A Hybrid Architecture to overcome the Error Propagation Effect of AES

B. Sarkar
Dr. B. C. Roy Engineering College
Durgapur, West Bengal, India.

C. T. Bhunia
National Institute of Technology,
Yupia, Arunachal Pradesh, India.

U. Maulik
Jadavpur University
Kolkata, West Bengal, India.

ABSTRACT

In the recent past, AES (Advanced Encryption Standard) has been developed to replace DES (Data Encryption Standard) due to several reports of failure [1, 2] of security or key of DES. The replacement has aimed to augment the level of security mainly with the higher key size. Besides the higher level of security, AES has aimed to provide higher efficiency and better flexibility by means of encryption at different levels and with different block sizes [3]. AES, however, suffers from a major limitation owing to error propagation in the encryption process, which is undoubtedly a great research challenge. The AES encryption is done at several rounds of iteration. Each round of iteration has different input data and different key. The input data and the keys of different rounds are all generated from the original source data and the source key respectively. On the basis of this theory the input data and the keys at rounds follow a data path and a key path respectively. Any bit error at any round, if occurs either at the data path or at the key path, the effect propagates and results in remarkably large number of errors. The research [4, 5] reported this limitation of AES in their authoritative work. In literature, several studies have been made on this issue and several techniques are suggested to tackle the effect. In this paper, we have made extensive studies on Error Propagation Effect of AES algorithm (data path) and reviewed the solutions provided through an efficient hybrid method so that the error propagation effect of AES can be eradicated. Certainly, there are some assumptions and considerations that are stated in appropriate points of the discussion.

General terms

Information security, Cryptography.

Keywords

AES, encryption, decryption, bit error, error propagation, Longitudinal Redundancy Check, majority rule, Selective encryption.

1. INTRODUCTION

The limitation of error propagation in AES leads to lower speed of encryption, more processing overhead and higher complexity because until and unless error free encryption be achieved, the transmission of the cipher will be meaningless. In order to tackle the error propagation of AES, two techniques, namely the redundancy based technique and the byte based parity technique were studied in literatures [3, 4]. The redundancy based technique needs two modules: encryption module and decryption module for producing error-free cipher at the transmitter end. The output cipher of the encryption module is decrypted by the decryption module. The decrypted output is that which is compared (comparison of 128 pair of bits in case of the 128-bit plain text) with the plain text to check whether there is any error at all. If they match, the cipher is taken for error-free and it is then duly transmitted to serve the purpose as required. The dual process of encryption and decryption by the technique make the encryption process slow and costly as well. The byte based parity technique as studied in [4] makes use of parity checking at each byte of plain text to combat error. The byte based parity technique is mainly suitable for hardware implementation [6]. On the contrary, the redundancy based technique is applicable to both hardware and software based implementations. The redundancy based technique is such a method as firmly guarantees the error correction in the case of all the error vectors that may generate in the AES encryption process. In this paper we have proposed a novel hybrid scheme as has introduced the modified version of the redundancy based technique [7] as one module and a probabilistic approach through majority rule as the other one (in parallel) in order to firmly assure the abolition of the effect of the error propagation.

2. PROPOSED SCHEME

The proposed scheme suggests the theory of the Longitudinal Redundancy Check (LRC) code and Majority rule towards the prevention of the error propagation effect of AES.

LRC [8] is a form of redundancy check that is applied independently to each of a parallel group of bit streams. The important benefits of LRC are that it reduces the bandwidth and I/Os required for repair reads over prior codes, while still allowing a significant reduction in storage overhead [9].

Majority rule processes do not require consensus for group action. Instead, decisions are made by voting with a majority determining the position of the entire group. It studies the problem of aggregating individual judgments into collective decisions. Its basic framework is that of the application of majority rules to choose between two options: acceptance or rejection of a given proposition. This approach has the advantage of being able to produce a prompt and clear decision [10].

