Efficient Fault Diagnosis Schemes for Reliable Lightweight Cryptographic ISO/IEC Standard CLEFIA Benchmarked on ASIC and FPGA

Mehran Mozaffari-Kermani, *Member, IEEE*, and Reza Azarderakhsh

***Abstract*—Lightweight block ciphers are essential for providing low-cost confidentiality to sensitive constrained applications. Nonetheless, this confidentiality does not guarantee their reliability in the presence of natural and malicious faults. In this paper, fault diagnosis schemes for the lightweight internationally standardized block cipher CLEFIA are proposed. This symmetric-key cipher is compatible with yet lighter in hardware than the Advanced Encryption Standard and enables the implementation of cryptographic functionality with low complexity and power consumption. To the best of the authors' knowledge, there has been no fault diagnosis scheme presented in the literature for the CLEFIA to date. In addition to providing fault diagnosis approaches for the linear blocks in the encryption and the decryption of the CLEFIA, error detection approaches are presented for the nonlinear S-boxes, applicable to their composite-field implementations as well as their lookup table realizations. Through fault-injection simulations, the proposed schemes are benchmarked, and it is shown that they achieve error coverage of close to 100%. Finally, both application-specific integrated circuit and field-programmable gate array implementations of the proposed error detection structures are presented to assess their efficiency and overhead. The proposed fault diagnosis architectures make the implementations of the International Organization for Standardization/International Electrotechnical Commission-standardized CLEFIA more reliable.**

***Index Terms*—Application-specific integrated circuit (ASIC), CLEFIA symmetric-key block cipher, efficient error detection, field-programmable gate array (FPGA), reliability.**

I. Introduction

THE FLOURISH of the Internet of Things and sensitive embedded systems which can be deployed in both personal and industrial setups has made lightweight cryptography essential to reach acceptable confidentiality without adding much overhead to the constrained nodes [1], e.g., utilized as an added security measure for Mobile Ad hoc NETworks (MANETs) which lack physical layer security [2]. These lightweight cryptographic solutions need to provide high security levels to counteract the malicious intents of adversaries, similar to those for the Advanced Encryption Standard (AES) [3]. The lightweight block cipher CLEFIA [4] (presented in 2007) has been recently standardized in the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) (ISO/IEC 29192-2) [5] to provide security measures for constrained applications.

The CLEFIA provides acceptable confidentiality and, compared to the AES, has more compact hardware implementations [6]. It is noted that the strength of the CLEFIA has been benchmarked in a number of previous research works such as [7]–[9]. The CLEFIA is a 128-b block cipher utilized to protect the data transmitted in constrained applications such as radio-frequency-identification tags deployed in industrial setups, MANETs, handheld smart devices, and low-energy wearable medical devices. The CLEFIA's S-boxes are different from those of the AES both in terms of the irreducible polynomial used and the different affine transformations used before and after the multiplicative inversion. Not only does this lightweight cryptographic algorithm provide confidentiality for resource-constrained applications, but it can be also utilized as the fundamental component for various security purposes, including authentication, integrity (including lightweight hash), and pseudorandom number generation.

Natural fault detection (defects) has been the center of many previous works, for instance, see [10]–[13] for industrial electronics applications. Natural faults in the very-large-scale integration (VLSI) implementations of the cryptographic hardware are common due to hardware failures such as natural VLSI single event upsets, radiations, e.g., electromagnetic waves, which could induce faults, or aging causes. In the software realizations of these algorithms, these natural causes for faults cannot occur, and therefore, the natural failures in software could be mainly because of human factors. For the importance of fault occurrence in the former realization method, i.e., the CLEFIA in hardware, and, also, for the fact that it is lighter in hardware compared to the AES, hardware architectures for the CLEFIA are much appealing. Therefore, although, in general, error detection schemes can be realized using hardware and software implementations, in the case of the CLEFIA, hardware realizations are preferred for both the original and the error detection schemes to have better implementation/performance metrics.

Moreover, in cryptographic hardware systems implemented in critical and potentially sensitive contexts, with the presence

Manuscript received July 12, 2012; revised September 17, 2012; accepted October 31, 2012. Date of publication November 19, 2012; date of current version June 21, 2013.

M. Mozaffari-Kermani is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544, USA (e-mail: mozafari@princeton.edu).

R. Azarderakhsh is with the Center for Applied Cryptographic Research (CACR), Department of Combinatorics and Optimization, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: razarder@uwaterloo.ca).

Digital Object Identifier 10.1109/TIE.2012.2228144

0278-0046/$31.00 © 2012 IEEE

of malicious attackers attempting to take over the secret key, the reliability of the CLEFIA may be compromised. Fault attacks on the CLEFIA, which take advantage of the side-channel information leaked through fault injections, have been the center of attention in previous research works (see, for example, [14]–[18]). In such attacks, preferably, single-bit faults using the stuck-at model are injected. By repeatedly comparing the erroneous and error-free outputs, the last subkey is derived, and eventually, the secret key is compromised (Noting the technological constraints, an attacker may not be able to inject a single stuck-at fault. Therefore, multiple bits might be flipped). We note that the stuck-at fault model (both single and multiple) is able to model both natural and malicious faults and thus is utilized throughout this paper to achieve this twofold goal of the proposed schemes.

