A Low-Power High-Performance Concurrent Fault Detection Approach for the Composite Field S-Box and Inverse S-Box

Mehran Mozaffari-Kermani, *Student Member*, *IEEE*, and Arash Reyhani-Masoleh, *Member*, *IEEE*

Abstract—The high level of security and the fast hardware and software implementations of the Advanced Encryption Standard have made it the first choice for many critical applications. Nevertheless, the transient and permanent internal faults or malicious faults aiming at revealing the secret key may reduce its reliability. In this paper, we present a concurrent fault detection scheme for the S-box and the inverse S-box as the only two nonlinear operations within the Advanced Encryption Standard. The proposed parity-based fault detection approach is based on the low-cost composite field implementations of the S-box and the inverse S-box. We divide the structures of these operations into three blocks and find the predicted parities of these blocks. Our simulations show that except for the redundant units approach which has the hardware and time overheads of close to 100 percent, the fault detection capabilities of the proposed scheme for the burst and random multiple faults are higher than the previously reported ones. Finally, through ASIC implementations, it is shown that for the maximum target frequency, the proposed fault detection S-box and inverse S-box in this paper have the least areas, critical path delays, and power consumptions compared to their counterparts with similar fault detection capabilities.

Index Terms—Advanced encryption standard, composite fields, fault detection, S-box, inverse S-box.

1 Introduction

For the drawbacks of the previous symmetric-key cryptographic standards such as the DES and the 3DES, they have been lately replaced by the Advanced Encryption Standard (AES) [1]. In particular, the AES has overcome the drawbacks of the previous standards in terms of vulnerability to brute force attacks and slow software implementations. Therefore, since its acceptance as the symmetric-key standard in 2001, the AES has been utilized in a variety of security-constrained applications.

Using the AES, the sender and the receiver of the sensitive data share a secret key to ensure the confidentiality of the information. Nonetheless, a malicious attacker can take over the secret key and compromise the standard. One of the methods for extracting the side-channel information is the fault attacks for which several approaches have been introduced, see, for instance, [2], [3], [4], [5], [6], and [7]. It is noted that the internal hardware failures may also result in malfunctioning of the AES encryption/decryption. Consequently, several fault detection schemes have been proposed to date to counteract the fault attacks and detect the natural faults, see, for example, [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], and [25].

• *The authors are with the Department of Electrical and Computer Engineering, The University of Western Ontario, 1151 Richmond Street North, Faculty of Engineering, Thompson Engineering Building, London, Ontario N6A 5B9, Canada. E-mail: {mmozaff, areyhani}@uwo.ca.*

Manuscript received 31 May 2009; revised 15 Mar. 2010; accepted 24 June 2010; published online 4 Apr. 2011.
Recommended for acceptance by C. Metra and R. Galivanche.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TCSI-2009-05-0240.
Digital Object Identifier no. 10.1109/TC.2011.85.

There exist a number of fault detection schemes based on the error detecting codes, see, for example, [9], [10], [11], [12], [13], [14], and [15]. Using one parity bit for each byte of a transformation, one can obtain the structure shown in Fig. 1 for the round i, $1 \leq i \leq 9$, of the encryption of the AES-128 (128-bit key) to achieve a parity-based fault detection scheme. Similar structure can be obtained for the AES-128 decryption. The AES-128 (referred to as the AES hereafter) encryption/decryption has 10 consecutive rounds which are similar except for the last one in which one of the transformations is not used. As seen in Fig. 1, the output parity bits of each transformation in every round of the AES encryption are predicted from the inputs using the prediction boxes denoted by $\hat{P}$ notations. Then, the comparisons between the predicted parities (shown by a matrix with 16-bit entries) and the actual parities (obtained using the actual parity block) in Fig. 1 can be scheduled so that the desired fault detection capability is obtained. Parity predictions of ShiftRows, InvShiftRows, and AddRoundKey are straightforward and those of MixColumns and InvMixColumns can be done using the equations given in [9], [10], [14], and [15]. It is noted that the parity predictions of the S-box and the inverse S-box proposed in [10] are based on look-up table (LUT) implementations in which 512×9 memory cells are used to generate the predicted parity bit as well as the 8-bit output. In Fig. 1, let k_0 be the 128-bit input key to the key expander. Then, all the modified keys, i.e., k_i', $0 \leq i \leq 10$, consist of the 128-bit expanded key k_i and 16-bit parities, if one bit parity is used for each byte. In [11] and [13], instead of using one parity bit or two signatures in case of using the scheme presented in [12] for each byte, one bit parity is used for 128-bit data using the LUT S-boxes.

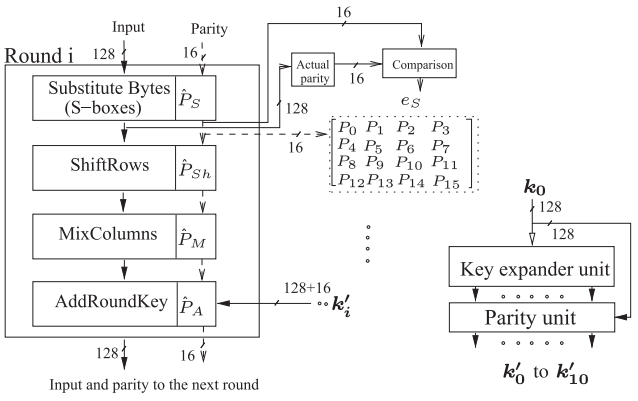

Fig. 1. Parity-based fault detection structure of the ith round in the AES-128 encryption.

The schemes presented in [8] and [16] use the redundant unit fault detection approach. It is noted that this results in the area, power, and delay overheads of approximately 100 percent. In addition, the scheme in [17] proposes using the transformations in an AES round twice for the same data to detect the transient errors. The approach in [18] presents new structures for the S-box and the inverse S-box with higher complexities compared to the original structures for detecting 100 percent of single faults. In [19], a concurrent fault detection scheme based on the merged S-box (SB) and inverse S-box (ISB) is proposed. It is also noted that in the schemes proposed in [20] and [21], all the search space of composite fields are considered for presenting optimum lightweight fault detection schemes. Moreover, the approach in [22] is based on implementing functional redundancy in the AES. The scheme presented in [23] is for all the transformations in the AES encryption/decryption independent of the ways these transformations are implemented. It is also noted that the scheme presented in [24] uses double-data-rate computation for counteracting the fault attacks. Additionally, a fault detection scheme based on the Hamming and Reed-Solomon codes for protecting the storage elements within the AES is proposed in [25]. Furthermore, for the logic elements, the scheme in [10] and the use of the partial duplication of the most vulnerable elements are proposed in [25].

Among the four different transformations in the AES, only the S-box and the inverse S-box are nonlinear. Additionally, all the S-boxes (respectively the inverse S-boxes) occupy much of the total AES encryption (respectively decryption) area and their power consumption is around three fourths of that of the entire AES [26]. LUTs can be utilized for implementing the AES S-boxes and inverse S-boxes in hardware. Nevertheless, this implementation is not suitable for the applications requiring fast and low-complexity AES implementations [27]. Therefore, in this paper, we focus on the low-area implementations of the S-boxes and the inverse S-boxes using composite fields. This approach has received much attention in the literature, see, for example, [26], [27], [28], [29], [30], [31], [32], [33], [34], and [35]. Moreover, there have been low-power implementations for the S-boxes (respectively the inverse S-boxes) such as the ones in [26] and [36]. It is noted that the low-power S-box (respectively inverse S-box) presented in [26] uses composite fields.

We have presented a low-power and high-performance parity-based fault detection approach for the S-box, the inverse S-box, and the merged S-box/inverse S-box within the AES using composite fields. The contributions of this paper are as follows:

- We have obtained new formulations for the five predicted parities for three blocks of the S-box and the inverse S-box. To reach high multiple and burst fault detection capabilities, multiple-bit signatures have been obtained within the blocks constituting more area in the structures of the S-box and the inverse S-box.
- Our simulation results show higher burst fault detection capability for the proposed scheme compared to the previously presented schemes with similar comparable overheads. This can be used as an effective countermeasure against the fault attacks noting that in realistic fault attacks, multiple adjacent bits are actually flipped [37]. Moreover, using the proposed scheme, for multiple random faults, the entire SubBytes and inverse SubBytes are capable of detecting very close to 100 percent of the injected faults.
- Through ASIC implementations, it is shown that for the maximum target frequency, the timing, power, and area of the proposed scheme are the least compared to the schemes with similar fault detection capabilities.

It is noted that the fault detection scheme proposed in this paper can also be applied to both the low-area S-box and inverse S-box presented in [27], [28], [30], [32], and the low-power one proposed in [26].

The organization of this paper is as follows: in Section 2, preliminaries related to the S-box and the inverse S-box are presented. The proposed fault detection approach for the S-box, the inverse S-box, and the merged structures is presented in Section 3. Furthermore, the time and hardware complexities analysis is preformed in this section. In Section 4, the results of the simulations of the proposed approach are presented; through which, the fault detection capabilities are derived. In Section 5, through ASIC implementations, the areas, power consumptions, and critical path delays of the proposed fault detection scheme and the previously reported ones are compared. Finally, conclusions are made in Section 6.

2 Preliminaries

In this section, we describe the S-box and the inverse S-box operations within the AES. Moreover, their low-power architectures using composite fields are presented.

2.1 The S-Box and the Inverse S-Box

Each S-box substitutes an 8-bit input with an 8-bit output using a nonlinear operation. In the S-box, the binary field $GF(2^8)$ is constructed using the irreducible polynomial $P(x) = x^8 + x^4 + x^3 + x + 1$. Let $X \in GF(2^8)$ and $Y \in GF(2^8)$ be the input and output of the S-box, respectively. Then, the S-box consists of the multiplicative inversion, i.e., $X^{-1} \in GF(2^8)$, followed by an affine transformation as:

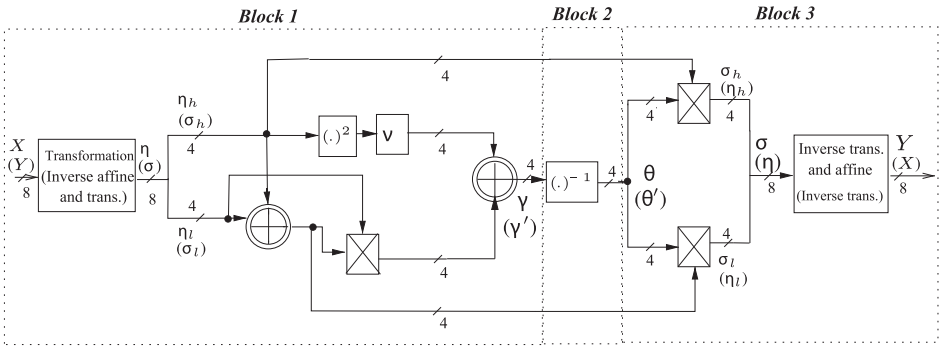

Fig. 2. The architecture of the S-box (respectively the inverse S-box) using composite field and polynomial basis [30].