First the LRC code, say L_1, is generated from the input state (plain text) P and then P is encrypted (by the key, say K) through the Encryptor–1 to generate the cipher text C which is then again decrypted through the decryptor to find P_1. Again another LRC code, say L_2, is generated from P_1. L_1 and L_2 are then compared in the comparator. If the comparison proves L_1 and L_2 to be same, it means, there is no error injected /

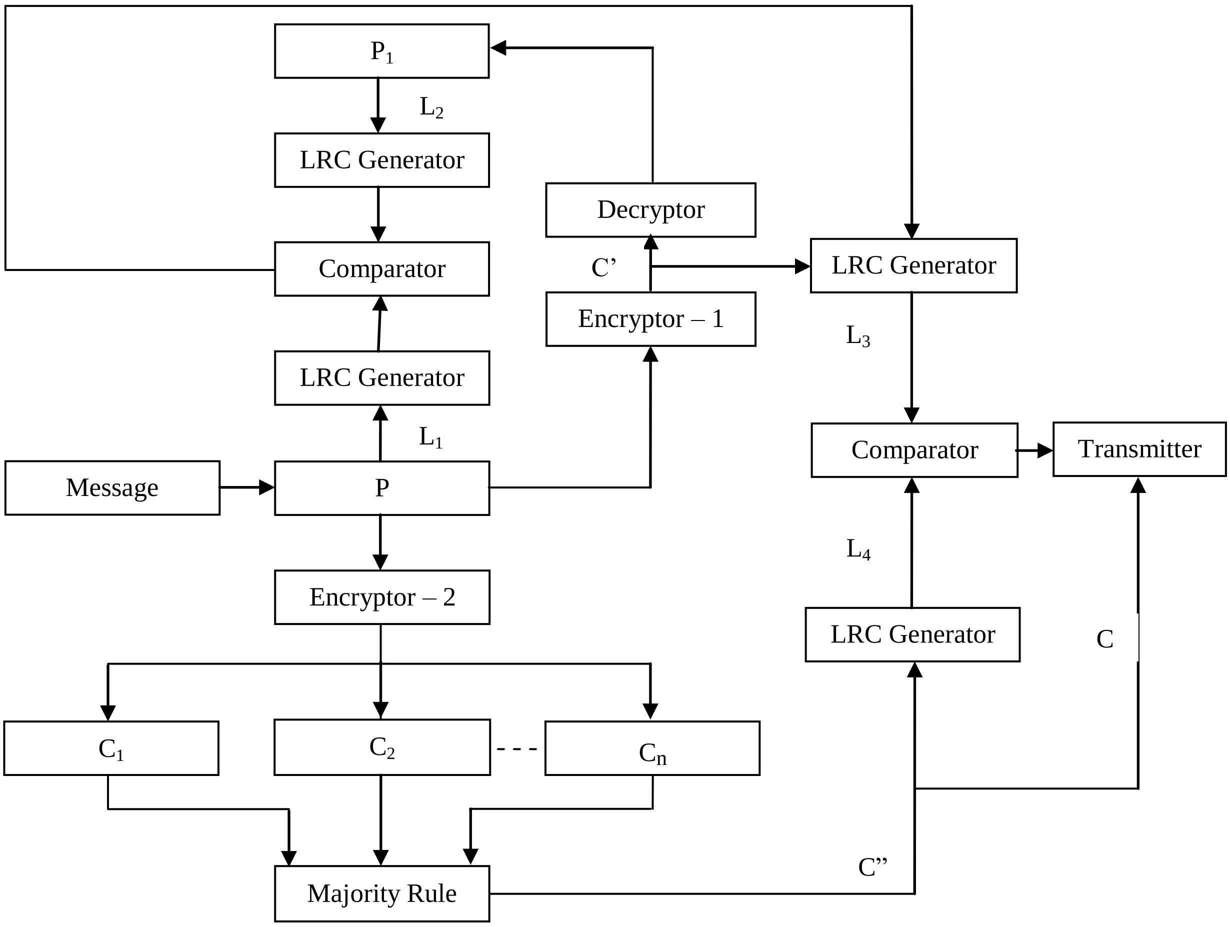


Fig 1: Block diagram of the proposed scheme

generated in the intermediate states of the encryption process, assuming that the decryption process is entirely error-free. Hence the most likely error free cipher, say C', is fed to another LRC code generator to find, say L_3 which is then fed to another comparator. It should be clearly noted that the introduction of Longitudinal Redundancy Check code leads to the comparisons
of only 8 pairs of bits instead of 128 pairs of bits.

Again, the plain text is encrypted (through the Encryptor–2) odd number of times, say n, with the same key K. As a consequence, n number of cipher text will be generated. Now it is assumed that the probability of occurrence of a single bit error amidst the rounds must not reach 0.5 so that out of n cipher texts at most (n – 1)/2 number of cipher texts may be erroneous whereas the least number of error free cipher texts is (n + 1)/2 out of n. After getting n cipher texts, Majority Rule is applied over them and as a result the most likely error free cipher text is achieved. It is assumed that the number of error free cipher texts is higher than that of the erroneous ciphers and hence the Majority Rule applied will always culminate in the generation of the most likely error free cipher, say C", which is also fed to another LRC code generator to find, say L_4 which is fed to the second comparator.

