the user text which is needed to be encrypted.
Cipher Text Box:	When user text is entered in plain text box and Encipher button is pressed, the encipher text of that user text appears on Cipher Text Box.
Decipher Text Box:	when the encipher text is placed on encipher text box and decipher button is pressed, the decipher text appears on Decipher Text Box

1.5 Simulation Buttons

Encipher:	This button is used to convert the user plain text into Encipher text.

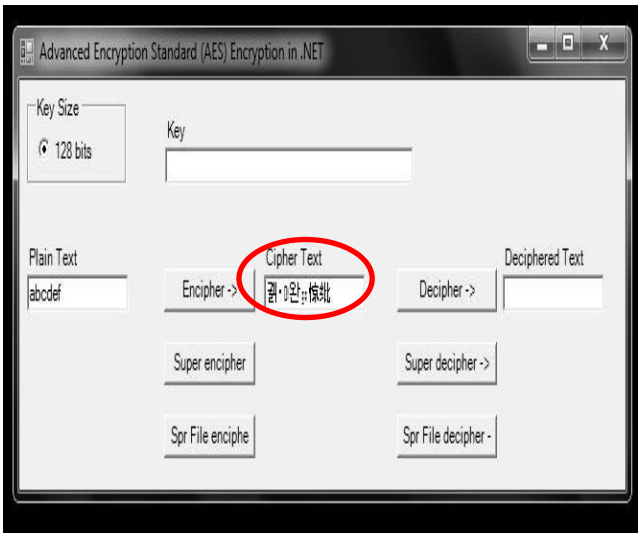


Figure 3 Standard AES Encryption

Decipher: This button is used to convert the Cipher text into plain text

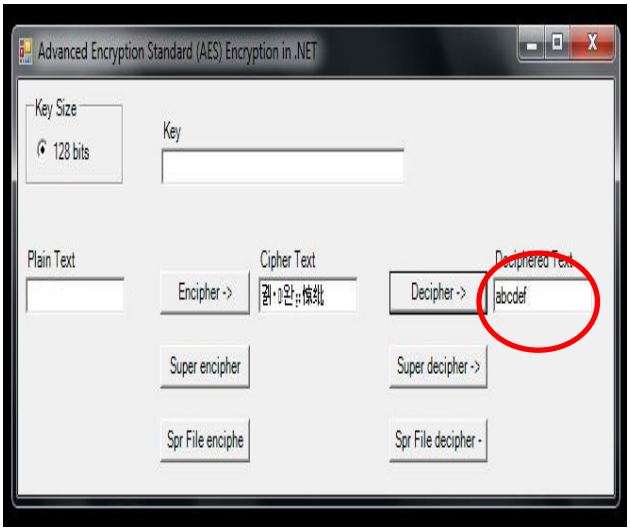


Figure 4 Standard AES Decryption

Super Encipher: This button is used to convert the plain text into cipher text using the modified AES algorithm.