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^{-1} + \boldsymbol{b} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \boldsymbol{x}^{-1} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \quad (1)$$

where $\boldsymbol{x}^{-1}$ and $\boldsymbol{y}$ are the corresponding vectors to the field elements X^{-1} and Y, respectively.

In the inverse S-box, an 8-bit input is substituted with an 8-bit output using a nonlinear operation which is the reverse of that of the S-box. Let $Y \in GF(2^8)$ and $X \in GF(2^8)$ be the input and output of the inverse S-box, respectively. Then, the inverse S-box consists of the inverse affine transformation and then the multiplicative inversion as follows:

$$\boldsymbol{x}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{y} + \boldsymbol{A}^{-1}\boldsymbol{b} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix} \boldsymbol{y} + \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad (2)$$

where $\boldsymbol{A}$ and $\boldsymbol{b}$ are presented in (1).

It is preferred that the multiplicative inversion of the S-box and the inverse S-box shown in (1) and (2) is performed in the composite fields [28]. This is because the direct calculation of the multiplicative inversion is costly [30]. The structures of the S-box and the inverse S-box using composite field and polynomial basis are shown in Fig. 2. As seen in Fig. 2, for the S-box, the transformation matrix $\boldsymbol{\Psi}$ transforms a field element $X = \sum_{i=0}^{7} x_i \alpha^i$ in the binary field $GF(2^8)$ to the corresponding representation in the composite field $GF(2^8)/GF(((2^2)^2)^2)$ for performing the multiplicative inversion. Then, using the inverse transformation matrix $\boldsymbol{\Psi}^{-1}$, the result of the multiplicative inversion, i.e., X^{-1}, is obtained. This is performed using the irreducible polynomial of $u^2 + u + \nu$. It is noted that the decomposition can be further applied to represent $GF((2^2)^2)$ as a linear polynomial over $GF(2^2)$ and then $GF(2)$ using the irreducible polynomials of $v^2 + v + \Phi$ and $w^2 + w + 1$, respectively. Eventually, as seen in Fig. 2, using the affine transformation, the 8-bit output of the S-box, i.e., Y, is derived. Furthermore, as seen in Fig. 2, for the inverse S-box, the reverse procedure is performed to obtain the output X from the input Y. It is noted that in Fig. 2, the notations for the inverse S-box are presented in parentheses.

All arithmetic operations including the multiplications, the inversion and the squaring in Fig. 2 are over $GF((2^2)^2)$. In Fig. 2, the two concentric circles with a plus inside represent four XOR gates which perform the modulo-2 addition. Moreover, the three finite field multiplications and the inversion in $GF((2^2)^2)$ are shown by crossed rectangles and $(.)^{-1}$, respectively. Furthermore, the multiplication by constant ν and squaring $(.)^2$ in $GF((2^2)^2)$ are shown in this figure. As seen in Fig. 2 for the S-box, for the output of the multiplicative inversion $\sigma_h x + \sigma_l = (\eta_h x + \eta_l)^{-1}$ we have the following [30]:

$$\begin{aligned} \sigma_h &= ((\eta_h + \eta_l)\eta_l + {\eta_h}^2\nu)^{-1}\eta_h, \\ \sigma_l &= ((\eta_h + \eta_l)\eta_l + {\eta_h}^2\nu)^{-1}(\eta_h + \eta_l). \end{aligned} \quad (3)$$

Moreover, for the inverse S-box in Fig. 2, one can swap η and σ in (3) to derive the relation for the multiplicative inversion.

2.2 Low-Power Architectures

In what follows, we present the low-power implementation of the S-box (respectively inverse S-box) presented in [26] using composite field in [30]. For reaching a low-power architecture with acceptable hardware complexity, it is suggested in [26] that the structures are partitioned into three blocks (see Fig. 3). Then, the logic gates within each of these blocks are implemented using two-level logics consisting of the arrays of ANDs and XORs. Although this method

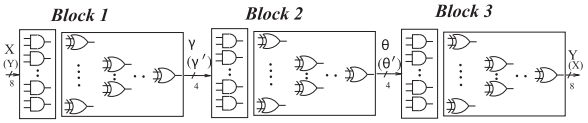

Fig. 3. Low-power S-box (respectively inverse S-box) architecture using composite fields and polynomial basis [26].

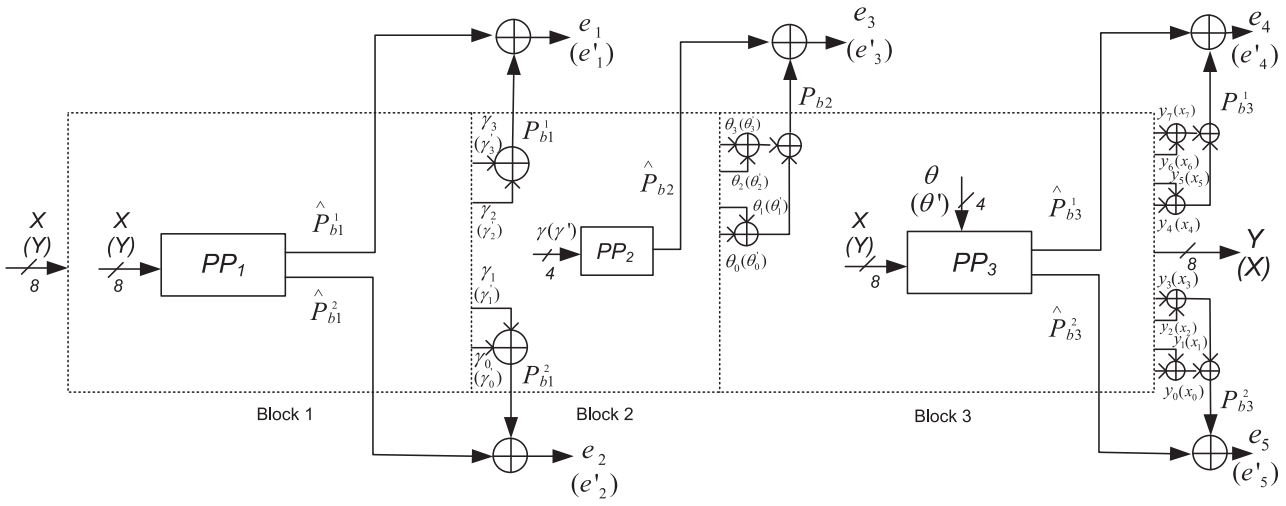

Fig. 4. The proposed parity-based fault detection scheme for the S-box (respectively inverse S-box).

increases the area of the composite fields implementation, it reduces the power consumption significantly [26]. The AND-XOR structure of each block shown in Fig. 3 results in the low number of transitions and thus low-power consumption. This is because the AND array has 50 percent propagation probability of signal transitions. In [26], similar to many other publications such as [27], [28], [30], and [32], the irreducible polynomials $u^2+u+\nu$ and $v^2+v+\Phi$, where $\nu=\{1100\}_2$ and $\Phi=\{10\}_2$, are used for the composite fields. Because of its widespread use in the literature, we also utilize this composite field in this paper. As seen in Figs. 2 and 3, for block 1, a field element X for the S-box (Y for the inverse S-box) in the binary field $GF(2^8)$ is converted to the corresponding representation in the composite field $GF(2^8)/GF(((2^2)^2)^2)$. The output of block 1 is then obtained as $\gamma \in GF(2^4)(\gamma' \in GF(2^4)$ for the inverse S-box). As seen in Figs. 2 and 3, $\theta \in GF(2^4)(\theta' \in GF(2^4)$ for the inverse S-box) is then derived as the output of block 2. Eventually, using the irreducible polynomials $u^2+u+\nu$ and $v^2+v+\Phi$, the output of the S-box, i.e., Y (X for the inverse S-box), is obtained after conversion from the composite field $GF(2^8)/GF(((2^2)^2)^2)$ to the binary field $GF(2^8)$.

3 Proposed Fault Detection Approach

The parity-based fault detection scheme has received much attention in the literature, see, for example, [38], [39], [40], [41], [42], and [43]. In such schemes, the parity of a block is predicted and compared with the actual parity of the block. The result is the error indication flag of the corresponding block which alarms the detected faults. Let τ and ρ be the input and the output of the block under test, respectively. Then, the predicted parity of ρ is obtained from the input τ, i.e., $\hat{P}_\rho(\tau)$, and the actual parity is implemented from the output ρ, i.e., $P_\rho(\rho)$. The comparison between the actual and predicted parities is implemented by an XOR gate to generate the error indication flag $e_\rho=\hat{P}_\rho(\tau)+P_\rho(\rho)$.

In the presented parity-based fault detection scheme, we divide the structures of the S-box and the inverse S-box using polynomial basis into 3 blocks as shown in Fig. 2 so that it can also be used for the low-power structures presented in [26] (see Fig. 3). One can obtain that for the S-box and inverse S-box presented in Fig. 2 [30], blocks 1 and 3 occupy around 86 percent of the area of the entire operations. Therefore, these two blocks are more susceptible to the internal faults and more prone to fault attacks. Consequently, we propose using two bits predicted parities for each of these two blocks. Furthermore, one predicted parity is used for block 2. The details of the proposed schemes are presented below.

3.1 S-Box

In the proposed scheme, five predicted parities are derived for three blocks of the S-box. Then, by comparing these with the five actual parities, five error indication flags are obtained. All five flags should be zero for the error free computations. The proposed fault detection scheme for the S-box is shown in Fig. 4. As seen in this figure, for block 1, two predicted parities, i.e., $\hat{P}^1_{b1}$ and $\hat{P}^2_{b1}$, are obtained using the parity prediction unit (PP_1). As seen from Fig. 4, the predicted parity of the second block $\hat{P}_{b2}$ is obtained by the parity prediction unit (PP_2). Furthermore, for block 3, two predicted parities, i.e., $\hat{P}^1_{b3}$ and $\hat{P}^2_{b3}$, are derived using the parity prediction unit (PP_3).

The derivations of the actual parities are also shown in Fig. 4. As seen from Fig. 4, two actual parities for the two most and least significant bits of γ, i.e., $P^1_{b1}=\sum_{i=2}^{3}\gamma_i$ and $P^2_{b1}=\sum_{i=0}^{1}\gamma_i$, have been derived from the output of block 1 using two trees of XOR gates. Similarly, as shown in Fig. 4, the two actual parities for block 3 are obtained from the output of block 3 for the four most and least significant bits of Y, i.e., $P^1_{b3}=\sum_{i=4}^{7}y_i$ and $P^2_{b3}=\sum_{i=0}^{3}y_i$. In addition, one actual parity is obtained for block 2 as $P_{b2}=\sum_{i=0}^{3}\theta_i$. Then, as shown in Fig. 4, by comparing the predicted and actual parities, the error indication flags of three blocks, i.e., e_1-e_5, are obtained.