In the second comparator, it is checked that whether L_3 and L_4 are equal. If L_3 and L_4 are found equal, C' and C" are also considered to be same (say C) and if so, it can be concluded that the cipher C is entirely free from errors and can be transmitted over the channel (Fig 1).

2.1 Proposed Algorithm: SBM 1.3

1. Input the plain text P and the key K, both of 128 bits.
2. Generate an LRC code (8 bits), say L_1, out of P.
3. Encrypt P with AES Encryptor to find the cipher text, say C'.
4. Decrypt C' with AES Decryptor to find P_1.
5. Generate an LRC code (8 bits), say L_2, out of P_1.
6. L_1 and L_2 are now compared. If L_1 and L_2 are found to be same, C' is fed to the LRC code generator to find L_3.
7. Encrypt P, n number of times with AES Encryptor to find ciphers $\{C_i \mid i = 1 \text{ to } n\}$, n being odd.
8. Majority rule is applied over C_i to find the cipher text C" which is also fed to the LRC code generator to find L_4.
9. If L_3 and L_4 are found to be same, C' and C" are also considered to be same (say C) and if so, C is transmitted through the channel.

3. EXPERIMENTAL RESULTS

We conduct the experiment with a 128-bit input state and a 128-bit key having the Hexadecimal values as follows:

Input State:

42	61	74	72
69	6D	20	6B
6B	6A	53	61
72	69	61	72

Key:

2B	7E	15	16
28	AE	D2	A6
AB	F7	15	88
09	CF	4F	3C

We have encrypted the above plain text (input state) through the above key 9 times. Each time, we have explicitly injected a 1-bit error in a particular inter mediate state, assuming that no other intermediate ciphers but the targeted one are affected by errors for the particular case. Thus we performed the AES encryption with the explicit injection of errors in all the 9 different intermediate states individually. As a consequence, erroneous ciphers have naturally generated at the outputs of all the cases. Each time, on comparing the erroneous cipher with the error free cipher, we get to know the number of bits in error in the output cipher.

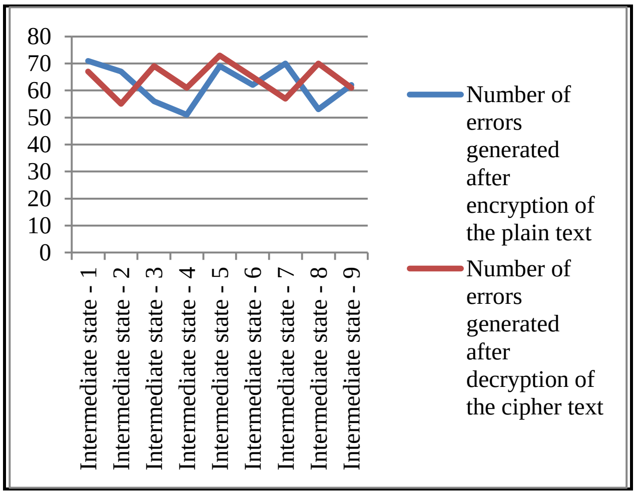
Again we got the erroneous ciphers decrypted back and found all the decrypted cipher texts differing from the actual plain text. The result of the above experiment is summarized below in Table 1.

Table 1. Error propagation effect after injecting a 1-bit error in the intermediate states

Error generation after execution of Process → Error injected in 8th bit position after rounds ↓	Number of errors generated after encryption of the plain text ↓	Number of errors generated after decryption of the cipher text ↓
1st	71	67
2nd	67	55
3rd	56	69
4th	51	61
5th	69	73
6th	62	65
7th	70	57
8th	53	70
9th	62	61

From the above table of data, we find, the average number of errors generated (through propagation over rounds) after encryption of the plain text is 62.33 (out of 128 bits) and naturally, the cipher text with 48.7 % (approximately 50%) in error is not acceptable for the transmission.