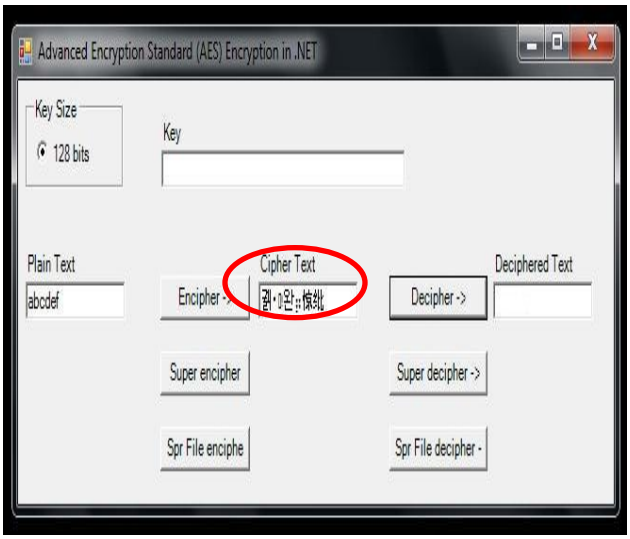


Figure 5 Modified AES Encryption

Super Decipher: This button is used to convert the encipher text into plain text

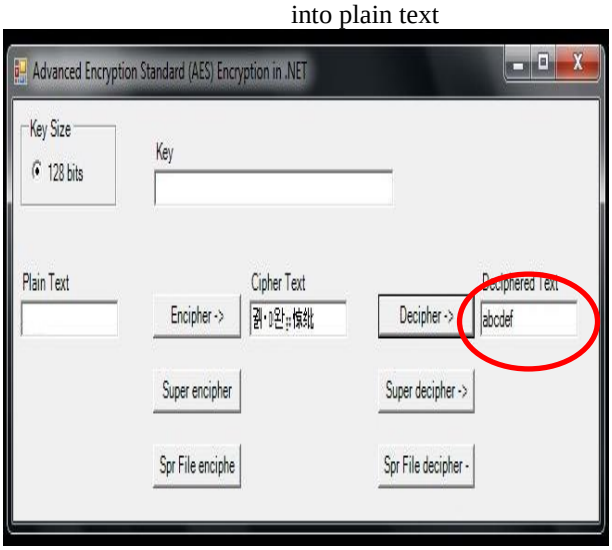


Figure 6 Modified AES Decryption

Super File Encipher: This button is used to encipher the jpeg image into encipher image using modified AES algorithm. Note although the encipher text is in image format yet it is not viewable through any image viewer software.

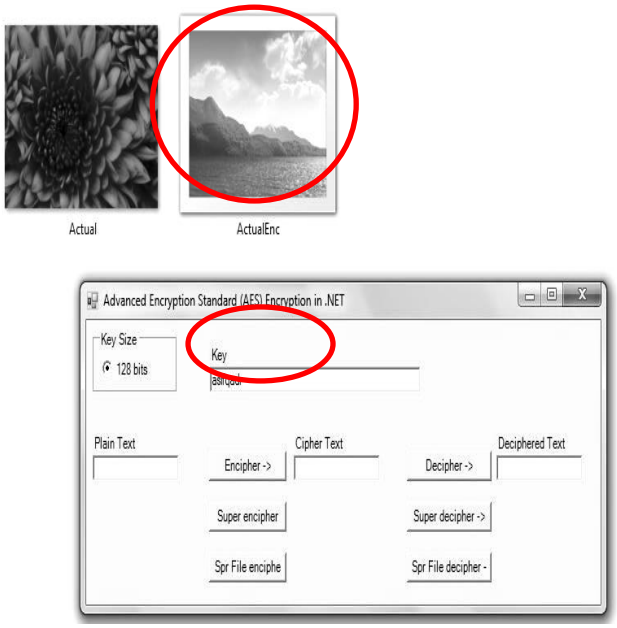


Figure 7 Super File Encipher using Key

Super File decipher: This button is used to decipher the jpeg image using modified AES algorithm.

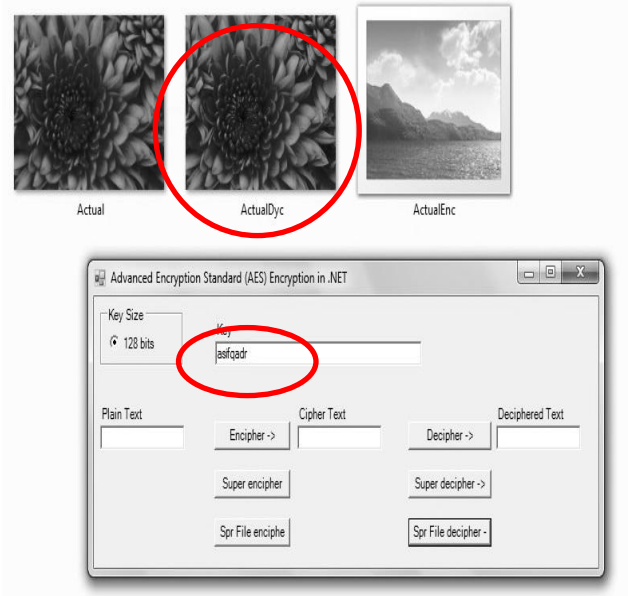


Figure 8 Super File Decipher using Key

To decipher the encipher image there should be a correct user key required, through which the actual image is enciphered. If the wrong key is used to decipher the encipher image, the result is in the form of error