The following lemma is used from [30] for the multiplication in $GF((2^2)^2)$ used in blocks 1 and 3. Then, using this lemma, the predicted parities for the S-box in Fig. 4 are derived.

Lemma 1 [30]. *Let $U = (u_3, u_2, u_1, u_0)$ and $V = (v_3, v_2, v_1, v_0)$ be the inputs of a multiplier in $GF((2^2)^2)$. Then, the result of multiplication, i.e., $Z = UV$, is*

$$\begin{aligned} z_3 &= u_3(v_3 + v_2 + v_1 + v_0) + u_2(v_3 + v_1) \\ &\quad + u_1(v_3 + v_2) + u_0 v_3, \\ z_2 &= u_3(v_3 + v_1) + u_2(v_2 + v_0) + u_1 v_3 + u_0 v_2, \\ z_1 &= u_3 v_2 + u_2(v_3 + v_2) + u_1(v_1 + v_0) + u_0 v_1, \\ z_0 &= u_3(v_3 + v_2) + u_2 v_3 + u_1 v_1 + u_0 v_0. \end{aligned} \tag{4}$$

Using Lemma 1, we present the formulations for these five predicted parities in the following theorem, the proof of which has been presented in Appendix A.

Theorem 1. *Let $X \in GF(2^8)$ be the input of the S-box. Then, the five predicted parities of the three blocks of the S-box in Fig. 4, i.e., $\hat{P}^1_{b1}$, $\hat{P}^2_{b1}$, $\hat{P}_{b2}$, $\hat{P}^1_{b3}$, and $\hat{P}^2_{b3}$, are obtained as follows:*

$$\hat{P}^1_{b1} = x_7(D + x_5) + x_4 B + x_3(B + x_4) + x_0 D + x_1 x_2, \tag{5}$$

$$\hat{P}^2_{b1} = x_7(G + x_6) + x_4 I + x_1(C + E) + x_2 \vee x_5 + P_X, \tag{6}$$

$$\hat{P}_{b2} = (\overline{\gamma}_2 \vee \gamma_1)\gamma_0 + P_{\gamma_l}\gamma_3, \tag{7}$$

$$\hat{P}^1_{b3} = \theta_3 H + \theta_2(G + x_7) + \theta_1(J + C) + \theta_0 J, \tag{8}$$

$$\hat{P}^2_{b3} = \theta_3(C + x_0) + \theta_2(H + x_3) + \theta_1(I + x_7) + \theta_0(A + x_2), \tag{9}$$

where $x_1 + x_6 = A$, $x_5 + A = B$, $x_3 + x_2 = C$, $P_X + H = D$, $x_0 + x_6 = E$, $x_2 + x_5 = F$, $F + x_4 = G$, $x_0 + x_7 = H$, $B + C = I$, and $E + F = J$. Furthermore, "+" and $\vee$ represent the modulo-2 addition using an XOR gate and the OR operation, respectively. Moreover, $P_X = \sum_{i=0}^{7} x_i$ and $P_{\gamma_l} = \gamma_1 + \gamma_0$.

3.2 Inverse S-Box

As seen in Fig. 4, similar to the S-box, for blocks 1-3 of the inverse S-box, five predicted parities are derived using the parity prediction units. This is also depicted in Fig. 4. It is noted that the notations for the inverse S-box are denoted by parentheses to be contrasted from those for the S-box. Additionally, similar to the S-box, the actual parities of the three blocks for the inverse S-box are derived using XOR trees. It is noted that for blocks 1 and 3, the actual parities are obtained as $P^1_{b1} = \sum_{i=2}^{3} \gamma'_i$ and $P^2_{b1} = \sum_{i=0}^{1} \gamma'_i$ for block 1 and $P^1_{b3} = \sum_{i=4}^{7} x_i$ and $P^2_{b3} = \sum_{i=0}^{3} x_i$ for block 3. Then, as seen in Fig. 4, by comparing the predicted and actual parities, five error indication flags of three blocks, i.e., e'_1-e'_5, are obtained.

Using Lemma 1 and considering Theorem 1, we present the formulations for the five predicted parities of the inverse S-box for the three blocks shown in Figs. 2 and 4 in the following theorem whose proof is presented in Appendix B.

Theorem 2. *Let $Y \in GF(2^8)$ be the output of the inverse S-box. The five predicted parities of the three blocks of the inverse S-box in Fig. 4 are obtained as follows:*

$$\hat{P}^1_{b1} = y_0 e + y_5(y_4 + y_3 + a) + y_2 b + y_7 y_4 + \overline{b}, \tag{10}$$

$$\hat{P}^2_{b1} = y_1(y_7 + y_5 + h) + y_2 a + y_3(y_5 + y_4) + y_5 h + y_0 + e, \tag{11}$$

$$\hat{P}_{b2} = (\overline{\gamma'}_2 \vee \gamma'_1)\gamma'_0 + P_{\gamma'_l}\gamma'_3, \tag{12}$$

$$\begin{aligned} \hat{P}^1_{b3} &= \theta'_3 \overline{f} + \theta'_2(\overline{P_Y} + d + y_7) + \theta'_1(\overline{c} + y_7 + y_4) \\ &\quad + \theta'_0(\overline{a} + y_4 + y_2), \end{aligned} \tag{13}$$

$$\hat{P}^2_{b3} = \theta'_3(y_1 + d) + \theta'_2(y_0 + g) + \theta'_1(y_6 + g) + \theta'_0(y_1 + f), \tag{14}$$

where $y_6 + y_7 = a$, $y_1 + a = b$, $y_1 + y_2 = c$, $y_3 + y_6 = d$, $c + d = e$, $P_Y + y_4 + y_6 = f$, $\overline{P_Y} + y_2 = g$, and $y_4 + y_0 = h$. Furthermore, "+" and $\vee$ represent the modulo-2 addition using an XOR gate and the OR operation, respectively. Moreover, $P_Y = \sum_{i=0}^{7} y_i$ and $P_{\gamma'_l} = \gamma'_1 + \gamma'_0$.

3.3 Merged S-Box and Inverse S-Box

In some low-complexity implementations that use encryption or decryption at a time, multiplicative inversions of the S-box and the inverse S-box are shared (see, for example, the joint encrypter/decrypter in [30] and [32] and the merged encryption and decryption S-boxes/inverse S-boxes in [31]). The multiplicative inversion in the finite field $GF(2^8)$ is needed for both the S-box and the inverse S-box. Therefore, one can merge them in order to reuse the multiplicative inversion and its parity predictions. It is noted that when there is no need to utilize both the S-box and the inverse S-box at the same time, this merged structure leads to a low-area design. Fig. 5 shows the merged S-box and inverse S-box and their corresponding predicted parities for the three blocks. As seen in this figure, the multiplicative inversion in Fig. 2 is used for both the S-box and the inverse S-box. On the other hand, as seen in Fig. 5, two multiplexers are used for choosing the transformation matrix and the inverse and affine transformations (for the S-box with the select input $SB = 1$) and the inverse affine and transformation matrices and the inverse transformation (for the inverse S-box with the select input $ISB = 1$). The parity prediction unit is also shown in Fig. 5. As seen in this figure, these multiplexers also choose between the predicted parities of blocks 1 and 3 for the S-box and the inverse S-box. As a result, a parity-based fault detection merged structure is obtained.

3.4 Complexity Analysis

In what follows, we obtain the hardware and time complexities of the proposed schemes for the S-box and the inverse S-box. We use two-input gates in the implementation of the predicted parities of the proposed schemes in (5)-(14). We have obtained the number of gates needed for implementing the predicted parities of the S-box in (5)-(9) as 33 XORs, 19 NANDs, two XNORs, and one NOR gate. Moreover, for the inverse S-box, one needs 40 XORs and 19 NANDs to implement (10)-(14). Furthermore, for obtaining the actual parities of blocks 1-3, 2 XORs (one XOR for each of P^1_{b1} and P^2_{b1}), 3 XORs (for P_{b2}), and 6 XORs (three XORs for each of P^1_{b3} and P^2_{b3}) are needed, respectively. Moreover, five XOR gates are used for comparing the five predicted and actual parities to obtain the indication flags. In Section 5, through ASIC implementations, we derive the chip area of the proposed schemes for the S-box and the inverse S-box. Furthermore,

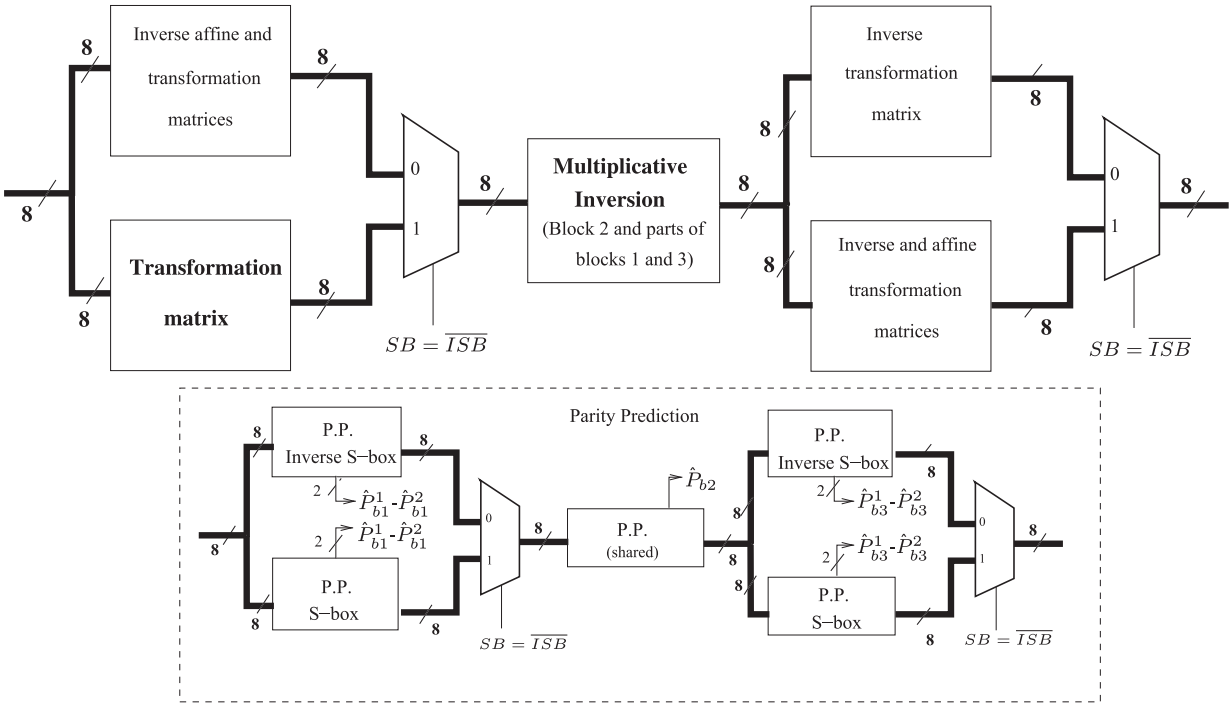

Fig. 5. Merged S-box and inverse S-box and the corresponding predicted parities for different blocks.

the area, critical path delay, and power consumption overheads are derived.