In cryptography, efficient implementations, e.g., for the point multiplication in elliptic curve cryptography [19], [20], [21] and fault diagnosis schemes, e.g., the schemes presented in [22]–[27], have been presented in some previous research works. Nevertheless, to the best of the authors' knowledge, this paper is the first to propose both lightweight and fault-immune architectures for the CLEFIA. In this paper, error detection schemes for this lightweight block cipher are proposed and benchmarked to reach more reliable hardware architectures. Almost all of the occurring natural faults are detected using the proposed methods. Although the proposed schemes may not result in a complete solution to the problem of intentionally injected faults, the high error coverage achieved would likely make the potential fault attacks more difficult.

The main contributions of this paper are summarized as follows.

1) Parity-prediction formulations are derived for the linear and nonlinear blocks of the CLEFIA and utilized in the proposed fault diagnosis scheme. For a subset of the S-boxes which are based on inversions in $GF(2^8)$, the schemes applicable to their composite-field and lookup-table-based hardware implementations are proposed.
2) The performed simulation results show high error coverage for the presented schemes. Using the proposed approaches, the error detection structures are capable of detecting very close to 100% of the injected faults.
3) Through application-specific integrated circuit (ASIC) syntheses using a 65-nm standard-cell library [28] and field-programmable gate array (FPGA) implementations on Virtex-5 FPGA devices [29], it is shown that the overheads of the proposed architectures are acceptable for resource-constrained applications.

The organization of this paper is as follows. In Section II, preliminaries related to the CLEFIA block cipher are presented. The proposed error detection approaches are presented in Section III. In Section IV, the results of the fault-injection simulations are presented. Moreover, through ASIC syntheses and FPGA implementations, the overheads are benchmarked. Finally, conclusions are made in Section V.

TABLE I
S-BOXES SS_0–SS_3 WITHIN THE S-BOX S_0 [4]

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$SS_0(x)$	e	6	c	a	8	7	2	f	b	1	4	0	5	9	d	3
$SS_1(x)$	6	4	0	d	2	b	a	3	9	c	e	f	8	7	5	1
$SS_2(x)$	b	8	5	e	a	6	4	c	f	7	2	3	1	0	d	9
$SS_3(x)$	a	2	6	d	3	4	5	e	0	7	8	9	b	f	c	1

II. PRELIMINARIES

The algorithm frame of the CLEFIA is as follows [4]. This block cipher consists of two parts, including a data processing part and a key scheduling part. The plaintext input and the ciphertext output (each contains 128-b blocks) are divided into 32-b parts. This is also performed for the whitening and round keys. The round keys are used as inputs to the main functions of the algorithm, and the whitening keys are XORed with the input and the output of the entire encryption/decryption. Then, the CLEFIA employs a type-2 generalized Feistel network with four data lines, each data line with a length of 32 b [4]. The CLEFIA is a 128-b block cipher with key lengths of 128, 192, and 256 b corresponding to 18, 22, and 26 numbers of rounds, respectively.

The data processing part of the CLEFIA consists of its encryption and decryption. The 128-b plaintext and ciphertext are divided into 32-b parts, and the encryption and the decryption are performed using whitening and round keys provided by the key scheduling unit [4]. Whitening keys are utilized at the beginning and the end of the CLEFIA, and round keys are used in its two main functions, i.e., 32-b functions F_0 and F_1 used in both the CLEFIA encryption and decryption [4].

Two nonlinear 8-b S-boxes are utilized in these functions, namely, S_0 and S_1. The S-box S_0 is generated by combining four 4-b random S-boxes, i.e., SS_0, SS_1, SS_2, and SS_3. This is shown in Table I in which the outputs for these four S-boxes are shown in hexadecimal form. The input $x \in GF(2^8)$ is divided into two parts, i.e., $x_0, x_1 \in GF(2^4)$, as follows:

$$\begin{aligned} t_0 &\leftarrow SS_0(x_0), \quad t_1 \leftarrow SS_1(x_1) \\ u_0 &\leftarrow t_0 \oplus 2.t_1, \quad u_1 \leftarrow 2.t_0 \oplus t_1 \\ y_0 &\leftarrow SS_2(u_0), \quad y_1 \leftarrow SS_3(u_1) \end{aligned} \tag{1}$$

where finite-field multiplications with the hexadecimal value 2 are performed in $GF(2^4)$ using $P_1(z) = z^4 + z + 1$.

The nonlinear S-box S_1 is based on the finite-field inversion over $GF(2^8)$. The primitive polynomial used for this multiplicative inversion is $P_2(z) = z^8 + z^4 + z^3 + z^2 + 1$. Two affine transformations are used in the S-box S_1, denoted hereafter as pre- and postinversion affine transformations, i.e., $f(.)$ and $g(.)$, respectively. Let $I, O \in GF(2^8)$ be the input and the output to this S-box, respectively. Then, if $f(I) = 0$, the output is $O = g(0) \in GF(2^8)$. Otherwise, the 8-b output of the S-box S_1 is computed as $O = g(f(I)^{-1}) \in GF(2^8)$.

