A Lightweight High-Performance Fault Detection Scheme for the Advanced Encryption Standard Using Composite Fields

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

***Abstract*—The faults that accidently or maliciously occur in the hardware implementations of the Advanced Encryption Standard (AES) may cause erroneous encrypted/decrypted output. The use of appropriate fault detection schemes for the AES makes it robust to internal defects and fault attacks. In this paper, we present a lightweight concurrent fault detection scheme for the AES. In the proposed approach, the composite field S-box and inverse S-box are divided into blocks and the predicted parities of these blocks are obtained. Through exhaustive searches among all available composite fields, we have found the optimum solutions for the least overhead parity-based fault detection structures. Moreover, through our error injection simulations for one S-box (respectively inverse S-box), we show that the total error coverage of almost 100% for 16 S-boxes (respectively inverse S-boxes) can be achieved. Finally, it is shown that both the application-specific integrated circuit and field-programmable gate-array implementations of the fault detection structures using the obtained optimum composite fields, have better hardware and time complexities compared to their counterparts.**

***Index Terms*—AES, composite fields, error coverage, fault detection.**

I. Introduction

THE Advanced Encryption Standard (AES) has been lately accepted by NIST [1] as the symmetric key standard for encryption and decryption of blocks of data. In encryption, the AES accepts a plaintext input, which is limited to 128 bits, and a key that can be specified to be 128 (AES-128), 192 or 256 bits to generate the ciphertext. In the AES-128, which is hereafter referred to as the AES, the ciphertext is generated after 10 rounds, where each encryption round (except for the final round) consists of four transformations. The four transformations in the AES encryption include *SubBytes* (implemented by 16 S-boxes), *ShiftRows*, *MixColumns*, and *AddRoundKey*. Furthermore, to obtain the original plaintext from the ciphertext, the AES decryption algorithm is utilized. The decryption transformations are the reverse of the encryption ones [1]. Among the transformations in the AES, only the S-boxes in the encryption and the inverse S-boxes in the decryption are nonlinear. It is interesting to note that these transformations occupy much of the total AES encryption/decryption area [1]. Therefore, the fault detection schemes for their hardware implementations play an important role in making the standard robust to the internal and malicious faults.

There exist many schemes for detecting the faults in the hardware implementation of the AES, see for example [2]–[15]. Among them, the schemes presented in [2]–[7] are independent of the ways the AES S-box and inverse S-box are implemented in hardware. Moreover, there exist other fault detection schemes that are suitable for a specific implementation of the S-box and the inverse S-box. The approach in [8] and the one in [9] which is extended in [10] are based on using memories (ROMs) for the S-box and the inverse S-box. Moreover, a fault tolerant scheme which is resistant to fault attacks is presented in [11]. To protect the combinational logic blocks used in the four transformations of the AES, either the parity-based scheme proposed in [10] or the duplication approach is implemented. Furthermore, to protect the memories used for storing the expanded key and the state matrix, either the Hamming or Reed–Solomon error correcting code is utilized. It is noted that our proposed fault detection approach is only applied to the composite field S-box and inverse S-box. Whereas, the scheme presented in [10] uses memories. Using ROMs may not be preferable for high performance AES implementations. Therefore, for applications requiring high performance, the S-box and the inverse S-box are implemented using logic gates in composite fields [16].

The schemes in [12]–[15] are suitable for the composite field implementation of the S-box and the inverse S-box. The approach in [12] is based on using the parity-based fault detection method for a specific S-box using composite field and polynomial basis for covering all the single faults. In the scheme of [13], the fault detection of the multiplicative inversion of the S-box is considered for two specific composite fields. The transformation and affine matrices are excluded in this approach. Moreover, in [14], predicted parities have been used for the multiplicative inversion of a specific S-box using composite field and polynomial basis. Furthermore, the transformation matrices are also considered. Finally, in the parity-based approach in [15], through exhaustive search among all the fault detection S-boxes utilizing five predicted parities using normal basis, the most compact one is obtained.

The contributions of this paper are as follows.

- We have presented a low-cost parity-based fault detection scheme for the S-box and the inverse S-box using composite fields. In the presented approach, for increasing the error coverage, the predicted parities of the five blocks of

Manuscript received April 06, 2009; revised July 24, 2009. First published November 10, 2009; current version published December 27, 2010. This work is supported in part by an NSERC Discovery grant awarded to A. Reyhani-Masoleh.

The authors are with the Department of Electrical and Computer Engineering, The University of Western Ontario, London, ON N6A 5B9, Canada (e-mail: mmozaff@uwo.ca; areyhani@uwo.ca).

Digital Object Identifier 10.1109/TVLSI.2009.2031651

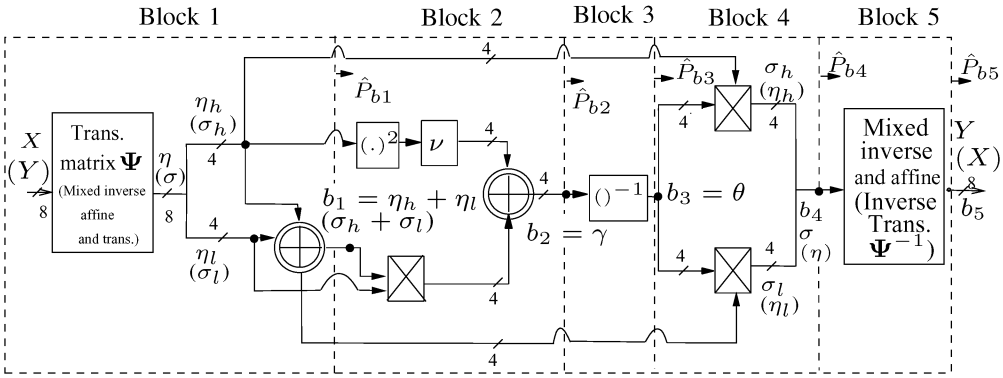

Fig. 1. The S-box (the inverse S-box) using composite fields and polynomial basis [17] and their fault detection blocks.