The timing overhead of the proposed scheme can be overlapped by the time needed for performing the operations in blocks 1-3. In other words, as seen in Fig. 4, the predicted parities are obtained concurrently with the time needed for the blocks. Table 1 presents the details of the timings of the three blocks for the S-box and the inverse S-box (presented in Fig. 2) as well as those for obtaining the predicted parities of these blocks. As seen in this table, for all the blocks, the times needed for deriving the predicted parities are less than those of the operations. Therefore, no overhead exists for obtaining these predicted parities. It is also noted that the actual parities are obtained in the time allotted to the next block. Therefore, the only timing overhead is for obtaining the actual parity of block 3 and comparing it with the corresponding predicted parity (see Fig. 4). These are equal to $2T_X$ and $1T_X$, respectively. Therefore, the total timing overhead is $3T_X$ for both operations.

The implementations of the S-box and the inverse S-box using composite fields are area efficient in comparison with those using LUTs. Moreover, the critical path delay can be reduced using subpipelining. In [27], subpipelining of the S-box and the inverse S-box is done by placing one, two, and three-stage registers between the blocks. Although the subpipelining techniques used in [27] are based on the implementations of the S-box and the inverse S-box over $GF((2^4)^2)$, similar pipelining techniques can be used for the composite field $GF(((2^2)^2)^2)$ (see, for example, [32]). The proposed fault detection scheme can take advantage of subpipelining without adding delay to the original pipelined structure. In the pipelined fault detection scheme, we use the parity prediction units of each pipelined block and obtain the error indication flag. According to Table 1, one can observe that the critical path delays of the predicted parity bits of each block of the S-box and the inverse S-box is less than the critical path delay of that block. Therefore, we can use the parity prediction schemes in the pipelined structures of the blocks without affecting the frequency of the clock signal; the predicted parity bits of the blocks are obtained in the same clock cycle as the outputs of the blocks are calculated. Calculating the actual parity and comparing it with predicted parity to obtain the error indication flag can be done in the next clock cycle. Using the abovementioned pipelined structure, one can see that the time overhead will be only one extra clock cycle which may be overlapped with other computations in the pipelined fault detection implementation of AES.

4 Simulation Results

In the following, we evaluate the proposed fault detection scheme for single stuck-at errors, burst faults, and multiple random faults to model both natural faults and fault attacks.

TABLE 1
The Timing Details of the Proposed Concurrent Scheme for the S-Box and the Inverse S-Box

Operation	Block 1		Block 2		Block 3	
	original	predicted parity	original	predicted parity	original	predicted parity
S-box	$10T_X + 1T_A$	$5T_X + 1T_A$	$3T_X + 2T_A$	$2T_X + 1T_A$	$8T_X + 1T_A$	$6T_X + 1T_A$
Inverse S-box	$10T_X + 1T_A$	$5T_X + 1T_A$	$3T_X + 2T_A$	$2T_X + 1T_A$	$7T_X + 1T_A$	$4T_X + 1T_A$

The single stuck-at errors are at the output of the S-box (the inverse S-box). Such errors are covered 100 percent in the proposed scheme which is the same as those of the schemes in [18] and [20]. However, due to the technological constraints, injecting single stuck-at errors may not be applicable in practice [37]. Therefore, we rely on simulations to consider both the burst and the multiple permanent and transient faults; the details of which are presented in the following.

4.1 Burst Faults

Although the fault attacker gains more information through injecting single faults, due to the technological constraints, injecting single stuck-at faults may not be applicable in the practical fault attacks [37]. Therefore, in realistic fault attacks, multiple adjacent bits are actually flipped. Moreover, natural failures can be of the correlated type causing neighboring faults [37]. Consequently, in what follows, we consider the fault detection capability of the proposed scheme for neighboring faults referred to as burst faults.

Because of the nonlinear structure of the S-box (respectively the inverse S-box), the burst faults in a block of the S-box (respectively the inverse S-box) appear as random multiple errors at the output of that block. Moreover, the burst faults that occur in two adjacent blocks appear as multiple random errors at the outputs of the adjacent blocks. For deriving the burst fault detection capability of the proposed scheme, we have performed error simulations for blocks 1-3 of the S-box and the inverse S-box in Fig. 4; the details of which are presented in the following.

Linear Feedback Shift Registers (LFSRs) are used for injecting the errors at the output of one block or two adjacent blocks for modeling the burst faults. The stuck-at error model used forces multiple output bits to be stuck at logic one (for stuck-at one) or zero (for stuck-at zero) independent of the error-free values. We use Fibonacci implementation of the LFSR with four (for the outputs of blocks 1 and 2) or eight (for the random input and output of block 3) output taps for injecting the errors, where the numbers, locations, and types of the errors are randomly chosen. In this regard, according to the maximum sequence length taps presented in [44], the maximum sequence length polynomial for the feedback are selected as $L_1(X) = X^4 + X$ and $L_2(X) = X^8 + X^4 + X^3 + X^2$ for the four and eight output taps, respectively. Moreover, for our simulations, we use the ModelSim SE 6.2d [45]. We have injected 100,000 burst faults at the outputs of the blocks for 100,000 random 8-bit inputs of the S-box and the inverse S-box. Then, we have used the five error indication flags at the outputs of three blocks of the S-box and the inverse S-box to detect the burst faults. The results of our simulations show that for the S-box and the inverse S-box 71,257 and 72,321 of the faults are detected, respectively. This yields to 71.3 and 72.3 percent burst fault detection capabilities for these two structures, respectively. It is noted that these are higher compared to the scheme in [18] for the original S-box and the one in [20], which have the burst fault detection capability of close to 50 percent. The complete comparison of the fault detection capabilities of the proposed schemes and the previous ones are presented in the next section.

TABLE 2
Fault Detection Capabilities of the Proposed Schemes After Injecting 1,000,000 Random Multiple Faults

Operation	Initial values	Detected	Fault Coverage (%)
S-box	$L_2 = \{9D\}_h$ $L_3 = \{AFA2\}_h$	966,324	$\approx 97\%$
	$L_2 = \{B0\}_h$ $L_3 = \{3DA9\}_h$	972,198	$\approx 97\%$
	$L_2 = \{73\}_h$ $L_3 = \{2BBF\}_h$	968,775	$\approx 97\%$
Inverse S-box	$L_2 = \{9D\}_h$ $L_3 = \{AFA2\}_h$	977,760	$\approx 98\%$
	$L_2 = \{B0\}_h$ $L_3 = \{3DA9\}_h$	969,139	$\approx 97\%$
	$L_2 = \{73\}_h$ $L_3 = \{2BBF\}_h$	971,815	$\approx 97\%$

4.2 Multiple Faults

The fault detection capability of the presented scheme depends on the number of the S-box and the inverse S-box blocks and the number of the predicted parities used for them. Two predicted parities have been used for blocks 1 and 3 of the S-box and the inverse S-box which constitute much of the area. Because at least one predicted parity is used for each block of the S-box and the inverse S-box, all odd number of errors in each of three blocks can be detected using the error indication flags. The error indication flags of blocks 1 and 3 can also detect certain even number of errors comprising two odd number of errors in two partitions of these blocks. In the remaining of this section, it is shown that for the entire SubBytes, the error coverage is very close to 100 percent.

For the randomly distributed multiple faults in the entire S-box and inverse S-box, the fault detection capabilities can be obtained. It is noted that in our simulations, we use a transient stuck-at error model. Nonetheless, the simulation results are also the same for the permanent errors, including the permanent internal failures and the malicious fault attacks aiming at destroying the chip. Similar to the burst faults, we use LFSRs for injecting the errors. This is performed using a 16-output tap LFSR for injecting the random multiple errors at the outputs of three blocks utilizing $L_3(X) = X^{16} + X^{12} + X^3 + X$ and an 8-bit LFSR for applying the random input of the S-box or the inverse S-box using $L_2(X) = X^8 + X^4 + X^3 + X^2$ [44].

The results of our simulations for three different initial values of the LFSRs L_2 and L_3 polynomials are depicted in Table 2. As seen in this table, after injecting 1,000,000 random multiple faults, the fault detection capabilities for one S-box or inverse S-box are close to 97 percent. It is interesting to note that for the entire SubBytes or inverse SubBytes, i.e., 16 S-boxes or inverse S-boxes, respectively, injecting this number of multiple faults resulted in the fault detection of very close to 100 percent. As a matter of fact, in this case, the faults are detected by the $5 \times 16 = 80$ flags for the entire SubBytes or inverse SubBytes transformations, yielding to approximately complete fault detection capabilities, i.e., approximately $100 \times (1 - 2^{-80})\%$.