The following graph may also be found out from the above table:



As a remedial measure of the error propagation effect, we have conducted an experiment with the same set of plain text and cipher key as per the steps of SBM 1.3:

(i) Input the Message P (128 bits) and the key K (128 bits):

The input state and the key are taken as follows:
Input State:
42 69 6B 72 61 6D 6A 69 74 20 53 61 72 6B 61 72

Key:
2B 28 AB 09 7E AE F7 CF 15 D2 15 4F 16 A6 88 3C

(ii) The LRC code L_1 is generated: The generation of the LRC code L_1 is shown below:

Original Message (Hex) ↓	*Character wise Binary Equivalent* ↓							
42	0	1	0	0	0	0	1	0
69	0	1	1	0	1	0	0	1
6B	0	1	1	0	1	0	1	1
72	0	1	1	1	0	0	1	0
61	0	1	1	0	0	0	0	1
6D	0	1	1	0	1	1	0	1
6A	0	1	1	0	1	0	1	0
69	0	1	1	0	1	0	0	1
74	0	1	1	1	0	1	0	0
20	0	0	1	0	0	0	0	0
53	0	1	0	1	0	0	1	1
61	0	1	1	0	0	0	0	1
72	0	1	1	1	0	0	1	0
6B	0	1	1	0	1	0	1	1
61	0	1	1	0	0	0	0	1
72	0	1	1	1	0	0	1	0
Generated LRC Code (L_1):	0	1	0	1	0	0	0	1

(iii) P is now encrypted with AES Encryptor – 1 to find the cipher text, say C':

We consider a 1-bit error to be injected in the 8th bit position of the intermediate cipher generated after 7th round and the encryption process continues through 3 more rounds of AES encryption. As a consequence, an erroneous cipher text C' is generated at the output as follows:

Generated cipher text C' (Hex):

8A 88 3E 2D DC 16 77 90 4D B3 05 3E CA 04 4D 0C

(iv) The generated cipher text C' is now decrypted with AES Decryptor and at the output of the Decryptor, P' is obtained whose hexadecimal equivalent is as follows:

B2 C9 A7 34 CF 60 C6 24 75 F5 4B CD 9F 97 3C 62

(v) The LRC code L_2 is generated: The generation of the LRC code L_2 is shown below:

Decryptor output (Hex) ↓	*Character wise Binary Equivalent* ↓							
B2	1	0	1	1	0	0	1	0
C9	1	1	0	0	1	0	0	1
A7	1	0	1	0	0	1	1	1
34	0	0	1	1	0	1	0	0
CF	1	1	0	0	1	1	1	1
60	0	1	1	0	0	0	0	0
C6	1	1	0	0	0	1	1	0
24	0	0	1	0	0	1	0	0
75	0	1	1	1	0	1	0	1
F5	1	1	1	1	0	1	0	1
4B	0	1	0	0	1	0	1	1
CD	1	1	0	0	1	1	0	1
9F	1	0	0	1	1	1	1	1
97	1	0	0	1	0	1	1	1
3C	0	0	1	1	1	1	0	0
62	0	1	1	0	0	0	1	0
Generated LRC Code (L_2):	1	1	1	1	0	1	0	1

(vi) L_1 and L_2 are now compared:

In our experiment we obtain L_1 and L_2 as follows:

L_1 =	0	1	0	1	0	0	0	1
L_2 =	1	1	1	1	0	1	0	1

It is seen that L_1 and L_2 are not same, which indicates that some error has occurred and the cipher text generated and hence it is useless to transmit. If L_1 and L_2 were same, C' would be taken to the comparator. Here we consider that the decryption process is error free.

(vii) The message P is encrypted n number of times with AES Encryptor – 2 to find ciphers $\{C_i \mid i = 1 \text{ to } n\}$, n being odd. In our experiment we have taken, n = 3:

Considering the condition that not more than one ((3 – 1) / 2) cipher text can be erroneous, let us say for example three cipher texts as listed below are generated:

D2 06 F1 26 BE 27 E9 EE FA CF AF 6B 55 25 D6 0D

FC 41 16 48 BE C0 16 A7 FC 5C 3F 43 F4 13 F4 A0

FC 41 16 48 BE C0 16 A7 FC 5C 3F 43 F4 13 F4 A0

(viii)Majority rule is applied over $\{C_i \mid i = 1 \text{ to } n\}$ to find the cipher text C":

As a result, the following string C" is realized as follows:

FC 41 16 48 BE C0 16 A7 FC 5C 3F 43 F4 13 F4 A0

(ix) L_3 and L4 are generated through C' and C" respectively and are compared:

If no error occurs in Encryptor – 1, the value of C' will be as follows:

FC 41 16 48 BE C0 16 A7 FC 5C 3F 43 F4 13 F4 A0

Considering that the probability of occurrence of a single bit error amidst the rounds must not reach 0.5, the value of C" will be as follows:

FC 41 16 48 BE C0 16 A7 FC 5C 3F 43 F4 13 F4 A0

Now it is seen that both C' and C" are same and obviously they produce the same LRC code ($L_3 = L_4$). This implies that no error has been injected / generated during the encryption process and hence the error free cipher text C (C' = C" = C) is permitted to get transmitted through the channel.