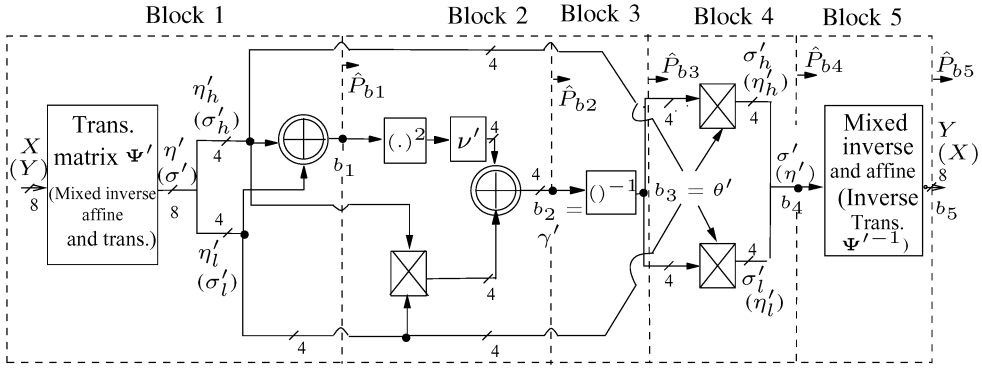

Fig. 2. The S-box (the inverse S-box) using composite fields and normal basis [16] and their fault detection blocks.

the S-box and the inverse S-box are obtained (three predicted parities for the multiplicative inversion and two for the transformation and affine matrices). It is interesting to note that the cost of our multi-bit parity prediction approach is lower than its counterparts which use single-bit parity. It also has higher error coverage than the approaches using single-bit parities. We have implemented both the proposed fault detection S-box and inverse S-box and other counterparts. Our both ASIC and FPGA implementation results show that compared to the approaches presented in [13] and [14], the complexities of the proposed fault detection scheme are lower.

- Through exhaustive searches, we obtain the least area and delay overhead fault detection structures for the optimum composite fields using both polynomial basis and normal basis. While in [15], only the S-box using normal basis has been considered.
- The proposed fault detection scheme is simulated and we show that the error coverages of close to 100% for 16 S-boxes (respectively inverse S-boxes) can be obtained.
- Finally, we have implemented the fault detection hardware structures of the AES using both 0.18-μm CMOS technology and on Xilinx Virtex-II Pro FPGA. It is shown that the fault detection scheme using the optimum polynomial and normal bases have lower complexities than those using other composite fields for both with and without fault detection capability.

II. Preliminaries

In this section, we describe the S-box and the inverse S-box operations and their composite-field realizations.

The S-box and the inverse S-box are nonlinear operations which take 8-bit inputs and generate 8-bit outputs. In the S-box, the irreducible polynomial of $P(x) = x^8+x^4+x^3+x+1$ is used to construct the binary field $GF(2^8)$. Let $X = \sum_{i=0}^{7} x_i\alpha^i \in GF(2^8)$ and $Y = \sum_{i=0}^{7} y_i\alpha^i \in GF(2^8)$ be the input and the output of the S-box, respectively, where α is a root of $P(x)$, i.e., $P(\alpha) = 0$. Then, the S-box consists of the multiplicative inversion, i.e., $X^{-1} \in GF(2^8)$, followed by an affine transformation [1]. Moreover, let $Y \in GF(2^8)$ and $X \in GF(2^8)$ be the input and the output of the inverse S-box, respectively. Then, the inverse S-box consists of an inverse affine transformation followed by the multiplicative inversion.

The composite fields can be represented using normal basis [16] or polynomial basis [17]–[20]. The S-box and inverse S-box for the polynomial and normal bases are shown in Figs. 1 and 2, respectively. As shown in these figures, for the S-box using polynomial basis (respectively normal basis), the transformation matrix $\mathbf{\Psi}$ (respectively $\mathbf{\Psi'}$[1]) transforms a field element X in the binary field $GF(2^8)$ to the corresponding representation in the composite fields $GF(2^8)/GF(2^4)$. It is noted that the result of $X = \eta_h u + \eta_l$ in Fig. 1 (respectively $X = \eta'_h u^{16} + \eta'_l u$ in Fig. 2) is obtained using the irreducible polynomial of $u^2 + \tau u + \nu$ (respectively $u^2 + \tau' u + \nu'$).

[1]We use prime notations for the composite fields using normal basis.

The multiplicative inversion in Fig. 1 consists of composite-field multiplications, additions and an inversion in the sub-field $GF(2^4)$ over $GF(2)/x^4+x+1$ [18]. The decomposition can be further applied to represent $GF(2^4)$ as a linear polynomial over $GF(2^2)$ and then $GF(2)$ using the irreducible polynomials of $v^2+\Omega v+\Phi$ and w^2+w+1, respectively. As a result, it is understood that the implementation of the multiplicative inversion can be performed using the field represented by $GF((2^4)^2)$, see for example [18] and [19], or the field represented by $GF(((2^2)^2)^2)$ and has been used in the literature, see for example [17] and [20]. Finally, as seen in Fig. 2 for normal basis, the decomposition is performed using the irreducible polynomials of $v^2+\Omega' v+\Phi'$ and w^2+w+1.