TABLE 3
Comparing the Areas, Critical Path Delays, Power Consumptions, and Fault Detection Capabilities of the Proposed and Previously Presented Fault Detection Schemes for the S-Box Using the 65-nm CMOS Standard Technology

Fault detection scheme	Target frequency: 500 MHz			Target frequency: 1 GHz			Target frequency: 1.1 GHz			Fault coverage (%)	
	Area (θm^2)	Delay (ns)	Total power (θW)	Area (θm^2)	Delay (ns)	Total power (θW)	Area (θm^2)	Delay (ns)	Total power (θW)	Burst faults	Multiple faults
Redundant units [8], United S-box [16] (LUTs)	52.3 $\times 10^3$	1.23	7.2 $\times 10^3$	54.2 $\times 10^3$	0.95	15.4 $\times 10^3$	54.7 $\times 10^3$	0.87	16.9 $\times 10^3$	100%	100%
Parity-based scheme in [13] (256 $\times$ 9 LUT)	29.5 $\times 10^3$	0.59	4.3 $\times 10^3$	29.5 $\times 10^3$	0.59	8.4 $\times 10^3$	29.5 $\times 10^3$	0.59	9.5 $\times 10^3$	$\approx$ 50% (SubBytes)	$\approx$ 50% (SubBytes)
Parity-based scheme in [10] (512 $\times$ 9 LUT)	57.1 $\times 10^3$	0.68	7.8 $\times 10^3$	57.1 $\times 10^3$	0.68	15.6 $\times 10^3$	57.1 $\times 10^3$	0.68	17.1 $\times 10^3$	$\approx$ 50%	$\approx$ 50%
Multiplication approach in [12] (polynomial basis)	876	1.88	630.3	1829	0.96	3000.7	2121	0.88	3600.1	$\approx$ 75% (multiplicative inversion)	$\approx$ 75% (multiplicative inversion)
Structure-independent scheme in [23] (polynomial basis)	754	1.90	574.9	1459	0.97	2263,8	1763	0.87	2902.5	$\approx$ 50%	$\approx$ 50%
Scheme in [18] for the original S-box (polynomial basis)	881	1.92	607.7	1748	0.96	2709.4	Target is not achieved	Target is not achieved	Target is not achieved	$\approx$ 50%	$\approx$ 97%
Parity-based scheme in [21] (polynomial basis)	865	1.82	616.2	1645	0.96	2507.8	1742	0.88	2921.8	$\approx$ 50%	$\approx$ 97%
Parity-based scheme in [20] (normal basis)	858	1.90	620.0	1755	1.0	2672.9	Target is not achieved	Target is not achieved	Target is not achieved	$\approx$ 50%	$\approx$ 97%
Proposed scheme (polynomial basis)	**953**	**1.80**	**712.3**	**1683**	**0.95**	**2600.2**	**1730**	**0.87**	**2912.2**	**71.3%**	$\approx$ **97%**

5 ASIC Implementations and Comparisons

In this section, we present the results of the syntheses we have performed for the proposed and previously presented fault detection schemes of the S-box and the inverse S-box. We have used the STM 65-nm CMOS standard technology [46] for the syntheses. Moreover, VHDL has been used as the design entry to the Synopsys Design Vision [47]. We have set the target frequency as 500 MHz, 1 GHz, and 1.1 GHz corresponding to the delays of 2, 1, and 0.91 ns, respectively. Using Synopsys Design Vision, we have obtained the maximum target frequency in which our fault detection structure can operate without violating the timing constraints. This maximum target frequency has been obtained as 1.1 GHz in the 65-nm technology. The proposed fault detection schemes and the ones presented in [8], [10], [12], [13], [16], [18], [20], [21], and [23] have been synthesized and their areas, delays, and power consumptions are derived. The results for different target frequencies are shown in Table 3 (for the S-box) and Table 4 (for the inverse S-box). As seen in these tables, areas (μm^2), critical path delays (ns), total power consumptions (μW), and fault coverages (percent) are shown. In the following, the syntheses details of the structures are explained.

TABLE 4
Comparing the Areas, Critical Path Delays, Power Consumptions, and Fault Detection Capabilities of the Proposed and Previously Presented Fault Detection Schemes for the Inverse S-Box Using the 65-nm CMOS Standard Technology

Fault detection scheme	Target frequency: 500 MHz			Target frequency: 1 GHz			Target frequency: 1.1 GHz			Fault coverage (%)	
	Area (θm^2)	Delay (ns)	Total power (θW)	Area (θm^2)	Delay (ns)	Total power (θW)	Area (θm^2)	Delay (ns)	Total power (θW)	Burst faults	Multiple faults
Redundant units [8], United S-box [16] (LUTs)	52.3 $\times 10^3$	1.23	7.2 $\times 10^3$	54.2 $\times 10^3$	0.95	15.4 $\times 10^3$	54.7 $\times 10^3$	0.87	16.9 $\times 10^3$	100%	100%
Parity-based scheme in [13] (256 $\times$ 9 LUT)	29.5 $\times 10^3$	0.59	4.3 $\times 10^3$	29.5 $\times 10^3$	0.59	8.4 $\times 10^3$	29.5 $\times 10^3$	0.59	9.5 $\times 10^3$	$\approx$ 50% (Inverse SubBytes)	$\approx$ 50% (Inverse SubBytes)
Parity-based scheme in [10] (512 $\times$ 9 LUT)	57.1 $\times 10^3$	0.68	7.8 $\times 10^3$	57.1 $\times 10^3$	0.68	15.6 $\times 10^3$	57.1 $\times 10^3$	0.68	17.1 $\times 10^3$	$\approx$ 50%	$\approx$ 50%
Structure-independent scheme in [23] (polynomial basis)	783	1.72	581.3	1450	0.97	2262.6	1683	0.89	2893.4	$\approx$ 50%	$\approx$ 50%
Scheme in [18] for the original S-box (polynomial basis)	886	1.85	629.4	1689	0.97	2711.1	1993	0.88	3612.6	$\approx$ 50%	$\approx$ 97%
Parity-based scheme in [21] (polynomial basis)	865	1.85	623.6	1667	0.96	2692.3	1964	0.88	3528.5	$\approx$ 50%	$\approx$ 97%
Parity-based scheme in [21] (normal basis)	855	1.85	574.0	1578	1.0	2374.4	Target is not achieved	Target is not achieved	Target is not achieved	$\approx$ 50%	$\approx$ 97%
Proposed scheme (polynomial basis)	**916**	**1.68**	**636.4**	**1481**	**0.96**	**2200.5**	**1709**	**0.88**	**2812.8**	**72.3%**	$\approx$ **97%**

As seen in Table 3 for the S-box, the first three schemes, i.e., the schemes presented in [8], [16], [13], and [10], use the LUT S-box in their structures. The schemes in [8] and [16] use the S-box followed by the inverse S-box. These can be implemented using two 256×8 LUTs. Then, the result is compared with the input to detect the faults in the structure of the S-box or the inverse S-box. It is noted that although its fault detection capability reaches 100 percent, this method has the critical path delay and the area overheads of close to 100 percent. Furthermore, as seen in Table 3, because of the use of LUT S-box, areas and power consumptions are higher than the schemes using composite fields.

Additionally, the schemes in [13] and [10] use the error detecting codes (parity) for the LUT S-box, where the S-box is expanded. Similar to the scheme in [8] and [16], using the LUT S-box increases the areas and power consumptions of these schemes considerably. In the low-cost scheme presented in [13], the modulo-2 addition of the predicted parities of the input and output of the S-box along with the S-box itself are stored in a 256×9 LUT. Then, a comparison with the actual parities is performed for deriving the error indication flags. As seen in Table 3, the burst and multiple fault detection capabilities of this scheme for the entire SubBytes (not each S-box) is around 50 percent. The parity-based scheme presented in [10] utilizes a 512×9 LUT to store the predicted parities as well as the output of the S-box. This results in reaching the burst and multiple fault detection capability of approximately 50 percent for each S-box at the cost of more area and power consumption and slightly more delay compared to the scheme in [13].

As presented in Table 3, the last six fault detection schemes use the S-box using composite fields; represented either in polynomial basis or normal basis. It is noteworthy that sub-pipelining of these fault detection S-boxes has not been performed and these syntheses are only intended to compare different presented schemes. The scheme in [12] uses two flags for the fault detection of the nonlinear part of the S-box, i.e., the multiplicative inversion. This is performed by comparing the result of multiplying the input and the output of the multiplicative inversion with the actual result, i.e., $\{01\}_2$. As seen in Table 3, this yields to the fault detection capability of approximately 75 percent. The structure-independent scheme in [23] uses one-bit parity in the multiplication scheme for obtaining the fault detection capability of around 50 percent for the S-box. Although the fault detection capability is less than that of [12], as seen in Table 3, better area and power consumption results are obtained.

The results for the proposed scheme in this paper are shown in bold face in Table 3. As depicted in the table, for the target frequency of 1.1 GHz, the proposed scheme in this paper for the S-box has the least area, power consumption, and critical path delay among the schemes that have similar or slightly more fault detection capabilities, i.e., the schemes presented in [8], [16], [18], [21] and [20]. Specifically, compared to the schemes presented in [18], [20], and [21], for the low frequency of 500 MHz, the presented scheme in this paper is faster at the expense of more area. Nonetheless, as seen from the table, the maximum target frequency of 1.1 GHz cannot be achieved for the schemes of [18] and [20]. Nevertheless, in higher frequencies, e.g., 1.1 GHz in Table 3, the presented scheme outperforms the one proposed in [21] in terms of area, power consumption, and delay. It is also noted that the schemes proposed in [18], [20], and [21] yield to the fault detection capability of around 50 percent for the burst faults which is less compared to the presented scheme in this paper.

It is also noted that compared to the schemes with lower fault detection capability in Table 3, for this maximum target frequency, the proposed scheme is more compact. Moreover, it has less power consumption except for the scheme presented in [23]. Nonetheless, the fault detection capabilities of the structure-independent scheme in [23] for burst and multiple faults are around 50 percent, i.e., approximately half of that of the proposed scheme for the multiple faults and less for burst faults. Finally, using subpipelining, the critical path delay of the proposed scheme can be considerably reduced. This can result in even better critical path delays compared to the schemes using LUTs at the expense of more hardware utilizations for the pipelining registers. It is noted that the subpipelined composite field structures are still much more compact than the schemes taking advantage of LUTs.

We have also implemented the proposed scheme for the inverse S-box for the three target frequencies; the results of which are presented in Table 4 in bold face. As seen in this table, in addition, the schemes for the inverse S-box presented in [8], [16], [10], [13], [23], [18], [21] and [20] have been synthesized and their areas, delays, and power consumptions are derived. As seen from Table 4, similar to the S-box, for the low frequency of 500 MHz, the presented scheme for the inverse S-box is the fastest compared to [18], [20], and [21]. Additionally, for the maximum target frequency of 1.1 GHz, it has the lowest area, delay, and power consumption compared to those of [18], [20], and [21]. It is also noted that as presented in Table 4, the target frequency of 1.1 GHz cannot be achieved by the scheme in [21]. As depicted in Table 4, for the highest frequency to achieve, i.e., 1.1 GHz, the proposed scheme in this paper is the most compact scheme with the lowest power consumption compared to the schemes presented in [8], [16], [10], [13], [18], [21] and [20]. It is also noted that similar to the S-box, the fault detection structure of the inverse S-box can be subpipelined so that with a reasonable hardware overhead, the critical path delay is highly reduced. The proposed scheme in this paper has more area and less power consumption compared to the one in [23]. As mentioned previously, however, the fault detection capability of the scheme in [23] for the burst and multiple faults is around 50 percent. This is less than the fault detection capabilities of 97 and 72.3 percent for the proposed scheme for the multiple and burst faults, respectively.

