Research Journal of Applied Sciences, Engineering and Technology 12(1): 19-26, 2016
DOI:10.19026/rjaset.12.2299
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: June 11, 2015 Accepted: August 5, 2015 Published: January 05, 2016

Research Article

Design of Low Complexity Fault Detection Scheme for AES using Composite Field Arithmetic

[1]G.I. Shamini and [2]Sunil Raj
[1]Department of Electronics and Communication Engineering, Sathyabama University, Chennai,
[2]Department of Electronics and Communication Engineering, Government Engineering College, Idukki, India

Abstract: The Advanced Encryption Standard (AES) is the symmetric cryptography standard that can be used to protect the electronic data. The natural and malicious injected faults may cause confidential information leakage and also reduce its reliability. In this study, we have explained a low complexity fault detection schemes for the AES architecture. The proposed work is low-complexity fault detection schemes using composite fields in polynomial basis for the AES encryption and decryption. These schemes are independent of the existing S-box and inverse S-box constructed. Here we have developed a new technique for the fault detection of subbyte and inverse subbyte using multiplicative inversion and affine transformation of the S-box and the inverse S-box. These are constructed in S-box and the inverse S-box. So this scheme can be used for the S-boxes and the inverse S-boxes in composite fields subbyte and inverse subbyte and using ROM. The proposed AES Fault detection scheme is coded in VHDL (Very High Speed Integrated Circuits Hardware Description Language), synthesized and simulated using EDA (Electronic Design Automation) tool-XilinxISEVirtex FPGA (http://www.xilinx.com/.). Finally the results are compared with Conventional ROM based subbyte and inverse subbyte to show the significant improvement in its efficiency in terms of path delay, speed and area.

Keywords: Advanced Encryption Standard (AES), composite field, decryption, encryption, fault detection, polynomial basis, S-box

INTRODUCTION

The Advanced Encryption Standard (AES) is the symmetric key cryptography standard that can encrypt and decrypt the electronic data. In encryption, AES accepts a plaintext (which is limited to 128 bits) and a key for generating the ciphertext. The key can be specified to be 128 bits (AES-128). In AES-128, the ciphertext is generated after 10 cycles of repetition. For encryption, each round, except the final round, consists of four transformations which includes Sub Bytes (which is implemented by 16 S-boxes), Shift Rows, Mix Columns, AddRoundKey. The decryption transformations are the reverse of the encryption transformations which is utilised to obtain original plain text from the cipher text. Among the transformations, the nonlinear ones are the S-boxes in the encryption and the inverse S-boxes in the decryption. It occupies much of the total AES encryption or decryption area.

There exist many schemes for detecting the faults in the AES hardware implementation, see for example (Karri *et al*., 2002; Rijmen, 2000; Satoh *et al*., 2001; Satoh *et al*., 2008; Mozaffari-Kermani and Reyhani-Masoleh, 2008). Among them, the schemes presented in Karri *et al*. (2001) and Maistri and Laveugle (2008) are independent of the ways the S-box and inverse S-box in the hardware implementation. The fault detection schemes using memories (ROMs) for the S-box and the inverse S-box are there. Further rmore, a fault tolerant scheme which is resistant to fault attacks is presented in Moratelli *et al*. (2008).

Either the parity-based scheme proposed in Bertoni *et al*. (2002) or the duplication approach is implemented to protect the combinational logic blocks used in the four transformations of the AES. Moreover, for storing the expanded key and the state matrix, either the Reed-Solomon error correcting code or Hamming code is utilized for protecting the memories. Our proposed scheme is only applied to the S-box and inverse S-box in composite field polynomial basis. While, the scheme presented in Bertoni *et al*. (2003); Wolkerstorfer *et al*., (2002) uses memories. But for high performance, using ROMs are not preferable. Thus, for high performance AES, the S-box and the inverse S-box are implemented using logic gates in composite fields (Canright, 2005; Yen and Wu, 2006).

Thus the schemes suitable for the S-box and the inverse S box in composite field implementation are

Corresponding Author: G.I. Shamini, Department of Electronics and Communication Engineering, Sathyabama University, Chennai, India

obtained in Kermani and Reyhani-Masoleh (2006) and Mozaffari-Kermani and Reyhani-Masoleh (2008). The approach in Kermani and Reyhani-Masoleh (2006); Karpovsky *et al*. (2004); Wu and Yen (2006) is based on using the parity-based fault detection method for a specific S-box in composite field and polynomial basis for covering all the single malicious faults. For the multiplicative inversion of the S-box, two specific composite fields are treated. Though the transformation and affine matrices are excluded in this approach. Furthermore, in Cohen (2007) Zhang and Parhi (2004, 2006), the fault detection scheme for the multiplicative inversion of a S-box in composite field polynomial basis, the systematic method including predicted parities have been used. The transformation matrices are also advised. Finally, in the parity-based approach in Mozaffari-Kermani and Reyhani-Masoleh (2008), through exhaustive search among all the fault detection S-boxes utilizing five predicted parities using polynomial basis, utmost compact one is obtained. The main objective of the work is to obtain low complexity fault detection schemes using composite field and the result is compared with conventional ROM to get efficient path delay, speed and area.