For calculating the multiplicative inversion, the most efficient choice is to let $\Omega=\tau=1$ ($\Omega'=\tau'=1$) in the above irreducible polynomials [16]. Then, we have the following for the multiplicative inversion of the S-box using polynomial basis (Fig. 1) and normal basis (Fig. 2), respectively, [16], [17]

$$\begin{aligned}(\eta_h u+\eta_l)^{-1}=&\left[\left((\eta_h+\eta_l)\eta_l+\eta_h^2\nu\right)^{-1}\eta_h\right]u\\&+\left((\eta_h+\eta_l)\eta_l+\eta_h^2\nu\right)^{-1}(\eta_h+\eta_l)\end{aligned}\quad(1)$$

$$\begin{aligned}\left(\eta_h' u^{16}+\eta_l' u\right)^{-1}=&\left[\left(\eta_h'\eta_l'+\left({\eta_h'}^2+{\eta_l'}^2\right)\nu'\right)^{-1}\eta_h'\right]u^{16}\\&+\left[\left(\eta_h'\eta_l'+\left({\eta_h'}^2+{\eta_l'}^2\right)\nu'\right)^{-1}\eta_l'\right]u.\end{aligned}\quad(2)$$

It is noted that one can replace $\eta(\eta')$ with $\sigma(\sigma')$ to obtain (1) and (2) for the inverse S-box. In the next section, we propose the low-cost fault detection scheme for the S-box and the inverse S-box.

III. Fault Detection Scheme

To obtain low-overhead parity prediction, we have divided the S-box and the inverse S-box into five blocks as shown in Figs. 1 and 2. In these figures, the modulo-2 additions, consisting of 4 XOR gates, are shown by two concentric circles with a plus inside. Furthermore, the multiplications in $GF(2^4)$ are shown by rectangles with crosses inside. Let b_i be the output of the block i denoted by dots in Figs. 1 and 2 for the S-box. As seen in Fig. 1, $b_1=\eta_h+\eta_l$, $b_2=\gamma$, $b_3=\theta$, $b_4=\sigma$, and $b_5=Y$. Similarly, from Fig. 2, $b_1=\eta_h'+\eta_l'$, $b_2=\gamma'$, $b_3=\theta'$, $b_4=\sigma'$, and $b_5=Y$. One can replace $\eta(\eta')$ with $\sigma(\sigma')$ and X with Y for the inverse S-box. In the following, we have exhaustively searched for the least overhead parity predictions of these blocks denoted by $\hat{P}_{b1}-\hat{P}_{b5}$ in Figs. 1 and 2.

A. The S-Box and the Inverse S-Box Using Polynomial Basis

The implementation complexities of different blocks of the S-box and the inverse S-box and those for their predicted parities are dependent on the choice of the coefficients $\nu\in GF(2^4)$ and $\Phi\in GF(2^2)$ in the irreducible polynomials $u^2+u+\nu$ and $v^2+v+\Phi$ used for the composite fields. Our goal in the following is to find $\nu\in GF(2^4)$ and $\Phi\in GF(2^2)$ for the composite fields $GF(((2^2)^2)^2)$ and $\nu\in GF(2^4)$ for the composite fields $GF((2^4)^2)$ so that the area complexity of the entire fault detection implementations becomes optimum. According to [21], 16 the possible combinations for $\nu\in GF(2^4)$ and $\Phi\in GF(2^2)$ exist. Moreover, for the composite fields $GF((2^4)^2)$, we have exhaustively searched and have found the possible choices for ν making the polynomial $x^2+x+\nu$ irreducible. These parameters determine the complexities of some blocks as explained next.

Blocks 1 and 5: Based on the possible values of ν and Φ in $GF(((2^2)^2)^2)$ (ν in $GF((2^4)^2)$), the transformation matrices in Fig. 1 in blocks 1 and 5 of the S-box and the inverse S-box can be constructed using the algorithm presented in [21]. Using an exhaustive search, eight base elements in $GF(((2^2)^2)^2)$ (or $GF((2^4)^2)$) to which eight base elements of $GF(2^8)$ are mapped, are found to construct the transformation matrix.

In [22], the Hamming weights, i.e., the number of nonzero elements, of the transformation matrices for the case $\Phi=\{10\}_2$ and different values of ν in $GF(((2^2)^2)^2)$ are obtained. It is noted that in [21], instead of considering the Hamming weights, subexpression sharing is suggested for obtaining the low-complexity implementations for the S-box. Here, we have also considered these transformation matrices for $\Phi=\{11\}_2$ as well as the transformation matrices for the inverse S-box for different values of ν and Φ and have derived their area and delay complexities. Moreover, the gate count and the critical path delay for blocks 1 and 5 and their predicted parities, i.e., $\hat{P}_{b1}$ and $\hat{P}_{b5}$, of the S-box and the inverse S-box in $GF((2^4)^2)$ have been obtained.

Blocks 2 and 4: As shown in Fig. 1, block 2 of the S-box and the inverse S-box consists of a multiplication, an addition, a squaring and a multiplication by constant ν in $GF((2^2)^2)$. We present the following lemma for deriving the predicted parity of the multiplication in $GF((2^2)^2)$, using which the predicted parities of blocks 2 and 4 in Fig. 1 are obtained.