4. ERROR FREE SELECTIVE AES

The performance of any cryptosystem is measured by two parameters: the level of security it provides and the speed of the encryption and decryption process. Selective encryption is the process of encrypting a fraction (r part) of data / message keeping the remaining portions ((1 – r) part) unencrypted. This process is faster and can reduce the effect of error [11]. Also, it provides a new type of system functionality. The major issue in selective encryption is the proper selection of the part r so as not to deteriorate the level of security. The criteria for the selection of the part r to be encrypted depend on the type of application and media under consideration. As r increases, the level of security increases but the advantage due to selective encryption in terms of enhanced speed of encryption decreases. Again if the r decreases, we may achieve higher speed of encryption and decryption but at the cost of decreased level of security. Here lies the necessity of a tread off. Fig 2 shows the block diagram of the above mentioned hybrid method introducing selective encryption. In the literature various methods are suggested for realizing selective encryption. One of those can be implemented in combination with the proposed algorithm (SBM 1.3) to find out a new algorithm SBM 1.4 in order to speed up the proposed method of operation.

4.1 Proposed Algorithm: SBM 1.4

1. Input Key Words for the message of N words. Message is divided into Q parts each of k blocks. Each block is of M words.
2. Find the occurrence of any keyword in the blocks, starting with the first block in the first part. If it occurs, encrypt that block using the algorithm SBM 1.3 and all the blocks thereafter in the part.
3. Repeat (1 - 2) for all parts, j=1 to Q. When j=Q, the proposed scheme of encryption is complete.

The algorithm used here, makes the scheme applicable generated as a number of streams [12]. It is assumed that the probability of occurrence of the keyword in the message is p. So the probability P that a block of M words be selected for encryption is given as follows:

$$P = \sum_{i=1}^{M} {}^{M}C_i * P_i^{\,i} * (1 - P_i)^{M-i}$$

The probability P_i that the ith block out of i sequential stream blocks is encrypted is given below:

$$P_i = P * (1 - P)^{i-1}$$

Thus, the value of r is obtained as follows:

$$r = (1 / Q) * (\sum_{i=1}^{Q} (Q - i + 1) * P_i)$$

The study reveals, the variation of r under the above mentioned selective encryption algorithm with message size N. From the study it is found that

- With N, r increases as expected, thereby increasing the level of security
- As expected, with the probability of occurrence of the keyword in the message, r increases.

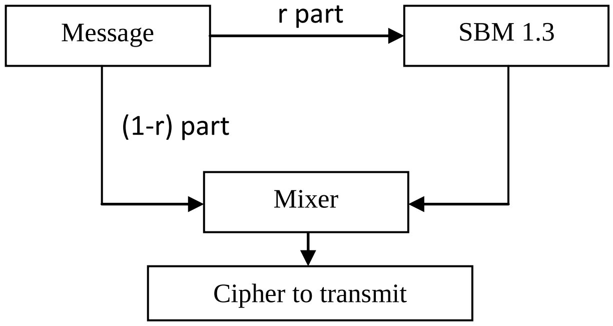


Fig 2: Block diagram of the Error Free Selective AES Encryption module

5. CONCLUSION

In this paper we have made intensive study experimentally on the error propagation effect of AES and proposed a hybrid approach based on the Redundancy Based Technique to tackle the effect. Redundancy Based Technique has a limitation of lower speed of encryption. According to this method, there are two modules which can be executed in parallel. One module involves a particular step of comparison of 8 pairs of bits (instead of 128 pairs of bits) based on Longitudinal Redundancy Check and hence the generation of the error free cipher text where as the other module will approaches to the generation of the error free cipher text based on the majority rule which is a binary decision rule that selects alternatives which have a majority, that is, more than half the votes. Although additional modules of LRC generator and Majority Rule based cipher text generator that have been introduced in our approach result in an extra overhead of the new approach, yet the proposed technique is superior with respect to accuracy because of the fact that the scheme suggests the comparison of error free cipher texts generated through two different and independent modules. Also, the selective encryption is implemented, which provides faster encryption and can reduce the effect of error propagation effect of AES.