Furthermore, we have compared the areas, critical path delays, and power consumptions of the proposed schemes for the S-box and the inverse S-box with those for the original ones presented in [32]. For this purpose, we have implemented both the original and the fault detection S-box and inverse S-box for several target frequencies ranging from 500 MHz to 1.1 GHz. The results are shown in Fig. 6. As seen in Figs. 6a and 6d for the S-box and the inverse S-box, respectively, the areas of both the original structures (solid lines with $\circ$ marks) and the fault detection ones

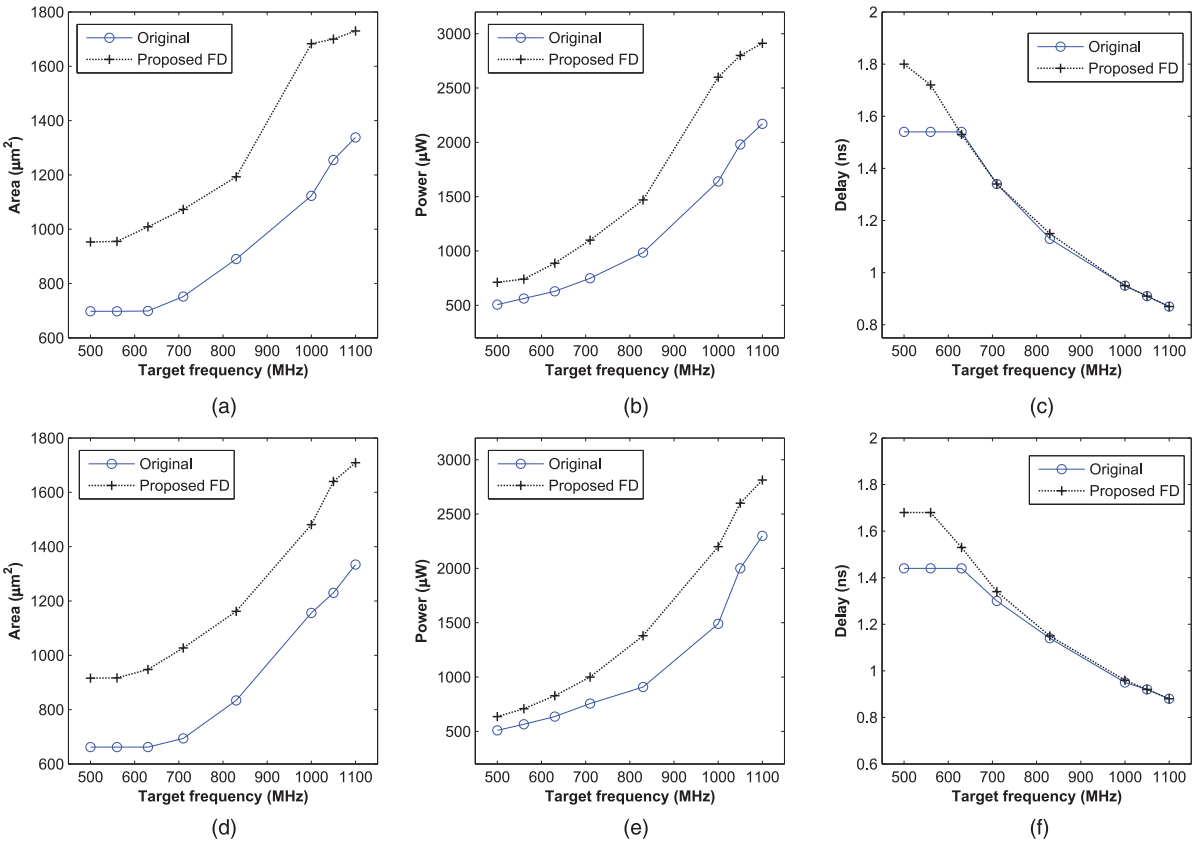

Fig. 6. The areas, critical path delays, and power consumptions of the original [32] and the proposed fault detection S-box and inverse S-box. (a) Area (S-box). (b) Power (S-box). (c) Delay (S-box). (d) Area (Inverse S-box). (e) Power (Inverse S-box). (f) Delay (Inverse S-box).

(dotted lines with + marks) for different target frequencies are depicted. As seen in these figures, as the target frequency increases, it is reached by increasing the occupied area. This yields to having the areas ranging from 698-1,338 and 662-1,334 μm^2 for the original S-box and inverse S-box, respectively. Moreover, for the fault detection S-box and inverse S-box presented in this paper, the areas of 953-1,730 and 916-1,709 μm^2 are achieved, respectively.

Moreover, the results of our implementations for the power consumptions of the original and the fault detection S-box and inverse S-box are depicted in Figs. 6b and 6e, respectively. As seen from these figures, for the low-target frequencies, the power consumptions of the original structures and the fault detection ones are close to each other. Nonetheless, as seen in Figs. 6b and 6e, these differences increase after applying tighter critical path delay constraints. As an example, for the target frequency of 1.1 GHz, the power consumption for the original S-box (respectively inverse S-box) becomes 2.2 mW (2.3 mW). Moreover, for the fault detection S-box (respectively inverse S-box) it reaches 2.9 mW (2.8 mW). Finally, the critical path delays of the original structures and those for the proposed scheme in this paper for the S-box and the inverse S-box are presented in Figs. 6c and 6f. As seen in these figures, for the target frequency of 500 MHz, the critical path delays of the original and the fault detection S-box are 1.54 ns (working frequency of 649 MHz) and 1.80 ns (working frequency of 555 MHz), respectively. Furthermore, for the inverse S-box, the critical path delays of 1.44 (working frequency of 694 MHz) and 1.68 ns (working frequency of 595 MHz) are obtained for the original and fault detection structures, respectively. It is also noted that, for the maximum target frequency to achieve, the original and fault detection S-box (inverse S-box) reaches the critical path delay of 0.87 ns (0.88 ns), i.e., the working frequency of 1.15 GHz (1.14 GHz). As seen in Fig. 6, this is for the cost of the increased areas and power consumptions for the structures.

We conclude this section by deriving the area, delay, and power consumption overheads of the proposed scheme for the S-box and the inverse S-box. To this end, we have considered the areas, delays, and power consumptions of the original operations presented in [32] and the fault detection structures shown in Fig. 6. Then, we have obtained the overheads; the results of which are presented in Fig. 7. The results in this table show that for the low frequency of 500 MHz for the S-box (see the dotted lines with ∘ marks) and the inverse S-box (see the solid lines with + marks), the area overheads are approximately 36 and 38 percent, respectively (see Fig. 7a). Moreover, in this frequency, the overheads for the critical path delays and the power consumptions for the S-box are 16 and 40 percent, respectively. Additionally, for the inverse S-box, for the target frequency of 500 MHz, the critical path delay and the power consumption overheads of 16 and 25 percent are obtained, respectively. However, as we increase the target frequency, the critical path delay overhead decreases (see Fig. 7c). It is noted that as seen in Fig. 7c, no timing overhead is observed for the target frequencies higher than 1 GHz. Finally, as presented in Section 4, with the mentioned overheads, the fault detection scheme proposed in this paper achieves high fault coverages. This makes the

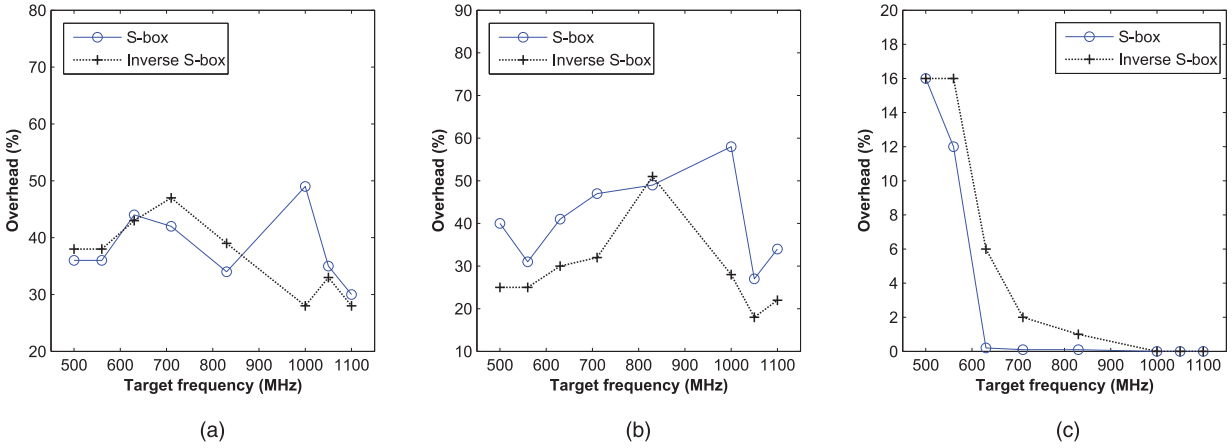

Fig. 7. The area, delay, and power consumption overheads of the proposed schemes for the S-box and the inverse S-box. (a) Area overhead. (b) Power overhead. (c) Delay overhead.

presented fault detection S-box and inverse S-box suitable choices in counteracting the fault attacks and detecting the internal failures.

6 Conclusions

In this paper, we have presented a high-performance fault detection approach for the S-box and the inverse S-box. The proposed scheme is based on using error detecting codes (parities) for the fault detection of the S-box, the inverse S-box, and the merged S-box/inverse S-box using composite fields. The structures of the S-box and the inverse S-box have been divided into three blocks. Then, based on the vulnerability of the blocks to the internal and malicious faults, the number of predicted parities are decided. Utilizing 5 predicted parities for the S-box and the inverse S-box, the fault detection capability of the proposed scheme is close to 97 percent for multiple faults for one S-box and inverse S-box. Moreover, if the entire SubBytes and inverse SubBytes are considered, this becomes very close to 100 percent.

Furthermore, we have performed ASIC implementations using the 65-nm CMOS standard technology for the proposed concurrent fault detection architectures and the previously reported ones. It is shown that for the maximum target frequency of 1.1 GHz, the proposed architectures for the S-box and the inverse S-box have the least areas, power consumptions, and critical path delays compared to the schemes with similar fault coverages. It is noted that using subpipelining, the maximum working frequencies for the proposed scheme in this paper can be considerably increased. Considering the fault detection capabilities of very close to 100 percent and the applicability of the proposed scheme to both the low-area and low-power S-box and inverse S-box, the proposed concurrent fault detection S-box and inverse S-box are suitable choices for having reliable AES encryption/decryption hardware architectures.