Lemma 1: Let $\lambda=(\lambda_3,\lambda_2,\lambda_1,\lambda_0)$ and $\delta=(\delta_3,\delta_2,\delta_1,\delta_0)$ be the inputs of a multiplier in $GF((2^2)^2)$. The predicted parities of the result of the multiplication of λ and δ in $GF((2^2)^2)$ for $\Phi=\{10\}_2$ and $\Phi=\{11\}_2$ are as follows, respectively

$$\begin{aligned}\hat{P}_\pi=&\lambda_3(\delta_3+\delta_2+\delta_0)+\lambda_2(\delta_3+\delta_1+\delta_0)+\lambda_1(\delta_2+\delta_0)\\&+\lambda_0(\delta_3+\delta_2+\delta_1+\delta_0)\quad\text{if}\quad\Phi=\{10\}_2\end{aligned}\quad(3)$$

$$\begin{aligned}\hat{P}_\pi=&\lambda_3(\delta_3+\delta_0)+\lambda_2(\delta_2+\delta_1+\delta_0)+\lambda_1(\delta_2+\delta_0)\\&+\lambda_0(\delta_3+\delta_2+\delta_1+\delta_0)\quad\text{if}\quad\Phi=\{11\}_2.\end{aligned}\quad(4)$$

Proof: One can perform modulo-2 addition of the coordinates of the result of the multiplication over $GF((2^2)^2)$ [17]. Then, by reordering and factoring of the result for $\Phi=\{10\}_2$ and $\Phi=\{11\}_2$, the predicted parities in (3) and (4) are obtained. ■

The predicted parity of block 2 of the S-box and the inverse S-box, i.e., $\hat{P}_{b2}=\hat{P}_{\eta_h{}^2\nu}+\hat{P}_{(\eta_h+\eta_l)\eta_l}$ in Fig. 1, depends on the choice of the coefficients $\nu\in GF((2^2)^2)$ and $\Phi\in GF(2^2)$. Using Lemma 1, we have derived the complexity of the predicted parity of block 2 for these coefficients. Furthermore, for block 4 in Fig. 1, which consists of two multiplications in $GF((2^2)^2)$, one can also use Lemma 1 to derive the predicted

parity. For block 2 of the S-box (respectively the inverse S-box) over $GF((2^4)^2)$ in Fig. 1, only the multiplication by constant ν is affected for different values of νs. For this block, we have exhaustively searched for and obtained the optimum implementation for different values of νs. Moreover, block 4 in Fig. 1 is independent of the value of ν. Therefore, the complexity of the predicted parity for this block is the same for all possible νs.

Block 3: We present the following theorem for block 3 of the S-box and the inverse S-box over $GF((2^2)^2)$ in Fig. 1.

Theorem 1: Let $\gamma = (\gamma_3, \gamma_2, \gamma_1, \gamma_0)$ be the input and $\theta = (\theta_3, \theta_2, \theta_1, \theta_0)$ be the output of an inverter in $GF((2^2)^2)$. The predicted parities of the result of the inversion in $GF((2^2)^2)$, i.e., $\hat{P}_{b3}$, for $\Phi = \{10\}_2$ and $\Phi = \{11\}_2$ are as follows, respectively

$$\hat{P}_\theta = (\overline{\gamma}_2 \vee \gamma_1)\gamma_0 + (\gamma_1 + \gamma_0)\gamma_3 \quad if \quad \Phi = \{10\}_2 \quad (5)$$

$$\hat{P}_\theta = (\gamma_2\gamma_1 \vee \gamma_0) + \gamma_3\gamma_1 \quad if \quad \Phi = \{11\}_2, \quad (6)$$

where, $\vee$ represents an OR operation.

Proof: By Modulo-2 addition of the coordinates of the result of the inversion in $GF((2^2)^2)$ for $\Phi = \{10\}_2$ in [17], one can obtain the predicted parity of θ as $\hat{P}_\theta = \gamma_2\gamma_0 + \gamma_2\gamma_1\gamma_0 + \gamma_3\gamma_1 + \gamma_0 + \gamma_3\gamma_0 = \gamma_0(\gamma_2(\gamma_1 + 1) + 1) + \gamma_3(\gamma_1 + \gamma_0)$. By noting that $\gamma_1 + 1 = \overline{\gamma}_1$ and $\overline{\gamma_2\overline{\gamma}_1} = \overline{\gamma}_2 \vee \gamma_1$, one can reach (5). Moreover, by XORing the result for $\Phi = \{11\}_2$, $\hat{P}_\theta$ is obtained as $\hat{P}_\theta = \gamma_3\gamma_1 + \gamma_2\gamma_1\gamma_0 + \gamma_2\gamma_1 + \gamma_0$. Noting that $\gamma_2\gamma_1\gamma_0 + \gamma_2\gamma_1 + \gamma_0 = \gamma_2\gamma_1 \vee \gamma_0$, one can simplify $\hat{P}_\theta$ to reach (6) and the proof is complete. ∎

It is noted that the inversion in $GF(2^4)$ in Fig. 1 is independent of the value of ν. Therefore, the complexity of the predicted parity for this block is the same for any possible νs.

Considering the discussions presented in this section for different combinations of ν and Φ for polynomial basis, we present the following for the optimum parity predictions.

Proposition 1: The fault detection S-box using composite fields $GF(((2^2)^2)^2)$ has the least area complexity for $\Phi = \{11\}_2$ and $\nu = \{1010\}_2$. For this optimum S-box (PB_1), we have the following predicted parities for the five blocks in Fig. 1: $\hat{P}_{b1} = x_0$, $\hat{P}_{b2} = \eta_3(\overline{\eta_7} + \eta_4) + \eta_2(\overline{\eta_7} + P_{\eta_h}) + \eta_1(\eta_6 + \eta_4) + \eta_0\overline{P_{\eta_h}} + \eta_6 + \eta_7$, $\hat{P}_{b3} = (\gamma_2\gamma_1 \vee \gamma_0) + \gamma_1\gamma_3$, $\hat{P}_{b4} = \eta_3(\theta_3 + \theta_0) + \eta_2(P_\Theta + \theta_3) + \eta_1(\theta_2 + \theta_0) + \eta_0 P_\Theta$, $\hat{P}_{b5} = \sigma_7 + \sigma_5 + \sigma_3 + \sigma_2 + \sigma_0$, where, $P_{\eta_h} = \eta_7 + \eta_6 + \eta_5 + \eta_4$ and $P_\Theta = \theta_3 + \theta_2 + \theta_1 + \theta_0$. Additionally, among all the possible values for ν using composite fields $GF((2^4)^2)$, $\nu = \{1010\}_2$ yields to the least-complexity architecture for the optimum S-box (PB_2), respectively. Then, for the S-box we have: $\hat{P}_{b1} = x_7 + x_0$, $\hat{P}_{b2} = \eta_3\eta_4 + \eta_2(\eta_5 + \eta_4) + \eta_1(\overline{P_{\eta_h}} + \eta_7) + \eta_0\overline{P_{\eta_h}} + P_{\eta_h} + \eta_4$, $\hat{P}_{b3} = \overline{\gamma_3}\gamma_2\overline{\gamma_0} + \gamma_0(\overline{\gamma_1} \vee \overline{(\gamma_2 + \gamma_3)})$, $\hat{P}_{b4} = \eta_3\theta_0 + \eta_2(\theta_1 + \theta_0) + \eta_1(P_\Theta + \theta_3) + \eta_0 P_\Theta$, $\hat{P}_{b5} = \sigma_4 + \sigma_3 + \sigma_2 + \sigma_1 + \sigma_0$.

