World Applied Sciences Journal 29 (Computer Sciences, Engineering and Its Applications): 167-171, 2014
ISSN 1818-4952
© IDOSI Publications, 2014
DOI: 10.5829/idosi.wasj.2014.29.csea.2252

Structure Independent Fault Detection Scheme for Obtaining a Reliable Advanced Encryption Standard

K.P. Thooyamani, R. Udayakumar and V. Khanaa

School of Computing Science, Bharath University, Chennai-73, India

Abstract: This paper presents concurrent fault detection schemes for reaching a reliable AES architecture. Specifically, we propose low-cost structure-independent fault detection schemes for the AES encryption and decryption. The usual and cruel faults reduce its reliability and may cause information leakage. We have obtained new formulation for the fault detection of Sub Bytes and inverse Sub Bytes using the relation between the input and the output of the S-box and the inverse S-box. The reported scheme is independent of the way the S-box and the inverse S-box are constructed. We developed the level 2 comparator for detecting the fault concurrently at every step of the encryption and decryption. The structure-independent scheme reaches the error coverage of approximately 99 percent.

Key words: Advanced encryption standard • Concurrent error detection • Reliable AES • Structure independent fault detection

INTRODUCTION

The National Institute of Standards and Technology initiated a process to select a symmetric key encryption/ decryption algorithm in 1997. Finally, Rijndael algorithm was accepted among other finalists as the Advanced Encryption Standard (AES) in 2001. The fast hardware and software implementations and the high level of security of the AES have led to its widespread usage in different critical applications needing reliable systems and architectures. The AES has been lately utilized for the bit stream security mechanisms.

The objective in using the AES is to transfer the data so that only the desired receiver with a specific key would be able to retrieve the original data [1][2]. However, with the existence of malicious injected faults in non-secure environments, the hardware implementation of the AES does not guarantee that the data are transferred reliably. In fact, several fault attacks on the AES are reported in the literature. These types of attacks are based on injecting faults to the structure of the AES to obtain the confidential information or simply to cause malfunctioning of the AES algorithm. This results in the incorrect output for the AES designs [3]. To make a robust implementation against these attacks, several fault detection schemes have been proposed.

There exist a number of fault detection schemes based on the error detecting codes for the basic parity based schemes [4]. Using a parity bit for each byte in the AES encryption transformations has been presented. A general method of concurrent checking for Substitution Permutation Networks (SPN) has been proposed. These were followed by other fault detection schemes for the entire AES. In these schemes, the output parity bits of each transformation in every round are predicted from the inputs of the corresponding transformation [5]. Then, the comparisons between the predicted parities and the actual parities can be scheduled so that the desired error coverage is obtained. A multiplication-based scheme is presented. In this scheme, the result of the multiplication of the input and the output of the multiplicative inversion is compared with the predicted result of unity. However, this scheme is not suitable for the S-boxes and inverse S-boxes implemented

Corresponding Author: R. Udayakumar, School of Computing Science,
Bharath University, Chennai-73, India.

using lookup tables (LUTs). This is because the output (the input) of the multiplicative inversion in the S-box (the inverse S-box) may not be accessible in the LUT-based implementations [6].

Therefore, the fault detection scheme presented is not applicable for these implementations. We have presented a systematic method for obtaining the fault detection signatures for the multiplicative inversion of the S-boxes (inverse S-boxes).

We have proposed new formulations resulting in novel fault detection schemes for checking Sub Bytes, inverse Sub Bytes and the other transformations in the encryption and the decryption of the AES. The proposed schemes are independent of the method the S-box (respectively, the inverse S-box) is implemented. Thus, they can be applied to both the LUT and composite fields implementations.

We have simulated the proposed fault detection structures for the AES encryption and decryption. Through our simulations after injecting up to 700,000 random stuck-at errors, we have shown that the proposed low-cost schemes reach the error coverage of greater than 99 percent.. Finally, our proposed fault detection schemes and almost all of the previously reported ones have been implemented on the recent Xilinx Virtex FPGAs and their area and delay overheads have been derived and compared. The FPGA implementation results show the low area and delay overheads for the proposed fault detection schemes.

AES Encryption: Considering the input state of an encryption round, the transformations in each round of encryption (except for the last round) are as follows:

Sub Bytes: The first transformation in each round is the bytes substitution (SubBytes) implemented by 16 S-boxes. Let $s_{r,c}\Sigma GF(2^8)$ and $s'_{r,c}\Sigma GF(2^8)$ be the 8-bit input and output of each S-box, respectively.