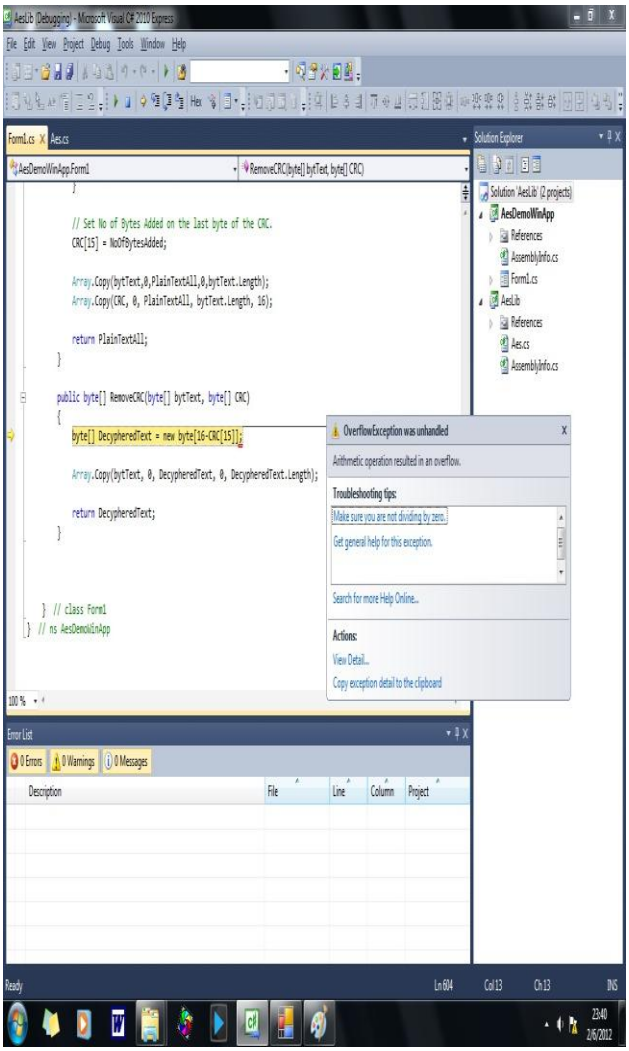


Figure 9 Error return when using wrong Key

5. CONCLUSION

The described modification and key establishment method of AES provide the customized way to produce encrypted data which can only be decrypted through the specific software and hardware. No other AES decryption software can decrypt the data although it has both data and key as well.

1. A unique method of Establishing encryption key provides more secure encrypted data.
2. The modification of AES algorithm provides the customized way to produce the encrypted data which can only be decrypted through the specific software and hardware no other AES encryption software can decrypt the data although it has both data and key.

3. Implementation of AES algorithm using customized lookup table provides less computational intensive algorithm.

6. FUTURE WORK

The modification provided in this paper is strong enough but still there is a need of such modification through which not only the key establishment procedure is secure but it also provides sufficient protection to modified AES algorithm. To protect modified algorithm from cryptanalysis the process of secondary key establishment should take place on some other platform outside the modified algorithm and the resultant secondary key is returned to the modified algorithm for further processing. For this there is a need for such a hard ware which not only processes modified secondary key generation but also provides the authentication of the user. The best solution for this is to make the dongle key which resolves not only all these issues but also ensure the authentication of the user.