Furthermore, we have the following for the inverse S-box.

Proposition 2: For the inverse S-box using composite field $GF(((2^2)^2)^2)$, choosing $\Phi = \{11\}_2$ and $\nu = \{1011\}_2$ and for the one using composite field $GF((2^4)^2)$ having $\nu = \{1001\}_2$ yields to the lowest area complexity architecture. It is noted that blocks 3 and 4 have the same predicted parities as the S-box by swapping η and σ. For other blocks of the optimum inverse S-box (PB_1) we have: $\hat{P}_{b1} = y_7 + y_6 + y_5 + y_2$, $\hat{P}_{b2} = \sigma_3(\overline{\sigma_7} + \sigma_4) + \sigma_2(\overline{\sigma_7} + P_{\sigma_h}) + \sigma_1(\sigma_6 + \sigma_4) + \sigma_0\overline{P_{\sigma_h}} + \sigma_4$, $\hat{P}_{b5} = \eta_7 + \eta_6 + \eta_3 + \eta_2 + \eta_0$. Additionally, for the optimum inverse S-box (PB_2) we have: $\hat{P}_{b1} = \overline{y_7 + y_6 + y_3}$, $\hat{P}_{b2} = \sigma_3\sigma_4 + \sigma_2(\sigma_5 + \sigma_4) + \sigma_1(\overline{P_{\sigma_h}} + \sigma_7) + \sigma_0\overline{P_{\sigma_h}} + \sigma_7$, $\hat{P}_{b5} = \eta_0$.

B. The S-Box and the Inverse S-Box Using Normal Basis

In [15], the optimum fault detection S-box using normal basis in Fig. 2 is derived. In this paper, we have also performed an exhaustive search for finding the optimum predicted parities based on the choice of the coefficients $\nu' \in GF(2^4)$ and $\Phi' \in GF(2^2)$ for the five blocks of the inverse S-box using normal basis. We have exhaustively searched for the least overhead transformation matrices and their parity predictions combined for the inverse S-box and have derived the total complexity for the predicted parities of blocks 1 and 5, i.e., $\hat{P}_{b1}$ and $\hat{P}_{b5}$, and the delays associated with them. These are used to obtain the optimum S-box inverse S-box and its parity predictions in this section. It is also noted that as shown in Fig. 2, blocks 2, 3, and 4 of the S-box and the inverse S-box are the same. Therefore, considering [15], the predicted parities of these blocks can be obtained for the inverse S-box. Using the discussions presented in this section, we present the following for the optimum parity predictions.

Proposition 3: For different combinations of ν' and Φ' for normal basis, for the S-box and the inverse S-box, $\Phi' = \{10\}_2$ and $\nu' = \{0001\}_2$ have the least area for the operations and their fault detection circuits combined. The following is the predicted parities for the S-box: $\hat{P}_{b1} = x_7 + x_5$, $\hat{P}_{b2} = (\eta'_7 \vee \eta'_3) + (\eta'_6 \vee \eta'_2) + (\eta'_4 \vee \eta'_0) + \eta'_5\eta'_1$, $\hat{P}_{b3} = \gamma'_2\gamma'_0(\gamma'_3 + \gamma'_1) + \gamma'_3\gamma'_1(\gamma'_2 + \gamma'_0)$, $\hat{P}_{b4} = (\eta'_7 + \eta'_3)\theta'_3 + (\eta'_6 + \eta'_2)\theta'_2 + (\eta'_5 + \eta'_1)\theta'_1 + (\eta'_4 + \eta'_0)\theta'_0$, $\hat{P}_{b5} = \sigma'_7 + \sigma'_5 + \sigma'_4 + \sigma'_3 + \sigma'_2$. Moreover, for the inverse S-box, $\hat{P}_{b2} - \hat{P}_{b4}$ are the same as those for the S-box by swapping η' and σ'. For the other blocks, we have: $\hat{P}_{b1} = y_7 + y_6 + y_2 + y_1$ and $\hat{P}_{b5} = \eta'_7 + \eta'_5 + \eta'_4 + \eta'_3 + \eta'_2$.

It is noted that the area overhead of the proposed scheme for the optimum structures consists of those of the optimum parity predictions. In addition, 23 XORs for the actual parities (3 XORs for adding the coordinates of each of $\eta'_h + \eta'_l$, γ', and θ' and 7 XORs each for those of σ' and Y) are utilized. Moreover, the delay overhead of the predicted parities of five blocks can overlap the delays for the implementations of five blocks in Figs. 1 and 2. The only delay overhead for this scheme is the delay of the actual parity of block 5, which is $3T_X$, where, T_X is the delay of an XOR gate.