Shift Rows: In the second transformation, ShiftRows, 4 bytes of the rows of the input state are cyclically shifted to the left and the first row is left unchanged to obtain the output state, i.e., SR(S').

Mix Columns: In the third transformation, MixColumns, the output state is obtained by multiplying a constant matrix with the output state of ShiftRows, SR(S') to obtain the output state of MixColumns, i.e., S''.

Add Round Key: The final transformation is AddRoundKey in which the input state is added with the key of the round.

AES Decryption: In the AES decryption rounds, four transformations, i.e., Inv Shift Rows, Inv Sub Bytes, Add Round Key and Inv Mix Columns, are utilized. Considering S' as the input state of a decryption round, in the first transformation, Inv Shift Rows, similar to ShiftRows in encryption, the first row of the input state remains unchanged. However, the other rows entries are cyclically shifted to the right the next transformation in each round is InvSubBytes implemented by 16 inverse S-boxes [8].

The next transformation is Add Round Key in which the input state is added with the key of the round. Finally, the last transformation, Inv Mix-Columns, is equivalent to multiplying the input state with a constant matrix with hexadecimal entries to obtain the output state of the round.

Among the four transformations in the encryption and the decryption of the AES, only the S-boxes and the inverse S-boxes are nonlinear operations [9]. Furthermore, not only are the S-boxes used in the AES transformations, but they are also used in the key expander unit, generating the keys used in the AES rounds. Therefore, the fault detection schemes of these operations affect the fault detection implementations of the entire AES.

Limitations of Existing System: Existing systems are based on parity based scheme and Error detecting codes. The existence of malicious injected faults in non-secure environments, the hardware implementation of the AES does not guarantee that the data are transferred reliably. It based on injecting faults to the structure of the AES to obtain the confidential information or simply to cause malfunctioning of the AES algorithm. It produces incorrect output for the AES designs. In existing system algorithm-level, round level, or operation-level fault detections are considered [10]. It unable to detect the permanent internal faults or the malicious injected faults lasting for a long period.

New Fault Detection Scheme: We present a systematic method for the fault detection of the multiplicative inversion of the S-box and the inverse S-box. Then, the new scheme for the entire S-box and the inverse S-box is presented.