REFERENCES:

Ahmad, N. and R. Hasan, (2010) "*Design of AES S-Box using combinational logic Optimization*", Faculty of Electrical and Electronic EngineeringbUniversiti Tun Hussein OnnMalaysia (UTHM)Johor, Malaysia, IEEE Symposium on Industrial Electronics and Applications (ISIEA), Penang, Malaysia, 696 – 699.

Bilski, P., W. Winiecki and T. Adamski, (2011) "*Implementation of symmetric cryptography in embedded systems for secure measurement systems",* Instrumentation and Measurement Technology Conference (I2MTC), IEEE, Binijang, 1 – 6

Chen, C., X. Li, L. Wu and X. Zhang, (2010) "*Design and Implementation of a Differential Power Analysis System for Cryptographic Devices*" Tsinghua National Laboratory for Information Science and Technology Institute of Microelectronics of Tsinghua University, Beijing 100084, China IEEE , 1-3

FIPS, (2001) *"Announcing the advanced encryption standard AES",* Federal Information Processing Standards Publication 197, United States, 1- 47

Gurkaynak, F. K. (2006) *"GALS System Design: Side Channel Attack Secure Cryptographic Accelerators"* Chapter 3, Cryptographic Accelerators, Hartung-Grove, ISSN 0936-5362, Volume 168, 21-48.

Hossain, F. S., L. Ali and N. Roy, (2010) *"Design and analysis of a high performance AES processor",* University of Engineering and Technology Dhaka , 6th International Conference on Electrical and Computer Engineering ICECE, Dhaka, Bangladesh, 562 – 565.

Hua, D. C., J. Jun, W. X. Ming and X. W. Yuan, (2009) *"Fast S-Box Substitution Instructions and Their Hardware Implementation for Accelerating Symmetric Cryptographic Processing",* International Conference on E-Business and Information System Security, Wuhan, 1 - 4.

Kamali, S.H. and Maysam Hedayati, (2010) "*A new modified version of advanced encryption standard based algorithm for image encryption",* International Conference on Electronics and Information Engineering (ICEIE) Islamic Azad University Qazvin Branch, Iran. Volume 1, 141-145.

Lee, H., K. Lee and Y. Shin, *(*2010) *"Implementation and Performance Analysis of AES-128 CBC algorithm in WSNs",* Department of Computing, Soongsil University, Korea, ISBN 978-89-5519-146-2, ICACT, 243-248.

Mohamed, A. and A. H. Madian, (2010) "*A Modified Rijndael Algorithm and it's Implementation using FPGA",* IEEE, Electrical and Electronics Engineering Department German University in Cairo, New Cairo City, Egypt, ICECS, 335-338.

Naziri, S. Z. M. and N. Idris, (2008) *"The Memory-less Method of Generating Multiplicative Inverse Values for S-box in AES Algorithm",* School of Microelectronic Engineering Universiti Malaysia Perlis, 2008 International Conference on Electronic Design Penang, Malaysia, IEEE, 1-5 .

Pham, T. A., M.S. Hasan and H. Yu, (2012) "*Area and power optimization for AES encryption module implementation on FPGA*", 18th International Conference on, Automation and Computing (ICAC), 1–6.

Rachh, R.R. and B.S. Anami, (2009) *"Efficient Implementations of S-Box and Inverse S Box for AES algorithm",* IEEE, KLE college of Engineering and Technology, Belgaum, India, 1- 6.

Thongkhome, K. and C. Thanavijitpun, (2011) " *A FPGA Design of AES Core Architecture for Portable Hard Disk",* Eighth International Joint Conference IEEE on Computer Science and Software Engineering (JCSSE), King Mongkut's Institute of Technology Ladkrabang Bangkok 10520, Thailand , 223-228.

Zhou, B., Y. Peng, K. Gaj and Z. Zhou, (2009) *"Implementation and comparative analysis of AES as a stream cipher"*, 2nd IEEE International Conference on Computer Science and Information Technology, Beijing, 396 – 400.