AES encryption: In this section, we briefly explain about the four transformations used in the AES encryption and decryption (National Institute of Standards and Technologies, 2001). In the AES-128 (128-bit key) transformation implementations, the irreducible polynomial of $P(x) = x^8+x^4+x^3+x+1$ is used for constructing the binary field GF(2^8). Each transformation in every round acts on its 128-bit input denoted as the state. The states are considered as 4×4 matrices whose entries are 8 bits. For example, the input state S with its 8-bit entries, i.e., $s_{r,c}$, 0≤ r, c≤3, is represented as follows:

$$S = [s_{r,c}]^3_{r,c}= 0 \quad (1)$$

Considering (1) as the input state of an encryption round. The transformations in each round, except the final round, are as follows:

SubBytes: In each round the first transformation is the bytes substitution (SubBytes) which is implemented by 16 S-boxes. Let the 8-bit input and output of each S-box $bes_{r,c}$€GF(2^8) and $s'_{r,c}$ € GF(2^8) respectively. The S-box consists of a multiplicative inversion, i.e., $s^{-1}r$, c € GF(2^8), followed by an affine transformation consisting of the matrix Γ and the vector γ to generate the output as:

$$s'_{r,c}=\Gamma s^{-1}_{r,c}+ \gamma= \begin{pmatrix} 1&0&0&0&1&1&1&1\\ 1&1&0&0&0&1&1&1\\ 1&1&1&0&0&0&1&1\\ 1&1&1&1&0&0&0&1\\ 1&1&1&1&1&0&0&0\\ 0&1&1&1&1&1&0&0\\ 0&0&1&1&1&1&1&0\\ 0&0&0&1&1&1&1&1 \end{pmatrix} s^{-1}_{r,c} + \begin{bmatrix} 1\\1\\0\\0\\0\\1\\1\\0 \end{bmatrix} \quad (2)$$

The 8-bit outputs of 16 S-boxes are used to obtain the output state of the SubBytes transformation as:

$$s' = [s'_{r,c}]_{r,c=0} \quad (3)$$

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

$$SR(S' = \begin{pmatrix} s'0,0 & s'0,1 & s'0,2 & s'0,3\\ s'1,1 & s'1,2 & s'1,3 & s'1,0\\ s'2,2 & s'2,3 & s'2,0 & s'2,1\\ s'3,3 & s'3,0 & s'3,1 & s'3,2 \end{pmatrix} = [S'_{(r,c) \bmod 4}]^3_{r,c} = 0 \quad (4)$$

Mix columns: In the third transformation, multiplying a constant matrix with the output state of ShiftRows, SR(S') in (4), to obtain the output state of MixColumns, i.e., the matrix S″, as:

$$S''=[s''_{r,c}]^3_{r,c=0} = \begin{pmatrix} \{2\}h & \{3\}h & \{1\}h & \{1\}h\\ \{1\}h & \{2\}h & \{3\}h & \{1\}h\\ \{1\}h & \{1\}h & \{2\}h & \{3\}h\\ \{3\}h & \{1\}h & \{1\}h & \{2\}h \end{pmatrix} \quad (5)$$

$$SR(S') = \begin{pmatrix} s'0,0 & s'0,1 & s'0,2 & s'0,3\\ s'1,3 & s'1,0 & s'1,1 & s'1,2\\ s'2,2 & s'2,3 & s'2,0 & s'2,1\\ s'3,3 & s'3,2 & s'3,3 & s'3,0 \end{pmatrix} \quad (6)$$

AddRoundKey: The final transformation is *AddRoundKey* in which the input state is added (modulo-2) with the key of the round. Considering the round key input state as the matrix K = [kr, c]3r,c = 0, with entries kr; c, 0≤r, c≤3, the output state of the AddRoundKey transformation, i.e., O, is obtained as:

$$O = [o_{r,c}]^3_{r;c}= 0= S''+K \quad (7)$$

$$O = [o_{r,c}]^3_{r,c} = \begin{pmatrix} \{0e\}h & \{0b\}h & \{0d\}h & \{09\}h\\ \{09\}h & \{0e\}h & \{0b\}h & \{0d\}h\\ \{0d\}h & \{09\}h & \{0e\}h & \{0b\}h\\ \{0b\}h & \{0d\}h & \{09\}h & \{0e\}h \end{pmatrix} S'' \quad (8)$$

FAULT DETECTION SCHEME

The systematic fault detection scheme for the multiplicative inversion of s-box and inverse s-box: This scheme explains the 8-bit input of the multiplicative inversion is multiplied by the 8-bit output. Also the n-bit result (1≤n≤8) of the multiplication is compared with the actually obtained n-bit result, i.e., 1 € GF (2^8). If s≠0 and 0 € GF (2^8). If s = 0 because the multiplicative inversion is also used in the inverse S-box, the same scheme can be used for the inverse S-box.