Furthermore, if the Encryptor – 2 of the proposed architecture is replaced by n number of independent encryptors, $\{C_i \mid i = 1 \text{ to } n\}$ will generate parallelly, which will obviously increase the speed of encryption, although the hardware complexity increases.

In this context, it may be noted that L_3 and L_4 are generated after the generation of L_1 and L_2. Also, it is clear that L_1 is generated much before (before the dual process of encryption and decryption) the generation of L_2. After the generation of L_2 (and L_1), L_3 generates. And if n increases, L_4 is supposed to be generated much after the generation of L_3. Hence, it is found that the sequence of the generation of the LRC codes is L_1, L_2, L_3, L_4. Since, the generations of these LRC codes are not parallel, a single LRC code generator may be used to generate all these four LRC codes. Also, the same comparator may be used for both the sets of comparisons (L_1 and L_2, L_3 and L_4.) since the comparisons do not take place parallelly. This will surely reduce the hardware complexity.

6. REFERENCES

[1] A. Household et al., “Computer Attack Trends Challenge Internet Security, Security and privacy”, IEEE Computer Society, 2002.

[2] NIST, “Announcing the Advanced Encryption Standard (AES)”, Federal Information Processing Standards Publication, No. 197, 2001.

[3] C. T. Bhunia, “Information Technology Network and Internet”, New Age International publication, 2005.

[4] G. Bertoni et al., “Error Analysis and Detection Procedures for a Hardware Implementation of the Advanced Encryption Standard”, IEEE Transactions on Computers, vol. 52, no. 4, 2003.

[5] C. T. Bhunia et al., Project Work on “AES Error Propagation”, Indian School of Mines, India, 2004.

[6] G. Bertoni et al., “A parity code Based Fault Detection for an implementation of The advanced encryption standard”, Proc IEEE Intl. Symp. Defect and Fault tolerance in VLSI systems, 2002.

[7] B. Sarkar et al., “Modified Redundancy based Technique — a New Approach to Combat Error Propagation Effect of AES”, Springer Journal of Institution of Engineers (India) Series B, 2012, DOI: 10.1007/s40031-012-0012-1.

[8] Williams, Ross N., "A Painless Guide to LRC and CRC Error Detection Algorithms V3.00", September 24, 1996. http://www.repairfaq.org/filipg/LINK/F_crc_v3.html. Retrieved on June 15, 2010.

[9] “Cheng Huang et al., “Erasure Coding in Windows Azure Storage”. http://www.google.co.in/url?sa=t&rct=j&q=benefit+of+l

rc+check+code&source=web&cd=6&ved=0CFgQFjAF&url=https%3A%2F%2Fwww.usenix.org%2Fsystem%2Ffiles%2Fconference%2Fatc12%2Fatc12-final181_0.pdf&ei=1_YrUPCbI8iIrAfl6ID4Cg&usg=AFQjCNGG6JqHzdhe9gM5N8KNjODF2NfAoQ. Retrieved on July 5, 2012.

[10] Conflict Research Consortium, "Majority Rule Processes", International Online Training Program on Intractable Conflict. http://www.colorado.edu/conflict/peace/treatment/majority.htm. Retrieved on July 5, 2012.

[11] B. Sarkar et al., "Approach towards realizing error propagation effect of AES and studies thereof", Int'l J BITM Transaction on EECC, Vol. 2, No. 1, 2010.

[12] C T Bhunia, New Approaches for Selective AES towards Tackling Error Propagation Effect of AES, Asian J of Information Technology, Pakistan, Vol 5, No. 9, pp 1017-1022, 2006.

[13] B. Sarkar et al., "Study and Analysis of Error Propagation Effect of Advanced Encryption Standard", Int'l J HIT Transaction on ECCN, Vol. – 2, No. – 7, 2008.

[14] G. Bertoni et al., "Fault Detection in the Advanced Encryption Standard," Proc. Conf. Massively Parallel Computing Systems (MPCS '02), pp. 92-97, 2002.

[15] G. Bertoni et al., "On the Propagation of Faults and Their Detection in a Hardware Implementation of the Advanced Encryption Standard," Proc. Int'l Conf. Application-Specific Systems, Architectures, and Processors (ASAP '02), pp. 303-312, 2002.

[16] Tom Lookabaugh et al, "Selective Encryption for Consumer Applications", IEEE Communication Magazine, Vol 42, no 5, pp.124-129, April'2004.