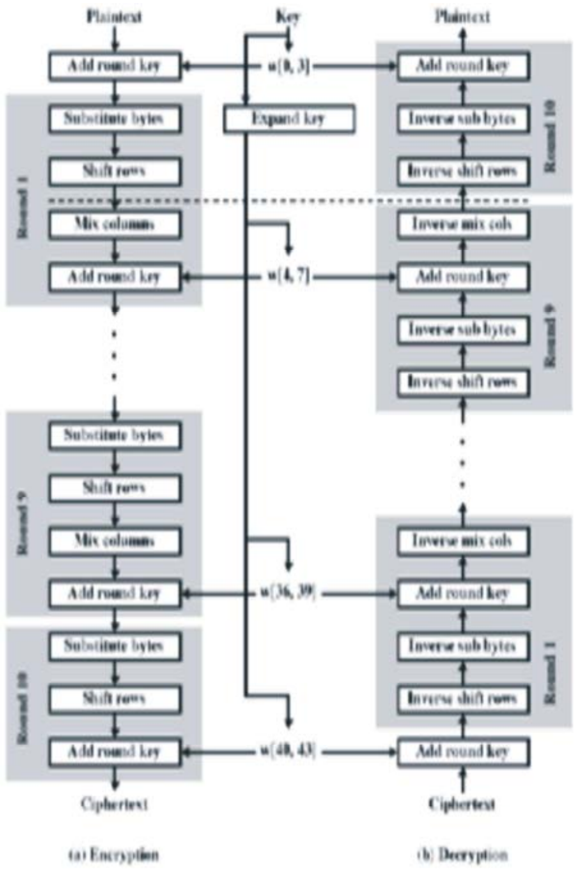

Fig. 1: General AES Encryption and Decryption

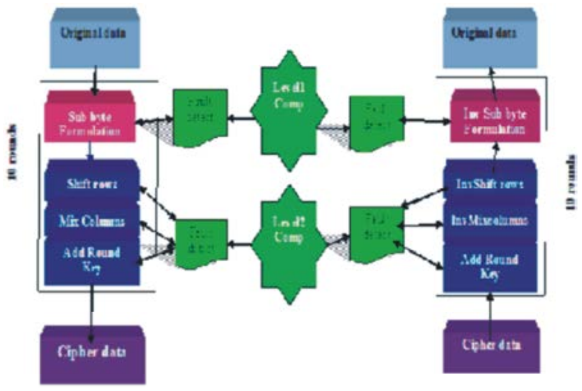

Fig. 2: System Architecture

System Architecture: System architecture shows the overview of my paper which is shown in the Figure 2. Here Level 1comparator and Level 2 comparator are used for the fault detection in sub byte formation and remaining process of AES. Original text is converted into cipher text in AES Encryption. The cipher text was converted into original text in AES Decryption.

System Modules: This paper consists of following modules:

- Subtype formulation and level 1 fault detection module
- Level 2 fault detection in AES encryption module
- Level 2 fault detection in AES decryption module
- Original data conversion module

Sub Bytes Formulation and Level Fault Detection Module: In this module, first get the input data from the user. From the input, we have to form the S Box. From the S Boxes form the Sub bytes [11]. Then, check the sub byte by the level 1 comparator. It uses to detect the fault in the sub byte formation. If fault is detected, reform the sub byte. Otherwise, move to next step and form the copy sub byte. It is called as copy S-box.

Here new formulation is introduced to detect the fault in sub byte formation. That is $P_{(MS'+m)=}u'$. The error detection bit is $e_{r,c}=P_{(MS'+m)}+ u'$. The error bit will be set by comparing the two side result of this formula. By using this formula, one can detect the fault in any sub bytes within 16 sub bytes. After the fault detection, we can get the overall sub byte without fault [12].

The Systematic Scheme for the Multiplicative Inversion:

Let $s=s_7\alpha^7+ s_6\alpha^6+ s_5\alpha^5 +s_4\alpha^4 +s_3\alpha^3 + s_2\alpha^2 +s_1\alpha +s_0$ and $s'= s'_7\alpha^7+ s'_6\alpha^6+ s'_5\alpha^5 +s'_4\alpha^4 +s'_3\alpha^3 + s'_2\alpha^2 +s'_1\alpha +s'_0 \Sigma GF(2^8)$

be the 8-bit input and output of the S-box. Then, the following equation holds for all the possible patterns of s and s':

$$P(Ms'+m)= u'$$

Where

$u'= (s_0 v s_1 v s_2 v s_3 v s_4 v s_5 v s_6 v s_7) v (s_0' v s_1' v s_2' v s_3' v s_4' v s_5' v s_6' v s_7')$

$P(Ms'+m) = s_0(s_b'+s_c') +s_1s_b' +s_2s_d'+ s_3s_4'+ s_4(s_c'+s_3') +s_5 s_a'+s_6(s_d'+s_6')+s_7(s_5'+s_4')$

Where

$$s_a'=s_0'+s_2'+s_3'+s_5'.$$
$$s_b'=s_a'+s_7'.$$
$$s_c'=s_1'+s_4'+s_6'.$$
$$s_d'=s_2'+ s_7'.$$

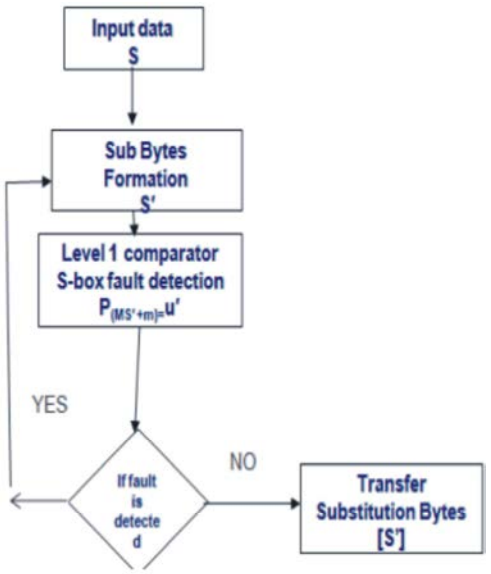

Fig. 3: Sub Bytes Formulation and Level Fault Detection Module

Level 2 Fault Detection in AES Encryption Module: The copy S-box added to the next step in multiply format and the level 2 comparator. AES encryption can perform Shift rows, Mix Columns and Add Round Key. Every step the comparator involves detecting the fault. If fault is detected move to previous step otherwise Move to next step. AES encryption produces the encrypted data. We can transfer encrypted data and the level 2 comparator to the receiver after 10 rounds.