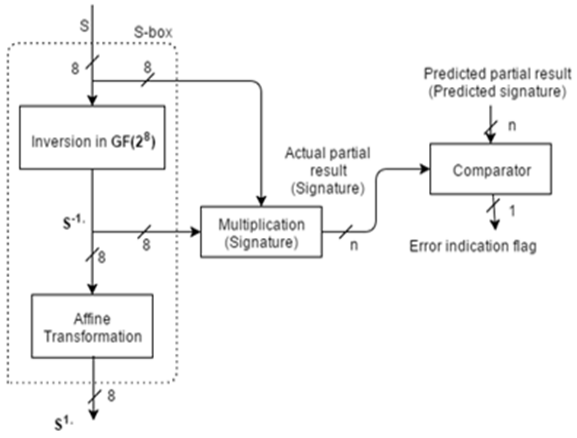

Fig. 1: The scheme based on multiplication for the fault detection of the multiplicative inversion

We present a systematic method for the fault detection scheme for the multiplicative inversion by deriving the matrix-based formulations for the multiplicative inversion in the S-box and inverses-box. We use the following theorem from Mentens *et al.* (2005) to obtain the multiplication of field elements $A=\sum_{i=0}^{m-1} ai\alpha i$ and $B=\sum_{i=0}^{m-1} bi\alpha i$ in the finite field $GF(2^m)$ constructed by the irreducible polynomial of P(x) with the primitive root of αi.

Let $s = s_7\alpha^7+s_6\alpha^6+s_5\alpha^5+s_4\alpha^4+s_3\alpha^3+s_2\alpha^2+s_1\alpha+s0$ and $s^{-1} = s_7^{-1}\alpha^7+s_6^{-1}\alpha^6+s_5^{-1}\alpha^5+s_4^{-1}\alpha^4+s_3^{-1}\alpha^3+s_2^{-1}\alpha^2+s_1^{-1}\alpha^1+s_0^{-1}$ be the 8-bit input and output of the multiplicative inversion in the binary field $GF(2^8)$, respectively. Considering the fact that the result of the multiplication of the 8-bit input s, $s \neq 0$ and the output s^{-1} of the multiplicative inversion is the unity polynomial $1 \in GF(2^8)$, the following is derived from Theorem 1 for the relation between s and s^{-1}.

Corollary 1: Let the vectors corresponding to the input and output of the multiplicative inversion be $s = [s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7]^T$ and $s^{-1} = [s_7^{-1}, s_6^{-1}, s_5^{-1}, s_4^{-1}, , s_3^{-1}, s_2^{-1}, s_1^{-1}, s_0^{-1}]^T$. Then, the matrix formulation of the S-box multiplicative inversion (respectively, the inverse S-box) is as follows in Fig. 1.

Theorem 1 (Mentens *et al.*, 2005): Let $C = \sum_{i=0}^{m-1} ci\alpha ibe$ the multiplication of A and $B \in GF(2^m)$. Then, the coordinates of C can be obtained from:

$$[c_0, c_1, c_2....c_{m-1}] = (L+Q^TU)b \quad (9)$$

where, $b = [b_0, b_1, b_2.....b_{m-1}]^T$

$$L = \begin{pmatrix} a0 & 0 & 0 & 0 & . & . & . & 0 \\ a1 & a0 & 0 & 0 & . & . & . & 0 \\ . & a1 & a0 & 0 & . & . & . & 0 \\ . & . & a1 & a0 & . & . & . & . \\ am-2 & am-3 & a2 & a1 & a0 & . & . & . \\ am-1 & am-2 & . & . & . & . & . & a0 \end{pmatrix} \quad (10)$$

$$U = \begin{pmatrix} 0 & am-1 & am-2 & . & . & a2 & a1 \\ 0 & 0 & am-1 & . & . & a3 & a2 \\ 0 & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & . & . & . & am-1 & am-2 \\ 0 & 0 & . & . & 0 & 0 & am-1 \end{pmatrix} \quad (11)$$

And the (m-1xm) binary matrix Q is obtained as follows:

$$[\alpha_m\ \alpha_{m+1} ..... \alpha_{12m-2}]T = Q\ [1, \alpha, \alpha_2..... \alpha_{m-1}]^T \text{ mod}(p(x)) \quad (12)$$

$$ZS^{-1} = u \quad (13)$$

u = [u' 0 0 0 0 0 0 0] where u' is obtained by logical OR operations of all inputs and outputs, $u' = (s_0 \vee s_1 \vee s_2 \vee s_3 \vee s_4 \vee s_5 \vee s_6 \vee s_7) \vee (s_7^{-1} \vee s_6^{-1} \vee s_5^{-1} \vee s_4^{-1} \vee s_3^{-1} \vee s_2^{-1} \vee s_1^{-1} \vee s_0^{-1})$ Moreover, the modulo-2 additions (XOR operations) of the coordinates of s are shown with commas in indices, e.g., $s_{7,0} = s_7 + s_0$:

$$Z = \begin{pmatrix} s0 & s7 & s6 & s5 & s4 & s7,3 & s7,6,2 & s6,5,1 \\ s1 & s7,0 & s7,6 & s6,5 & s5,4 & s7,4,3 & s6,3,2 & s7,5,2,1 \\ s2 & s1 & s7,0 & s7,6 & s6,5 & s5,4 & s7,4,2 & s6,3,2 \\ s3 & s7,2 & s6,1 & s7,5,0 & s7,6,4 & s7,6,5,3 & s7,6,5,4,2 & s7,6,5,4,3,1 \\ s4 & s7,3 & s7,6,2 & s6,5,1 & s7,5,4,0 & s6,4,3 & s5,3,2 & s7,4,2,1 \\ s5 & s4 & s7,3 & s7,6,2 & s6,5,1 & s7,5,4,0 & s6,4,2 & s5,3,2 \\ s6 & s5 & s4 & s7,3 & s7,6,2 & s6,5,1 & s7,5,4,0 & s6,4,3 \\ s7 & s6 & s5 & s4 & s7,3 & s7,6,2 & s6,6,1 & s7,5,4,0 \end{pmatrix} \quad (14)$$