IV. Error Simulations

If exactly one biterror appears at the output of the S-box (respectively inverse S-box), the presented fault detection scheme is able to detect it and the error coverage is about 100%. This is because in this case, the error indication flag of the corresponding block alarms the error. However, due to the technological constraints, single stuck-at error may not be applicable for an attacker to gain more information [23]. Thus, multiple bits will actually be flipped and hence multiple stuck-at errors are also considered in this paper covering both natural faults and fault attacks [23].

For the calculation of the error coverage for the multiple errors, we define p_i as the probability of error detection in block

TABLE I
ERROR SIMULATION RESULTS OF THE OPTIMUM S-BOX AND INVERSE S-BOX AFTER INJECTING 500 000 ERRORS

Operations	Field	Errors covered	Error Coverage
S-box (Inverse S-box)	PB_1	485,008 (485,106)	97.002% (97.021%)
	PB_2	485,039 (485,015)	97.008% (97.002%)
	NB	485,015 (485,174)	97.003% (97.035%)

i, $1 \leq i \leq 5$, in Figs. 1 and 2. Then, the probability of not detecting the errors in block i is $(1-p_i)$. For randomly distributed errors in the S-box (respectively inverse S-box), this probability for each block is independent of those of other blocks. Therefore, one can derive the equation for the error coverage of the randomly distributed errors as

$$EC\% = 100 \times \left(1 - \prod_{i \in \mathbb{S}} (1 - p_i)\right)\% \tag{7}$$

where $\mathbb{S}$ is the set of the block numbers where the faults are injected. For randomly distributed errors, the error coverage for each block is $p_i \approx 1/2$. Then, the representation of (7) can be simplified as $EC\% = 100 \times (1 - (1/2)^n)\%$, where, n is the number of blocks. Therefore, if multiple errors are randomly distributed in all blocks, the error coverage reaches 97% using $n = 5$ error indication flags.

We have performed error simulations for the S-boxes and the inverse S-boxes using the optimum composite field obtained in the previous section to confirm our above theoretical computation. In our simulations, we use stuck-at error model at the outputs of the five blocks forcing one or multiple nodes 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 LFSRs for injecting random multiple errors, where, the numbers, the locations and the types of the errors are randomly chosen. In this regard, the maximum sequence length polynomial for the feedback is selected. The injected errors are transient, i.e., they last for one clock cycle. However, the results would be the same if permanent errors are considered.

The results of the error simulations using Xilinx ISE version 9.1i Simulator (ISim)[2] are presented in Table I. As seen in this table, up to 500 000 random errors are injected for both the S-box and the inverse S-box. It is noted that in these tables, the optimum polynomial basis $GF(((2^2)^2)^2)$ denoted by PB_1, $GF((2^4)^2)$ denoted by PB_2 and normal basis (NB) are presented. As shown in the table, using five parity bits of the five blocks, the error coverage for random faults reaches 97% which is the same as our theoretical computation in this section. This error coverage will be increased if the outputs of more than one S-box (respectively inverse S-box) of the AES implementation are erroneous. In this case, the errors detected in any of 16 S-boxes (respectively inverse S-boxes) contribute to the total error coverage. Thus, error coverage of very close to 100% is achieved.

[2]Xilinx [Online]. Available: http://www.xilinx.com/

V. ASIC AND FPGA IMPLEMENTATIONS AND COMPARISONS

In this section, we compare the areas and the delays of the presented scheme with those of the previously reported ones in both both application-specific integrated circuit (ASIC) and field-programmable gate array (FPGA) implementations. We have implemented the S-boxes using memories and the ones presented in [20] (the hardware optimization of [17]), [18], and [22] which use polynomial basis representation in composite fields. We have also implemented the fault detection schemes proposed in [2], [8] (both united and parity-based), and [10] which are based on the ROM-based implementation of the S-box. The results of the implementations for both original and fault detection scheme (FDS) in terms of delay and area have been tabulated in Tables II and III. As seen in these tables, the original structures are not divided into blocks and full optimization of the original entire architecture as a single block is performed in both ASIC and FPGA. This allows us to find the actual overhead of the presented fault detection scheme as compared to the original structures which are not divided into five blocks. We have used 0.18-μm CMOS technology for the ASIC implementations. These architectures have been coded in VHDL as the design entry to the Synopsys Design Analyzer. The results are tabulated in Table II. Moreover, for the FPGA implementations in Table III, the Xilinx Virtex-II Pro FPGA (xc2vp2-7) [24] is utilized in the Xilinx ISE version 9.1i. Furthermore, the synthesis is performed using the XST.

As seen in Tables II and III, we have implemented the fault detection scheme presented in [2] and [8] based on using redundant units for the S-box (united S-box). Furthermore, the fault detection scheme proposed in [10] is implemented. This scheme uses 512×9 memory cells to generate the predicted parity bit and the 8-bit output of the S-box [10]. One can obtain from Tables II and III that for both of these schemes, the area overhead is more than 100%. As mentioned in the introduction, the approach in [11] utilizes the scheme in [10] for protecting the combinational logic elements, whose implementation results are also shown in Tables II and III. Additionally, for certain AES implementations containing storage elements, one can use the error correcting code-based approach presented in [11] in addition to the proposed scheme in this paper to make a more reliable AES implementation. Moreover, the parity-based scheme in [8] which only realizes the multiplicative inversion using memories is implemented. As seen in these tables, we have also implemented the schemes in [13] and [14]. It is noted that the scheme in [13] is for the multiplicative inversion and does not present the parity predictions for the transformation matrices. Moreover, we have applied the presented fault detection scheme to the S-boxes in [18] and [22]. As seen in bold faces in Tables II and III, with the error coverage of close to 100%, the presented low-complexity fault detection S-boxes (presented in Section III) are the most compact ones among the other S-boxes. The optimum S-box and inverse S-box using normal basis have the least hardware complexity with the fault detection scheme. Moreover, as seen in the tables, the optimum structures using composite fields and polynomial basis (PB_1 and PB_2) have the least post place and route timing overhead among other schemes. It is noted that using sub-pipelining for the presented fault detection scheme in this paper, one can reach much more faster hardware implementations of the composite field fault detection structures.