Appendix A

Proof of Theorem 1. First, we obtain the two predicted parities of block 1, i.e., $\hat{P}^1_{b1} = \hat{P}_{\gamma_h}$ and $\hat{P}^2_{b1} = \hat{P}_{\gamma_l}$ in (5) and (6). As seen from Fig. 2, block 1 consists of the transformation matrix Ψ, a field multiplication, modulo-2 additions, and squaring followed by the multiplication by the constant ν. From [30], one can obtain that for the input of $\eta_h = (\eta_7, \eta_6, \eta_5, \eta_4)$, the result of the squarer-ν is

$$\eta_h{}^2\nu = (\eta_7 + \eta_4, \eta_7 + \eta_6 + \eta_5, \eta_4, \eta_5). \tag{15}$$

Moreover, using (4) with the inputs $u = \eta_l$ and $v = \eta_h + \eta_l$, one can obtain the result of the field multiplication in this block. By modulo-2 adding the coordinates of $\gamma_h = (\gamma_3, \gamma_2)$ and $\gamma_l = (\gamma_1, \gamma_0)$, i.e., two most and least significant bits of (15) and that of the result of the multiplication, respectively, one can obtain

$$\begin{aligned}\hat{P}^1_{b1} = {} & \eta_3(\eta_6 + \eta_4) + \eta_2(\eta_7 + \eta_6 + \eta_5 + \eta_4) + \eta_1\eta_6 \\ & + \eta_0(\eta_7 + \eta_6) + \eta_7 + \eta_6 + \eta_5 + \eta_2,\end{aligned} \tag{16}$$

$$\hat{P}^2_{b1} = \eta_3\eta_7 + \eta_2\eta_6 + \eta_1\eta_4 + \eta_0(\eta_5 + \eta_4) + \eta_6 + \eta_2 + \eta_0. \tag{17}$$

By substituting the coordinates of η with those of X and reordering the results in (16) and (17), one reaches the following

$$\begin{aligned}\hat{P}^1_{b1} = {} & x_7(x_6 + x_4 + x_3 + x_2 + x_1) + x_4(x_6 + x_5 + x_1) \\ & + x_3(x_6 + x_5 + x_4 + x_1) + x_0(x_6 + x_5 + x_4 + x_3 \\ & + x_2 + x_1) + x_1x_2,\end{aligned} \tag{18}$$

$$\begin{aligned}\hat{P}^2_{b1} = {} & x_7(x_6 + x_5 + x_4 + x_2) + x_4(x_6 + x_5 + x_3 \\ & + x_2 + x_1) + x_1(x_6 + x_3 + x_2 + x_0) + x_2 \vee x_5 + P_X.\end{aligned} \tag{19}$$

Using subexpression sharing, it is straightforward to obtain (5) and (6) from (18) and (19), respectively. It is also noted that the predicted parity of block 2 in (7) is derived from that of block 3 in the scheme in [18] noting that $P_{\gamma_l} = \gamma_1 + \gamma_0$.

Now, we derive the two predicted parities of block 3, i.e., $\hat{P}^1_{b3} = \hat{P}_{Y_h}$ and $\hat{P}^2_{b3} = \hat{P}_{Y_l}$. As seen from Fig. 2, block 3 consists of the mixed inverse and affine transformation matrices and two field multiplications. It is straightforward that using (1), we obtain the formulations for these mixed transformation matrices as follows:

$$\boldsymbol{y} = A\Psi^{-1}\boldsymbol{\sigma} + \boldsymbol{b} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \boldsymbol{\sigma} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}. \tag{20}$$

Eventually, $\hat{P}_{Y_h}$ and $\hat{P}_{Y_l}$, i.e., two predicted parities of block 3 in Fig. 2, are obtained as follows $\hat{P}_{Y_h} = \sigma_6 + \sigma_5 + \sigma_3 + \sigma_1 + \sigma_0$ and $\hat{P}_{Y_l} = \sigma_5 + \sigma_4 + \sigma_3 + \sigma_2$. Then, by multiplying $u = \theta$ and $v = \eta_h + \eta_l$ and also $u = \theta$ and $v = \eta_h$ using (4), one can obtain the coordinates of σ. Substituting these in above, the following is obtained for the two predicted parities of block 3 of the S-box in Fig. 2:

$$\hat{P}^1_{b3} = \theta_3(x_7 + x_0) + \theta_2(x_7 + x_5 + x_4 + x_2) + \theta_1(x_6 + x_5 + x_3 + x_0) + \theta_0(x_6 + x_5 + x_2 + x_0), \tag{21}$$

$$\hat{P}^2_{b3} = \theta_3(x_3 + x_2 + x_0) + \theta_2(x_7 + x_3 + x_0) + \theta_1(x_7 + x_6 + x_5 + x_3 + x_2 + x_1) + \theta_0(x_6 + x_2 + x_1). \tag{22}$$

Then, using subexpression sharing for (21) and (22), one can obtain (8) and (9) and the proof is complete. □

APPENDIX B

Proof of Theorem 2. As seen in Fig. 2, the S-box and the inverse S-box share block 2. Therefore, the predicted parity of this block is the same for them.

Now, we obtain the two predicted parities of block 1, i.e., $\hat{P}^1_{b1}$ and $\hat{P}^2_{b1}$ in (10) and (11). As seen from Fig. 2, block 1 consists of the transformation matrix $\boldsymbol{\Psi}$ preceded by the inverse affine transformation. Moreover, as seen in Fig. 2, similar to the S-box, a field multiplication, modulo-2 additions, and squaring followed by the multiplication by the constant ν are utilized in this block. Similar to the S-box, using (4) with the inputs $u = \sigma_l$ and $v = \sigma_h + \sigma_l$, one can obtain the result of the field multiplication in this block. Moreover, one can obtain that the result of the squarer-ν in Fig. 2 is

$${\sigma_h}^2\nu = (\sigma_7 + \sigma_4, \sigma_7 + \sigma_6 + \sigma_5, \sigma_4, \sigma_5). \tag{23}$$

By modulo-2 adding the two most and least significant bits of the result of the squarer-ν in (23) and that of the result of the multiplication, respectively, one can obtain

$$\hat{P}^1_{b1} = \sigma_3(\sigma_6 + \sigma_4) + \sigma_2(\sigma_7 + \sigma_6 + \sigma_5 + \sigma_4) + \sigma_1\sigma_6 + \sigma_0(\sigma_7 + \sigma_6) + \sigma_7 + \sigma_6 + \sigma_5 + \sigma_2, \tag{24}$$

$$\hat{P}^2_{b1} = \sigma_3\sigma_7 + \sigma_2\sigma_6 + \sigma_1\sigma_4 + \sigma_0(\sigma_5 + \sigma_4) + \sigma_6 + \sigma_2 + \sigma_0. \tag{25}$$

One can substitute the coordinates of σ with those of Y using (2). This is performed by utilizing the following as the result of mixing the inverse affine and transformation matrices

$$\boldsymbol{\sigma} = \Psi A^{-1}\boldsymbol{y} + \Psi A^{-1}\boldsymbol{b} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \boldsymbol{y} + \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}. \tag{26}$$

Then, by reordering the result in (24) and (25), the following is derived

$$\hat{P}^1_{b1} = y_0(y_6 + y_3 + y_2 + y_1) + y_5(y_7 + y_6 + y_4 + y_3) + y_2(y_7 + y_6 + y_1) + y_7y_4 + \overline{y_7 + y_6 + y_1}, \tag{27}$$

$$\hat{P}^2_{b1} = y_1(y_7 + y_5 + y_4 + y_0) + y_2(y_7 + y_6) + y_3(y_5 + y_4) + y_5(y_4 + y_0) + y_6 + y_3 + y_2 + y_1 + y_0. \tag{28}$$

Using subexpression sharing, it is straightforward to obtain (10) and (11) from (27) and (28), respectively.

Now, we derive the two predicted parities of block 3 of the inverse S-box in Fig. 2, i.e., $\hat{P}^1_{b3} = \hat{P}_{X_h}$ and $\hat{P}^2_{b3} = \hat{P}_{X_l}$. As seen from Fig. 2, block 3 consists of the inverse transformation and two field multiplications. It is straightforward that considering the inverse transformation matrix we obtain $\hat{P}_{X_h}$ and $\hat{P}_{X_l}$ as follows $\hat{P}_{X_h} = \eta_7 + \eta_5 + \eta_4 + \eta_1$ and $\hat{P}_{X_l} = \eta_7 + \eta_6 + \eta_5 + \eta_2 + \eta_0$. Then, by multiplying $u = \theta'$ and $v = \sigma_h + \sigma_l$ and also $u = \theta'$ and $v = \sigma_h$ using (4), the coordinates of η are obtained. Substituting these in above, the following is derived

$$\hat{P}^1_{b3} = \theta'_3(P_Y + y_6 + y_4 + 1) + \theta'_2(P_Y + y_7 + y_6 + y_3 + 1) + \theta'_1(y_7 + y_4 + y_2 + y_1 + 1) + \theta'_0(y_7 + y_6 + y_4 + y_2 + 1), \tag{29}$$

$$\hat{P}^2_{b3} = \theta'_3(y_6 + y_3 + y_1) + \theta'_2(P_Y + y_2 + y_0 + 1) + \theta'_1(P_Y + y_6 + y_2 + 1) + \theta'_0(P_Y + y_6 + y_4 + y_1). \tag{30}$$

Then, using subexpression sharing for (29) and (30), one can obtain (13) and (14) and the proof is complete. □

ACKNOWLEDGMENTS

The authors would like to thank the reviewers for their comments. This work has been supported in part by an NSERC Discovery grant awarded to A. Reyhani-Masoleh.

REFERENCES

[1] Nat'l Inst. of Standards and Technologies, "Announcing the Advanced Encryption Standard (AES)," Federal Information Processing Standards Publication, no. 197, Nov. 2001.

[2] J. Blömer and J.P. Seifert, "Fault Based Cryptanalysis of the Advanced Encryption Standard (AES)," *Proc. Financial Cryptography*, pp. 162-181, Jan. 2003.

[3] G. Piret and J.J. Quisquater, "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),* pp. 77-88, Sept. 2003.

[4] P. Dusart, G. Letourneux, and O. Vivolo, "Differential Fault Analysis on AES," *Proc. Int'l Conf. Applied Cryptography and Network Security (ACNS '03),* pp. 293-306, Oct. 2003.

[5] C. Giraud, "DFA on AES," *Proc. Advanced Encryption Standard,* pp. 27-41, May 2004.

[6] J. Blömer and V. Krummel, "Fault Based Collision Attacks on AES," *Proc. Int'l Workshop Fault Diagnosis and Tolerance in Cryptography (FDTC '06),* pp. 106-120, Oct. 2006.

[7] J. Takahashi, T. Fukunaga, and K. Yamakoshi, "DFA Mechanism on the AES Key Schedule," *Proc. Int'l Workshop Fault Diagnosis and Tolerance in Cryptography (FDTC '07),* pp. 62-72, Sept. 2007.

[8] R. Karri, K. Wu, P. Mishra, and K. Yongkook, "Fault-Based Side-Channel Cryptanalysis Tolerant Rijndael Symmetric Block Cipher Architecture," *Proc. IEEE Int'l Symp. Defect and Fault Tolerance in VLSI Systems (DFT '01),* pp. 418-426, Oct. 2001.