Proof: We prove (13) for two cases of s = 0 and $s \neq 0$ separately. Let the input ($s \neq 0$) be a nonzero field element in $GF(2^8)$ generated by $P(x) = x^8+x^4+x^3+x+1$. Then, the multiplicative inversion should generate s-1. Using (12) in Theorem 1 and considering the irreducible polynomial of P(x), the (7×8) matrix Q can be obtained as:

$$Q = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \quad (15)$$

This matrix is obtained by using the representations of $\alpha^8\ \alpha^9 ..... \alpha^{14}$ with respect to the polynomial basis for different rows of Q. Considering A= $s \neq 0$ and B = s^{-1} in Theorem 1, the matrices L and U in (10) and (11) are functions of the 8-bit input vector *s* as:

$$L = \begin{pmatrix} s0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s1 & s0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s2 & s1 & s0 & 0 & 0 & 0 & 0 & 0 \\ s3 & s2 & s1 & s0 & 0 & 0 & 0 & 0 \\ s4 & s3 & s2 & s1 & s0 & 0 & 0 & 0 \\ s5 & s4 & s3 & s2 & s1 & s0 & 0 & 0 \\ s6 & s5 & s4 & s3 & s2 & s1 & s0 & 0 \\ s7 & s6 & s5 & s4 & s3 & s2 & s1 & s0 \end{pmatrix} \quad (16)$$

$$U = \begin{pmatrix} 0 & S7 & S6 & S5 & S4 & S3 & S2 & S1 \\ 0 & 0 & S7 & S6 & S5 & S4 & S3 & S2 \\ 0 & 0 & 0 & S7 & S6 & S5 & S4 & S3 \\ 0 & 0 & 0 & 0 & S7 & S6 & S5 & S4 \\ 0 & 0 & 0 & 0 & 0 & S7 & S6 & S5 \\ 0 & 0 & 0 & 0 & 0 & 0 & S7 & S6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & S7 \end{pmatrix} \quad (17)$$

Substituting Q, L and U (15)-(17) from (9) and denoting Z = L + Q^TU. Since s ≠ (0 0, 0, 0... 0) € GF(2^8), u = 1 and the result of multiplication is:

C = A.B mod P(x) = 1 € GF(2^8)
i.e., c = $[c_0, c_1, . c_7]^T$ = [1 0... 0]^T

Therefore using (9) one can prove that (13) is valid for s ≠ 0. moreover, for s = 0, the output of the multiplicative inversion generates 0 = (0 0.... 0). Thus, all entries of the matrix Z and hence, all eight entries of the left-hand side vector of (13) are equal to zero. In such a case, the vector u = [0 0... 0]^T since the result of the OR operation among all *sis* and s^{-1} *i s* are zero, i.e., u =0. Therefore, the proof is complete.

One can figure out that implementation (13) needs 64 ANDs, 15 ORs and 143 XOR gates. Also it is noted that XOR gates can be reduced to 84, if sub expression sharing is used. If one implements the S-box using the composite field presented in Breveglieri *et al*. (2007), it requires 36 and gates and 123 XOR gates for the original S-box implementation. Then, adding this fault detection scheme would require approximately 91% area overhead. Also the silicon area of an AND is 0.6 that of an XOR gate and is derived assuming that an XOR gate is implemented by 10 transistors.

The proposed fault detection scheme for the S-Box and the inverse S-Box: If the SubBytes implementation in the AES is using LUTs, there will be no means of entry to the output of the multiplicative inversion. Thus, the aforementioned scheme cannot be used. We propose a new scheme which is independent of the way the implementation of S-box and the inverse S-box. First, we obtain the matrix-based S-box formulations as follows:

Theorm 2: Let $s = s_7\alpha^7+s_6\alpha^6+s_5\alpha^5 + s_4\alpha^4+s_3\alpha^3+s_2\alpha^2 + s_1\alpha + s_0$ *a n d* $s' = s'_7\alpha^7+s'_6\alpha^6+s'_5\alpha^5+s'_4\alpha^4+s'_3\alpha^3+s'_2\alpha^2+ s'_1\alpha^1+ s'_0$ be the 8-bit input and output of the S-box. Thus the relation between the input and output of the S-box can be obtained as:

$$Ms' + m = u' \quad (18)$$

Moreover, the (8×8) matrix M is denoted as:

$$M = \begin{pmatrix} s6,5,6 & s5,4,1 & s7,5,3,0 & s6,4,2 & s7,5,3,1 & s7,6,5,2,0 & s7,6,5,4,1 & s7,6,3,0 \\ s7,5,3,2,0 & s6,4,2,1 & s7,6,5,4,3,1 & s6,6,5,4,3,1 s & 7,6,5,4,3,2,1 & s5,3,2,1 & s4,2,1,0 & s6,4,3,1 \\ s6,4,3,1 & s7,5,3,2,0 & s7,6,5,4,2 & s7,6,5,4,3,1 s & 7,6,5,4,3,2,0 & s6,4,3,2 & s5,3,2,1, & s7,5,4,2,0 \\ s7,6,4,0 & s6,5,3 & s6,0 & s7,5 & s6,4 & s6,4,3,2,0 & s7,5,3,2,1 & s7,5,1 \\ s7,6,2,1 & s7,6,5,1,0 & s5,3,1 & s4,2,0 & s3,1 & s6,4,3,2,1 & s7,5,3,2,1,0 & s7,2,3 \\ s7,3,2 & s7,6,2,1 & s6,4,2,0 & s5,3,1 & s4,2,0 & s7,5,4,3,2 & s6,4,3,2,1 & s4,3,0 \\ s4,3,0 & s7,3,2 & s7,5,3,1 & s6,4,2,0 & s5,3,1 & s6,5,4,3,0 & s7,5,4,3,2 & s5,4,1 \\ s5,4,1 & s4,3,0 & s6,4,2 & s7,5,3,1 & s6,4,2,0 & s7,6,5,4,1 & s6,5,4,3,0 & s6,5,2 \end{pmatrix} \quad (19)$$

where, $u' = [u',0,0,0,0,0,0,0]T$ $u' = (s_0 \vee s_1 \vee s_2 \vee s_3 \vee s_4 \vee s_5 \vee s_6 \vee s_7) \vee (s'_7 \vee s'_6 \vee s'_5 \vee s'_4 \vee s'_3 \vee s'_2 \vee s'_1 \vee s'_0)$ and $m = [s_{6;0}, s_{7;6;1}, s_{7;2;0}, s_{6;3;1}, s_{7;6;4;2}, s_{7;5;3}, s_{6;4}, s_{7;5}]^T$.

Proof: We prove (18) for two cases of s ≠ 0 and s = 0 separately. Let 8-bit input s be a nonzero field element in GF(2^8). Considering (2), one can obtain:

$$s^{-1} = \Gamma^{-1} s' + \Gamma^{-1} \gamma = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} s^1_{r,c} + \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (20)$$

By substituting s-1 from (20) into (13), one reaches $Z\Gamma^{-1}$ s'+$Z\Gamma^{-1}\gamma$. Now, let us denote $Z\Gamma^{-1}$ = M and $Z\Gamma^{-1}$ γ = m. Then, the left-hand side of (18) is obtained. Since s ≠0 = (0, 0... 0) € GF(2^8) u' = 1. i.e., the result of multiplication C = AB mod P(x) = 1 € GF(2^8). This implies that the left-hand side of (18) be Zs^{-1} = [1 0... 0] T = u'. Furthermore, because we have Zs^{-1} = Ms'+ m one can prove that (18) is valid for s≠0. Moreover, according to (2), for the input s = 0 = (0, 0... 0) € GF(2^8).

We have the output as s' = $[s'_0\ s'_1 ... s'_7]^T$ = [1 1 0 0 0 1 1 0]^T which corresponds to the field element s' = {63} h = (0 1 1 0 0 0 1 1) € GF(2^8). From the Theorm 2, u' = [0 0... 0]^T since we have u' = $(s_0 \vee s_1 \vee s_2 \vee s_3 \vee s_4 \vee s_5 \vee s_6 \vee s_7) \vee (s'_7 \vee s'_6 \vee s'_5 \vee s'_4 \vee s'_3 \vee s'_2 \vee s'_1 \vee s'_0)$. Then the vector [0 0... 0]^T = u'. Therefore, the proof is complete (Fig. 2).

Let us consider (18) for the input s = 0 = [0 0... 0] 2 € GF(2^8). For this input, the correct output is s' = $\{63\}_h$= (01 1 0 0 0 1 1) € GF(2^8). If the erroneous

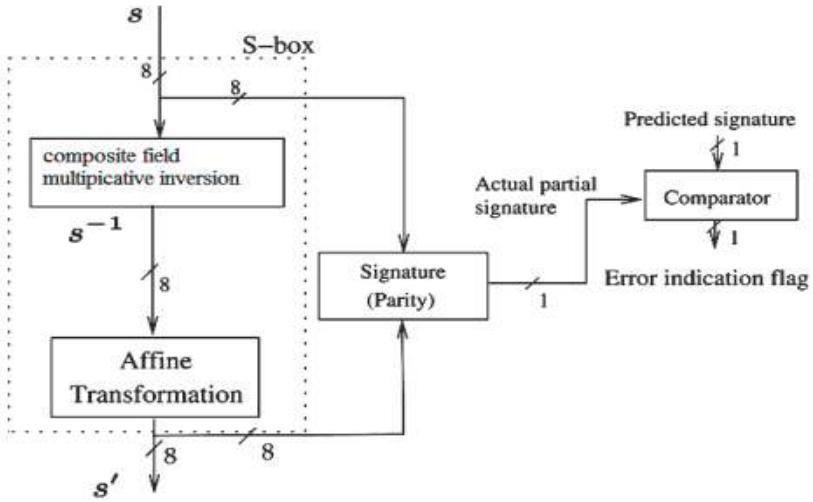

Fig. 2: The proposed fault detection scheme of the S-box

output is not s' = {63}h = (0 1 1 0 0 0 1 1) € GF(2^8) in the right hand side of (18), we have u' = 1, whereas the left-hand side is zero and therefore, the wrong output is detected.