TABLE II
ASIC IMPLEMENTATIONS OF THE FAULT DETECTION SCHEMES FOR THE S-BOX (SB) AND THE INVERSE S-BOX USING 0.18-μm CMOS TECHNOLOGY

Operation	Architecture		Area (μm^2) / Delay (ns)	
	Structure	FDS	Original	FDS
S-box	ROM SB	United S-box [2], [8]	169×10^3 / 5.4	344×10^3 / 7.7
	ROM SB	Two 256×9 ROMs [10]	169×10^3 / 5.4	378×10^3 / 5.8
	ROM (mult. inv.)	Parity-based SB [8]	185×10^3 / 5.8	191×10^3 / 5.9
	PB [20]	[13] (mult. inv.)	5315 / 12.0	6869 / 12.8
	PB [20]	[14]	5315 / 12.0	7047 / 14.1
	PB [20]	[12] for the original SB	5315 / 12.0	6763 / 14.1
	PB [18]	Proposed scheme applied	5642 / 11.3	7113 / 13.0
	PB [22]	Proposed scheme applied	5547 / 13.2	7034 / 13.8
	NB	[15]	5179 / 12.9	6712 / 14.7
	PB_1	**This work**	**5217 / 10.6**	**6723 / 12.5**
	PB_2	**This work**	**5290 / 9.2**	**6739 / 11.5**
Inverse S-box	**NB**	**This work**	**5187 / 13.2**	**6480 / 14.5**
	PB_1	**This work**	**5225 / 10.9**	**6537 / 13.0**
	PB_2	**This work**	**5274 / 9.4**	**6619 / 11.3**

TABLE III
XILINX VIRTEX-II PRO FPGA IMPLEMENTATIONS (xc2vp2-7) OF THE FAULT DETECTION SCHEMES FOR THE S-BOX (SB) AND THE INVERSE S-BOX

Operation	Architecture		Slice / Delay (ns)	
	Structure	FDS	Original	FDS
S-box	ROM (SB)	United SB [2], [8]	69 / 3.826	150 / 5.398
	ROM (SB)	Two 256×9 ROMs [10]	69 / 3.826	159 / 4.287
	ROM (mult. inv.)	Parity-based SB [8]	88 / 5.734	100 / 6.370
	PB [20]	[13] (mult. inv.)	33 / 9.375	44 / 9.869
	PB [20]	[14]	33 / 9.375	47 / 9.996
	PB [20]	[12] for the original SB	33 / 9.375	42 / 10.317
	PB [18]	Proposed scheme applied	38 / 8.285	50 / 9.582
	PB [22]	Proposed scheme applied	37 / 9.986	47 / 10.832
	NB	[15]	31 / 9.339	39 / 10.026
	PB_1	**This work**	**31 / 7.284**	**40 / 7.465**
	PB_2	**This work**	**32 / 7.356**	**41 / 8.150**
Inverse S-box	**NB**	**This work**	**31 / 7.736**	**38 / 7.964**
	PB_1	**This work**	**32 / 6.992**	**42 / 7.423**
	PB_2	**This work**	**32 / 7.550**	**44 / 8.181**

TABLE IV
ASIC IMPLEMENTATIONS OF THE FAULT DETECTION SCHEMES OF THE AES ENCRYPTION USING 0.18-μm MOS TECHNOLOGY

AES encryption	Optimum S-box	Area (μm^2)		Freq. (MHz)
		S-boxes	All	
Original without fault detection	PB_1	692781 (80%)	859471	79.4
	PB_2	704490 (80%)	871180	91.8
	NB	680590 (80%)	845426	73.5
Presented scheme for SubBytes (ShiftRows)	PB_1	956233	-	78.8
	PB_2	972217	-	89.2
	NB	946476	-	69.5
Presented scheme for SubBytes (ShiftRows) scheme in [10] for others	PB_1	-	1268520	68.2
	PB_2	-	1280412	70.1
	NB	-	1256812	60.3

We have also implemented the AES encryption using the presented optimum S-boxes excluding the key expansion. Then, we have added the proposed scheme for SubBytes and ShiftRows considering that ShiftRows is the rewiring from the output of SubBytes. The results are presented in Tables IV and V. As one can notice, the S-boxes occupy more than three fourths of the AES encryption. As shown in these tables, the most compact AES encryption with and without the fault detection scheme is for normal basis. Furthermore, the frequency degradation is negligible. Moreover, the original AES encryption for PB_2 and the ones with fault detection for PB_1 and PB_2 have the highest working frequencies. In addition, as seen in the tables, we have applied the presented scheme to SubBytes and ShiftRows and used the scheme in [10] for the other transformations.

TABLE V
XILINX VIRTEX-II PRO FPGA IMPLEMENTATIONS OF THE FAULT DETECTION SCHEMES OF THE AES ENCRYPTION

AES encryption	Optimum S-box	Slice		Freq. (MHz)
		S-boxes	All	
Original without fault detection	PB_1	5248 (77%)	6760	81.1
	PB_2	5417 (78%)	6913	89.8
	NB	5112 (78%)	6579	75.8
Presented scheme for SubBytes (ShiftRows)	PB_1	6896	-	79.3
	PB_2	6958	-	84.0
	NB	6342	-	73.2
Presented scheme for SubBytes (ShiftRows) scheme in [10] for others	PB_1	-	9881	65.8
	PB_2	-	9921	64.8
	NB	-	9405	60.8

VI. CONCLUSION

In this paper, we have presented a high performance parity-based concurrent fault detection scheme for the AES using the S-box and the inverse S-box in composite fields. Using exhaustive searches, we have found the least complexity S-boxes and inverse S-boxes as well as their fault detection circuits. Our error simulation results show that very high error coverages for the presented scheme are obtained. Moreover, a number of fault detection schemes from the literature have been implemented on ASIC and FPGA and compared with the ones presented here. Our implementations show that the optimum S-boxes and the inverse S-boxes using normal basis are more compact than the ones using polynomial basis. However, the ones using polynomial basis result in the fastest implementations. We have also implemented the AES encryption using the proposed fault detection scheme. The results of the ASIC and FPGA mapping show that the costs of the presented scheme are reasonable with acceptable post place and route delays.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their comments.