[9] G. Bertoni, L. Breveglieri, I. Koren, P. Maistri, and V. Piuri, "A Parity Code Based Fault Detection for an Implementation of the Advanced Encryption Standard," *Proc. IEEE Int'l Symp. Defect and Fault Tolerance in VLSI Systems (DFT '02),* pp. 51-59, Nov. 2002.

[10] G. Bertoni, L. Breveglieri, I. Koren, P. Maistri, and V. Piuri, "Error Analysis and Detection Procedures for a Hardware Implementation of the Advanced Encryption Standard," *IEEE Trans. Computers,* vol. 52, no. 4, pp. 492-505, Apr. 2003.

[11] R. Karri, G. Kuznetsov, and M. Goessel, "Parity-Based Concurrent Error Detection of Substitution-Permutation Network Block Ciphers," *Proc. Int'l Workshop Cryptographic Hardware and Embedded Systems (CHES '03),* pp. 113-124, Sept. 2003.

[12] M. Karpovsky, K.J. Kulikowski, and A. Taubin, "Differential Fault Analysis Attack Resistant Architectures for the Advanced Encryption Standard," *Proc. Conf. Smart Card Research and Advanced Applications (CARDIS '04),* vol. 153, pp. 177-192, Aug. 2004.

[13] K. Wu, R. Karri, G. Kuznetsov, and M. Goessel, "Low Cost Concurrent Error Detection for the Advanced Encryption Standard," *Proc. Int'l Test Conf.,* pp. 1242-1248, Oct. 2004.

[14] G. Bertoni, L. Breveglieri, I. Koren, and P. Maistri, "An Efficient Hardware-Based Fault Diagnosis Scheme for AES: Performances and Cost," *Proc. IEEE Int'l Symp. Defect and Fault Tolerance in VLSI Systems (DFT '04),* pp. 130-138, Oct. 2004.

[15] L. Breveglieri, I. Koren, and P. Maistri, "Incorporating Error Detection and Online Reconfiguration into a Regular Architecture for the AES," *Proc. IEEE Int'l Symp. Defect and Fault Tolerance in VLSI Systems (DFT '05),* pp. 72-80, Oct. 2005.

[16] C.H. Yen and B.F. Wu, "Simple Error Detection Methods for Hardware Implementation of Advanced Encryption Standard," *IEEE Trans. Computers,* vol. 55, no. 6, pp. 720-731, June 2006.

[17] T.G. Malkin, F.X. Standaert, and M. Yung, "A Comparative Cost/Security Analysis of Fault Attack Countermeasures," *Proc. Int'l Workshop Fault Diagnosis and Tolerance in Cryptography (FDTC '06),* pp. 159-172, Oct. 2006.

[18] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "Fault Detection Structures of the S-boxes and the Inverse S-boxes for the Advanced Encryption Standard," *J. Electronic Testing,* vol. 25, no. 4, pp. 225-245, Aug. 2009.

[19] A. Satoh, T. Sugawara, N. Homma, and T. Aoki, "High-Performance Concurrent Error Detection Scheme for AES Hardware," *Proc. Int'l Workshop Cryptographic Hardware and Embedded Systems (CHES '08),* pp. 100-112, Aug. 2008.

[20] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "A Lightweight Concurrent Fault Detection Scheme for the AES S-boxes Using Normal Basis," *Proc. Int'l Workshop Cryptographic Hardware and Embedded Systems (CHES '08),* pp. 113-129, Aug. 2008.

[21] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "A Lightweight High-Performance Fault Detection Scheme for the Advanced Encryption Standard Using Composite Fields," *IEEE Trans. Very Large Scale Integration Systems,* vol. 19, no. 1, pp. 85-91, Jan. 2011.

[22] G. Di Natale, M. Doulcier, M.L. Flottes, and B. Rouzeyre, "A Reliable Architecture for Parallel Implementations of the Advanced Encryption Standard," *J. Electronic Testing,* vol. 25, no. 4, pp. 269-278, Aug. 2009.

[23] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "Concurrent Structure-Independent Fault Detection Schemes for the Advanced Encryption Standard," *IEEE Trans. Computers,* vol. 59, no. 5, pp. 608-622, May 2010.

[24] P. Maistri and R. Leveugle, "Double-Data-Rate Computation as a Countermeasure against Fault Analysis," *IEEE Trans. Computers,* vol. 57, no. 11, pp. 1528-1539, Nov. 2008.

[25] C. Moratelli, F. Ghellar, E. Cota, and M. Lubaszewski, "A Fault-Tolerant DFA-Resistant AES Core," *Proc. IEEE Int'l Symp. Circuits and Systems (ISCAS '08),* pp. 244-247, May 2008.

[26] S. Morioka and A. Satoh, "An Optimized S-Box Circuit Architecture for Low Power AES Design," *Proc. Int'l Workshop Cryptographic Hardware and Embedded Systems (CHES '02),* pp. 172-186, Aug. 2002.

[27] A. Hodjat and I. Verbauwhede, "Area-Throughput Trade-Offs for Fully Pipelined 30 to 70 Gbits/s AES Processors," *IEEE Trans. Computers,* vol. 55, no. 4, pp. 366-372, Apr. 2006.

[28] V. Rijmen, "Efficient Implementation of the Rijndael S-box," Katholieke Universiteit Leuven, Dept. of ESAT, Belgium, http://www.esat.kuleuven.ac.be/rijmen/rijndael/sbox.pdf, 2000.

[29] A. Rudra, P.K. Dubey, C.S. Jutla, V. Kumar, J.R. Rao, and P. Rohatgi, "Efficient Rijndael Encryption Implementation with Composite Field Arithmetic," *Proc. Int'l Workshop Cryptographic Hardware and Embedded Systems (CHES '01),* pp. 171-184, May 2001.

[30] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A Compact Rijndael Hardware Architecture with S-Box Optimization," *Proc. Seventh Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '01),* pp. 239-254, Dec. 2001.

[31] J. Wolkerstorfer, E. Oswald, and M. Lamberger, "An ASIC Implementation of the AES SBoxes," *Proc. Cryptographers' Track RSA Conf. Topics in Cryptology (CT-RSA '02),* pp. 67-78, Jan. 2002.

[32] X. Zhang and K.K. Parhi, "High-Speed VLSI Architectures for the AES Algorithm," *IEEE Trans. Very Large Scale Integration Systems,* vol. 12, no. 9, pp. 957-967, Sept. 2004.

[33] D. Canright, "A Very Compact S-Box for AES," *Proc. Int'l Workshop Cryptographic Hardware and Embedded Systems (CHES '05),* pp. 441-455, Aug. 2005.

[34] X. Zhang and K.K. Parhi, "On the Optimum Constructions of Composite Field for the AES Algorithm," *IEEE Trans. Circuits and Systems II: Express Briefs,* vol. 53, no. 10, pp. 1153-1157, Oct. 2006.

[35] S. Nikova, V. Rijmen, and M. Schlaffer, "Using Normal Bases for Compact Hardware Implementations of the AES S-Box," *Proc. Security in Comm. Networks,* pp. 236-245, 2008.

[36] G. Bertoni, M. Macchetti, and L. Negri, "Power-Efficient ASIC Synthesis of Cryptographic Sboxes," *Proc. ACM 14th Great Lakes Symp. VLSI (GLSVLSI '04),* pp. 277-281, Apr. 2004.

[37] L. Breveglieri, I. Koren, and P. Maistri, "An Operation-Centered Approach to Fault Detection in Symmetric Cryptography Ciphers," *IEEE Trans. Computers,* vol. 56, no. 5, pp. 534-540, May 2007.

[38] M. Nicolaidis, R.O. Duarte, S. Manich, and J. Figueras, "Fault-Secure Parity Prediction Arithmetic Operators," *IEEE Design and Test of Computers,* vol. 14, no. 2, pp. 60-71, Apr.-June 1997.

[39] N.A. Touba and E.J. McCluskey, "Logic Synthesis of Multilevel Circuits with Concurrent Error Detection," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,* vol. 16, no. 7, pp. 783-789, July 1997.

[40] S. Fenn, M. Goessel, M. Benaissa, and D. Taylor, "On-Line Error Detection for Bit-Serial Multipliers in GF(2^m)," *J. Electronic Testing,* vol. 13, pp. 29-40, 1998.

[41] C. Metra, M. Favalli, and B. Ricco, "Novel Implementation for Highly Testable Parity Code Checkers," *Proc. Int'l Workshop On-Line Testing,* pp. 167-171, 1998.

[42] A. Reyhani-Masoleh and M.A. Hasan, "Fault Detection Architectures for Field Multiplication Using Polynomial Bases," *IEEE Trans. Computers,* vol. 55, no. 9, pp. 1089-1103, Sept. 2006.

[43] G.C. Cardarilli, M. Ottavi, S. Pontarelli, M. Re, and A. Salsano, "Fault Localization, Error Correction, and Graceful Degradation in Radix 2 Signed Digit-Based Adders," *IEEE Trans. Computers,* vol. 55, no. 5, pp. 534-540, May 2006.

[44] M. George and P. Alfke, "Linear Feedback Shift Registers in Virtex Devices," *Xilinx Application Note 210,* http://www.xilinx.com/support/documentation/application_notes/xapp210.pdf, 2010.

[45] ModelSim, http://www.model.com/, 2010.

[46] STMicroelectronics, http://www.st.com/, 2010.

[47] Synopsys, http://www.synopsys.com/, 2010.

Mehran Mozaffari-Kermani received the BSc degree in electrical and computer engineering from the University of Tehran in 2005, and the MESc degree in electrical and computer engineering in 2007 from the University of Western Ontario, where he is currently working toward the PhD degree at the Department of Electrical and Computer Engineering. His current research interests include secure cryptographic systems, fault diagnosis and tolerance, VLSI reliability, and computer arithmetic. He is a student member of the IEEE.

Arash Reyhani-Masoleh received the BSc degree in electrical and electronic engineering from Iran University of Science and Technology in 1989, the MSc degree in electrical and electronic engineering from the University of Tehran in 1991, both with the first rank, and the PhD degree in electrical and computer engineering from the University of Waterloo in 2001. From 1991 to 1997, he was with the Department of Electrical Engineering, Iran University of Science and Technology. From June 2001 to September 2004, he was with the Centre for Applied Cryptographic Research, University of Waterloo, where he was awarded a Natural Sciences and Engineering Research Council of Canada (NSERC) postdoctoral fellowship in 2002. In October 2004, he joined the Department of Electrical and Computer Engineering, University of Western Ontario, London, Canada, where he is currently a tenured associate professor. Currently, he is also serving as an associate editor for Integration, the *VLSI Journal* (Elsevier). His current research interests include algorithms and VLSI architectures for computations in finite fields, fault-tolerant computing, and error-control coding. He has been awarded a NSERC Discovery Accelerator Supplement (DAS) in 2010. He is a member of the IEEE and the IEEE Computer Society.