Although checking the formulation of (18) detects all errors in the output of the S-box, its implementation is very costly (Proposition 1). To reduce the overhead of the fault detection scheme (Fig. 2), we have obtained the single-bit parity for the formulation of (18). In Fig. 2, this is obtained in order to compare only 1 bit for an 8-bit data to detect any combination of odd number of erroneous bits at the result of the left-hand side of (18). Thus, one can check the parity of two sides of (18) to obtain 1-bit equation for checking the S-box as follows:

Theorem 2: Let $s = s_7\alpha+s_6\alpha^6+s_5\alpha^5+s_4\alpha^4+s_3\alpha^3+s_2\alpha^2+s_1\alpha+s_0$ and $s' = s'_7\alpha^7+s'_6\alpha^6+s'_5\alpha^5+s'_4\alpha^4+s'_3\alpha^3+s'_2\alpha^2+s'_1\alpha^1+s'_0$ be the 8-bit input and output of the S-box. The equation holds for all the possible patterns of s and s' is as follows:

$$P\ (M\ s'+m) = s_0(s'_b+s'_c)+s_1s'b+s_2s'd+s_3s'4+s_4(s'c+s'_3)+s_5s'_a+s_6(s'_d+s'_6)+s_7(s'_5+s'_4) = u'$$

where

$$s'_a = s'_0+s'_2+s'_3+s'_5., s'_b = s'_a+s'_7, s'_c = s'+s'_4+s'_6 \text{ and } s'_d = s'_2+s'_7 \quad (21)$$

Proof: The parity of two sides of (18) as obtained and we have:

$$P_{(M\,s'+m)} = Pu' = u' \quad (22)$$

where, M, m and u' are presented in Theorem 2. Considering the fact that parity is a linear operation, P(M s'+m) = PM s'+P m. Then, using M and m defined in Theorem 2 one can obtain:

$$P_{M\,s'} = s_as'_0+s_bs'_1+s_cs'_2+s'_3\,(s_a+s_4)\ +s'_4\,(s_b+s_3+s_7)+s'_5(s_a+\ s_7)+s'_6\,(s_b+\ s_6)+s'_7\,(\ s_5+s_c)$$

And $P_m = s_6+s_7$, where $s_a = s_0+s_1+s_5$, $s_b = s_0+s_4$, $s_c = s_a+ s_2+ s_6$, after rearranging, the proof is complete.

Corollary 2: For the fault detection of the inverse S-box, one can use by changing the place of the input and output, i.e., swapping the coordinates of s with s'.

SIMULATION RESULTS

Here we have considered both the single and multiple stuck-at errors for the proposed scheme. And these models covers both natural faults and fault attacks. In the AES encryption or decryption rounds, if exactly 1 bit error appears at the output, this proposed scheme detects it, the error coverage is about 100%. Because in this case, one of the 8-bit four error indication flags in alarms the error. However, multiple stuck-at errors are also considered. Because multiple bits will actually be flipped due to the reason an attacker cannot be able to flip exactly 1 bit in a single stuck at error to gain more information by some technical constraints.

The AES algorithm and low complexity fault detection scheme for composite s-box was described in VHDL and we used the *Modelsim* 6.3 g_*b*1 tool to simulate the code. We analyzed the area and internal and external fault in AES. The fault detection schemes of sub byte in existing and proposed compare the performance in Table 1. Figure 3 to 6 shows the simulation results.

The implementation of s-box requires large number of gates in traditional (LUT) Look up table. Also the unbreakable delay is longer than that of the total delay. Also it is not suitable for resource constrained use

Table 1: Comparision of s-box

Design	Area	Delay	Power
LUT-Based	262144	31.824 ns	35 mw
Composite field based	28514	8.129 ns	34 mw