REFERENCES

[1] National Institute of Standards and Technologies, Announcing the Advanced Encryption Standard (AES) FIPS 197, Nov. 2001.
[2] R. Karri, K. Wu, P. Mishra, and K. Yongkook, "Fault-based side-channel cryptanalysis tolerant Rijndael symmetric block cipher architecture," in *Proc. DFT*, Oct. 2001, pp. 418–426.
[3] R. Karri, K. Wu, P. Mishra, and Y. Kim, "Concurrent error detection schemes for fault-based side-channel cryptanalysis of symmetric block ciphers," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 21, no. 12, pp. 1509–1517, Dec. 2002.
[4] A. Satoh, T. Sugawara, N. Homma, and T. Aoki, "High-performance concurrent error detection scheme for AES hardware," in *Proc. CHES*, Aug. 2008, pp. 100–112.
[5] L. Breveglieri, I. Koren, and P. Maistri, "Incorporating error detection and online reconfiguration into a regular architecture for the advanced encryption standard," in *Proc. DFT*, Oct. 2005, pp. 72–80.
[6] M. Karpovsky, K. J. Kulikowski, and A. Taubin, "Differential fault analysis attack resistant architectures for the advanced encryption standard," in *Proc. CARDIS*, Aug. 2004, vol. 153, pp. 177–192.
[7] 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.
[8] 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, Jun. 2006.
[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," in *Proc. DFT*, Nov. 2002, pp. 51–59.
[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] C. Moratelli, F. Ghellar, E. Cota, and M. Lubaszewski, "A fault-tolerant DFA-resistant AES core," in *Proc. ISCAS*, 2008, pp. 244–247.
[12] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "Parity-based fault detection architecture of S-box for advanced encryption standard," in *Proc. DFT*, Oct. 2006, pp. 572–580.
[13] S.-Y. Wu and H.-T. Yen, "On the S-box architectures with concurrent error detection for the advanced encryption standard," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E89-A, no. 10, pp. 2583–2588, Oct. 2006.
[14] A. E. Cohen, "Architectures for Cryptography Accelerators," Ph.D. dissertation, Univ. Minnesota, Twin Cities, Sep. 2007.
[15] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "A lightweight concurrent fault detection scheme for the AES S-boxes using normal basis," in *Proc. CHES*, Aug. 2008, pp. 113–129.
[16] D. Canright, "A very compact S-box for AES," in *Proc. CHES*, Aug. 2005, pp. 441–455.
[17] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact Rijndael hardware architecture with S-box optimization," in *Proc. ASIACRYPT*, Dec. 2001, pp. 239–254.
[18] J. Wolkerstorfer, E. Oswald, and M. Lamberger, "An ASIC implementation of the AES SBoxes," in *Proc. CT-RSA*, 2002, pp. 67–78.
[19] V. Rijmen, Dept. ESAT, Katholieke Universiteit Leuven, Leuven, Belgium, Efficient Implementation of the Rijndael S-Box, 2000.
[20] X. Zhang and K. K. Parhi, "High-speed VLSI architectures for the AES algorithm," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. VLSI-12, no. 9, pp. 957–967, Sep. 2004.
[21] X. Zhang and K. K. Parhi, "On the optimum constructions of composite field for the AES algorithm," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 10, pp. 1153–1157, Oct. 2006.
[22] N. Mentens, L. Batina, B. Preneel, and I. Verbauwhede, "A systematic evaluation of compact hardware implementations for the Rijndael S-box," in *Proc. CT-RSA*, Feb. 2005, pp. 323–333.
[23] L. Breveglieri, I. Koren, and P. Maistri, "An operation-centered approach to fault detection in symmetric cryptography ciphers," *IEEE Trans. Computers*, vol. C-56, no. 5, pp. 534–540, May 2007.
[24] Xilinx [Online]. Available: http://www.xilinx.com/

Mehran Mozaffari-Kermani (S'07) received the B.Sc. degree in electrical engineering from the University of Tehran, Tehran, Iran, in 2005, and the M.E.Sc. degree in computer from the University of Western Ontario, London, ON, Canada, in 2007. He is currently working towards the Ph.D. degree in the Department of Electrical and Computer Engineering at the University of Western Ontario.

His current research interests include secure cryptographic systems, fault diagnosis and tolerance, VLSI reliability, and computer arithmetic.

Arash Reyhani-Masoleh (M'03) received the B.Sc. degree from Iran University of Science and Technology, Tehran, Iran, in 1989, the M.Sc. degree from the University of Tehran, Tehran, Iran, in 1991, both with the first rank in electrical and electronic engineering, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, 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. In October 2004, he joined the Department of Electrical and Computer Engineering, University of Western Ontario, London, Ontario, Canada, as an Assistant Professor. His current research interests include algorithms and VLSI architectures for computations in finite fields, fault tolerant computing, and error-control coding.

Dr. Reyhani-Masoleh was awarded a Natural Sciences and Engineering Research Council of Canada (NSERC) Postdoctoral Fellowship in 2002. Currently, he is an Associate Editor of *Integration, the VLSI Journal* (Elsevier).