Level 2 Fault Detection in AES Decryption Module: Receiver receives the encrypted data and the level 2 comparator data. Then receiver performs decryption by his/her private key. AES decryption can perform Add Round Key, Inverse Mix Columns and Inverse Shift rows. Every step the comparator involves detecting the fault. If fault is detected move to previous step otherwise Move to next step. After the completion of ten rounds, finally we get overall sub byte.

Original Data Conversion Module: Check the sub byte by the level 1 comparator. If fault is, detected Display message as fault is detected. otherwise perform Inverse Sub byte that produce original data. Here level1 comparator uses same formula in AES encryption to detect the fault.

Further Improvements: Using sub expression sharing, the proposed fault detection scheme for a typical AES decryption round can be modified so that its hardware complexity is reduced. The error indication flags of InvShiftRows and InvSubBytes are obtained utilizing the output state of InvSubBytes, i.e., This output state is also used in obtaining the error indication flags of AddRoundKey and InvMixColumns. Therefore, similar to the fault detection scheme for the AES encryption, we can perform sub expression sharing to obtain these two sets of error indication flags to have low-complexity fault detection scheme of the AES decryption.

CONCLUSION

In this paper, new fault detection schemes, which are independent of the structures of the S-boxes and the inverse S-boxes have been presented for the AES encryption and decryption. We can concurrently detect the fault in the encryption and decryption by use of level 1 comparator. We developed the level 2 comparator for detecting the fault for concurrently at every step of the encryption and decryption. These structure-independent schemes reach the error coverage of approximately 100 percent. The structure-independent schemes have the highest efficiencies, showing reasonable area.

REFERENCES

1. Bertoni, G., L. Breveglieri, I. Koren, P. Maistri and V. Piuri, 2003. “Error Analysis and Detection Procedures for a Hardware Implementation of the Advanced Encryption Standard,” IEEE Trans. Computers, vol. 52.
2. Dusart, P., G. Letourneux and O. Vivolo, 2003. “Differential Fault Analysis on AES,” Proc. Int’l Conf. Applied Cryptography and Network Security (ACNS ’03).
3. Karri, R., G. Kuznetsov and M. Goessel, 2003. “Parity-Based Concurrent Error Detection of Substitution-Permutation Network Block Ciphers,” Proc. Int’l Workshop Cryptographic Hardware and Embedded Systems (CHES ’03), pp: 11.
4. National Institute of Standards and Technologies, 2001. “Announcing the Advanced Encryption Standard (AES),”Federal Information Processing Standards Publication, pp: 197.
5. Piret, G. and J.J. Quisquater, 2003. “A Differential Fault Attack Technique against SPN Structures, with Application to the AES and Khazad,” Proc. Int’l Workshop Cryptographic Hardware and Embedded Systems (CHES ’03).
6. Reyhani-Masoleh, A. and M. Hasan, 2004. “Low Complexity Bit Parallel Architecture for Polynomial Basis Multiplication over GFð2mÞ,” IEEE Trans. Computers, pp: 53.

7. Zhang, X. and K.K. Parhi, 2004. "High-Speed VLSI Architectures for the AES algorithm," IEEE Trans. Very Large Scale Integration Systems, pp: 12.
8. Zimmermann, G. and W. Fichtner, 1997. "Low-Power Logic Styles: CMOS versus Pass-Transistor Logic," IEEE J. Solid-State Circuits, pp: 32.
9. Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. Online answerback and reply-voice recording, Middle-East Journal of Scientific Research, ISSN: 1990-9233, 15(12): 1861-1865.
10. Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. A frame work for modelling task coordination in Multi-agent system, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1851-1856.
11. Saravanan, T., V. Srinivasan and R. Udayakumar, 2013. Images segmentation via Gradient watershed hierarchies and Fast region merging, Middle-East Journal of Scientific Research, ISSN: 1990-9233, 15(12): 1680-1683.
12. Udayakumar, R., V. Khanna, T. Saravanan and G. Saritha, 2013. Cross Layer Optimization For Wireless Network (Wimax), Middle-East Journal of Scientific Research, ISSN: 1990-9233, 16(12): 1786-1789.