Fig. 3: Output of encryption for composite field s-box without error

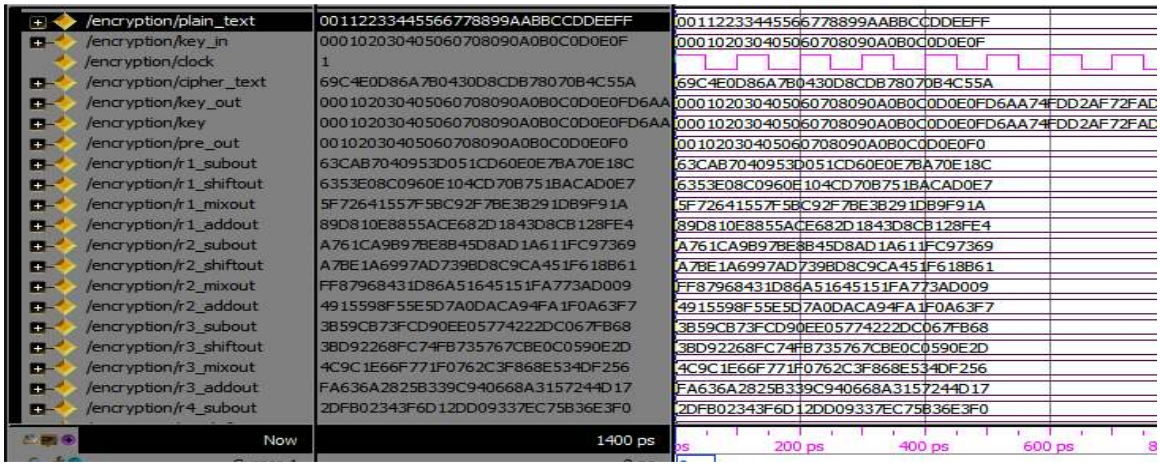

Fig. 4: Output of encryption for composite field s-box with error

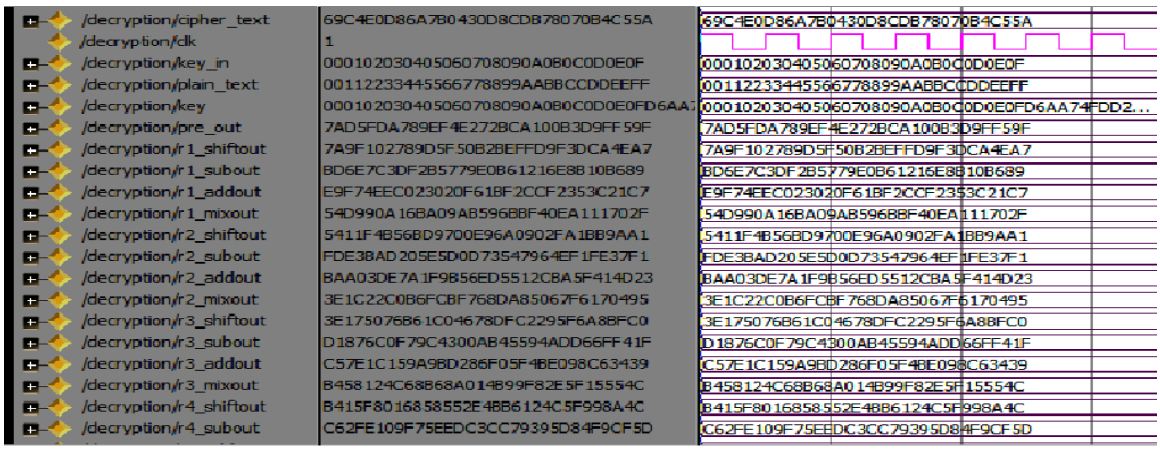

Fig. 5: Output of decryption for composite field s-box without error

Fig. 6: Output of decryption for composite field s-box with error

because it costs a large area. Thus the composite field arithmetic is used to solve these problems. The fault detection scheme implemented by (LUT) look up table in VHDL code synthesised using Xilinix 9.1ISE and get the report of gate count. The gate count value of LUT based fault detection scheme is 262144 logic gates. This fault detection scheme requires 4 times greater than the proposed one. We have to compare it with proposed once and our aim is to reduce the gate count to maximum possible extent.

Composite field implementation of s-box needs less number of gates. We can describe in VHDL and perform synthesis using Xilinix 9.1 In our synthesis report we got a comparatively small value of value of 28514 numbers of gates. Our aim is to reduce the gate count to maximum possible extent. We got gate count almost one tenth of look up table implementation of s box.

Encryption output for composite field s-box without error:

Plain text: x"00112233445566778899aabbccd deeff"
Key: x"000102030405060708090a0b0c0d0e0f"
Cipher text: x"69c4e0d86a7b0430d8cdb78070b 4c55"

Figure 3 shows the output for composite field s-box without error any for a particular input. Theta, eta, gamma, sigma values are obtained and the fault output value is zero.

Encryption output for composite field s-box without error: Figure 4 shows the Output for encryption for composite field s-box without error for a particular 128-bits input and128-bit input key. Fault detection scheme implemented by composite field s-box and detect a internal fault. We will consider the initial round pre_out output value and replace output of pre_out with another 128-bit value and stimulate with modelsim we will get faulty output Theta, eta, gamma, sigma values are obtained and the fault output value is obtained. In this waveform fault occur in add roundkey transformation in initial round.

Decryption output for composite field s-box without error (Fig. 5):

Decryption output for composite field s-box with error: Figure 6 shows the Output of decryption for composite field s-box with error any for a particular 128-bits cipher input and 128-bit input cipher key of each round. Fault detection scheme implemented by composite field s-box and detect a internal fault. We will consider the internal inverse round key output value and replace output of inverse add round key with another 128-bit value and stimulate with modelsim we will get faulty output Theta, eta, gamma, sigma values are obtained and the fault output value is obtained. In this wave form fault occur in add round key transformation of initial round.

CONCLUSION

We have presented a high performance low complexity parity based fault detection scheme for the AES. These schemes are constructed using the S-box and the inverse S-box using composite fields. We have obtained the least complexity S-boxes and inverse S-boxes including their fault detection circuits. The new fault detection schemes are independent of the structures of the S-boxes and the inverse S-boxes. So that we have used parity based method in s box. Therefore it improves the fault coverage to a greater extent because here the error detection at s box takes two times of it. This simulation results shows that the proposed structure-independent schemes have the highest efficiencies with acceptable error coverage. It

shows reasonable area also the time complexity overheads.

REFERENCES

Bertoni, G., L. Breveglieri, I. Koren, P. Maistri and V. Piuri, 2002. A parity code based fault detection for an implementation of the advanced encryption standard. Proceeding of the 17th IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT, 2002), pp: 51-59.

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 T. Comput., 52(4): 492-505.

Breveglieri, L., I. Koren and P. Maistri, 2007. An operation-centered approach to fault detection in symmetric cryptography ciphers. IEEE T. Comput., C-56(5): 534-540.

Canright, D., 2005. A very compact S-box for AES. In: Rao, J.R. and B. Sunar (Eds.), CHES, 2005. LNCS 3659, Springer, Berlin, Heidelberg, pp: 441-455.

Cohen, A.E., 2007. Architectures for cryptography accelerators. Ph.D. Thesis, University of Minnesota, Twin Cities.

Karpovsky, M.G., K.J. Kulikowski and A. Taubin, 2004. Differential fault analysis attack resistant architectures for the advanced encryption standard. In: Quisquater, J.J., P. Paradinas, Y. Deswarte and A.A. El Kalam (Eds.), Smart Card Research and Advanced Applications VI (CARDIS, 2004). Kluwer Academic Publishers, Amsterdam, 153: 177-192.

Karri, R., P. Mishra, K. Wu and K. Yongkook, 2001. Fault-based side-channel cryptanalysis tolerant Rijndael symmetric block cipher architecture. Proceeding of the IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT, 2001), pp: 418-426.

Karri, R., P. Mishra, K. Wu and Y. Kim, 2002. Concurrent error detection schemes for fault-based side-channel cryptanalysis of symmetric block ciphers. IEEE T. Comput. Aid. D., 21(12): 1509-1517.

Kermani, M.M. and A. Reyhani-Masoleh, 2006. Parity-based fault detection architecture of S-box for advanced encryption standard. Proceeding of the IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT, 2006), pp: 572-580.

Maistri, P. and R. Leveugle, 2008. Double-data-rate computation as a countermeasure against fault analysis. IEEE T. Comput., 57(11): 1528-1539.

Mentens, N., L. Batina, B. Preneel and I. Verbauwhede, 2005. A systematic evaluation of compact hardware implementations for the Rijndael S-box. Proceeding of the Cryptographers' Track at the RSA Conference (CT-RSA, 2005), pp: 323-333.

Moratelli, C., F. Ghellar, E. Cota and M. Lubaszewski, 2008. A fault-tolerant, DFA-resistant AES core. Proceeding of the IEEE International Symposium on Circuits and Systems (ISCAS, 2008), pp: 244-247.

Mozaffari-Kermani, M. and A. Reyhani-Masoleh, 2008. A lightweight concurrent fault detection scheme for the AES S-boxes using normal basis. Proceeding of the International Workshop Cryptographic Hardware and Embedded Systems (CHES '08), pp: 113-129.

National Institute of Standards and Technologies (NIST), 2001. Announcing the Advanced Encryption Standard (AES). FIPS Publication 197, National Institute of Standards and Technologies, Washington, DC, pp: 51.

Rijmen, V., 2000. Efficient implementation of the Rijndael S-box. Department of ESAT, Katholieke Universiteit Leuven, Leuven, Belgium.

Satoh, A., S. Morioka, K. Takano and S. Munetoh, 2001. A compact Rijndael hardware architecture with S-box optimization. Proceeding of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT, 2001), pp: 239-254.

Satoh, A., T. Sugawara, N. Homma and T. Aoki, 2008. High-performance concurrent error detection scheme for AES hardware. Proceeding of the CHES, pp: 100-112.

Wolkerstorfer, J., E. Oswald and M. Lamberger, 2002. An ASIC implementation of the AES S-boxes. In: Preneel, B. (Ed.), CT-RSA, 2002. LNCS 2271, Springer-Verlag, Berlin, Heidelberg pp: 67-78.

Wu, S.Y. and H.T. Yen, 2006. On the S-box architectures with concurrent error detection for the advanced encryption standard. IEICE T. Fund. Electr., E89-A(10): 2583-2588.

Yen, C.H. and B.F. Wu, 2006. Simple error detection methods for hardware implementation of advanced encryption standard. IEEE T. Comput., 55(6): 720-731.

Zhang, X. and K.K. Parhi, 2004. High-speed VLSI architectures for the AES algorithm. IEEE T. VLSI Syst., 12(9): 957-967.

Zhang, X. and K.K. Parhi, 2006. On the optimum constructions of composite field for the AES algorithm. IEEE T. Circuits-II, 53(10): 1153-1157.