Finally, two 4×4 diffusion matrices, i.e., $\boldsymbol{M}_0$ in F_0 and $\boldsymbol{M}_1$ in F_1, perform multiplications in $GF(2^8)$ using $P_2(z)$; more details are presented in the next section. The key scheduling part of the CLEFIA supports 128-, 192-, and 256-b keys to derive the whitening keys and round keys for the data processing part; one can refer to [4] for more details on the key scheduling unit.

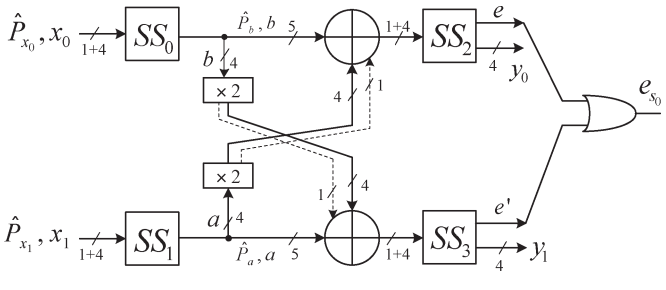

Fig. 1. Parity-based error detection structure of the S-box S_0.

III. Proposed Error Detection Schemes

In this section, the error detection approaches of subblocks in the CLEFIA encryption and decryption are proposed.

A. Proposed Scheme for the Nonlinear S-Boxes

Different approaches for the error detection of arithmetic units and cryptographic architectures (including the parity-based ones) have been investigated in previous works; see, for instance, [30]–[32]. It is noted that, in the parity-based fault diagnosis approaches used in this paper, the parity of a block (or subblock) is predicted using a preferably lightweight circuitry and compared with the actual parity to derive the error indication flag. In this section, error detection schemes for two nonlinear S-boxes in the F-functions of the encryption and the decryption of the CLEFIA are proposed.

1) S-Box S_0: The S-box S_0 presented in (1) is first considered. The first and last steps in (1) are substitution layers which are based on four random 4-b S-box circuits, i.e., SS_0–SS_3. These 4-b S-boxes can be implemented using lookup tables. For this purpose, 16×4 synthesized lookup tables or memory macros in ASIC or block or distributed memories in FPGAs (see [33] for the AES counterpart) can be used to store the 16 possible 4-b outputs of each S-box. Finally, the middle step is a linear finite-field multiplication layer.

Two error detection schemes for S_0 are proposed through Fig. 1. As seen in this figure, the linear and nonlinear layers of this S-box are depicted. The first scheme is based on the modification of the original S-boxes, and the second one is based on investing on the derived lightweight parity prediction circuitry of the intact original S-box S_0.

First, the scheme which is based on expanding the S-box S_0 is devised. It is noted that P and $\hat{P}$ denote the actual and the predicted parities, respectively. The original 16×4 S-boxes SS_0–SS_3 have been modified to 16×5 ones, i.e., their entries are now 5 b and the added bit is the modulo-2 addition (XOR) of the input and the output parities. For instance, as seen in Fig. 1, for SS_0 with the input x_0 and the output b, $P_{x_0} \oplus \hat{P}_b$ is added to each S-box entry, where $\oplus$ denotes the XOR operation. The predicted parity of the 4-b output vector b, i.e., $\hat{P}_b$, is then derived by adding the input parity $(\hat{P}_{x_0})$ and this stored value (the lookup table address is x_0). Similarly, $\hat{P}_a$ is derived for SS_1, where a is the 4-b output of this S-box. The same procedure is done for SS_2 and SS_3, as seen in Fig. 1. Finally, the error indication flag of the S-box, i.e., e_{S_0}, is derived as shown in Fig. 1. The following lemma is presented for the parity prediction of the linear multiplication layer of the S-box S_0.

Lemma 1: Let $P_1(z) = z^4 + z + 1$ be denoted as the primitive polynomial in $GF(2^4)$ for the finite-field multiplications in the S-box S_0. Let $\theta \in GF(2^4)$ be the input to the multiplication with constant 2 using $P_1(z)$, i.e., $z \in GF(2^4)$. Then, the predicted parity of the output $\delta \in GF(2^4)$ is derived as $\hat{P}_\delta = P_\theta + \theta_3$, where θ_3 is the most significant bit of $\theta = \theta_3 z^3 + \theta_2 z^2 + \theta_1 z + \theta_0$.

Proof: Considering $P_1(z) = z^4 + z + 1$, one can derive the result of $2 \times \theta$ after reduction as $\delta = \theta_2 z^3 + \theta_1 z^2 + (\theta_0 + \theta_3)z + \theta_3$. Therefore, by the modulo-2 addition of the coefficients, one can obtain $\hat{P}_\delta = P_\theta + \theta_3$, and the proof is complete. ■

Lemma 1 is used in Fig. 1 for the parity predictions of the two finite-field multiplications (see dotted lines in this figure).

The second parity-based scheme proposed for the S-box S_0 is based on deriving the predicted parities of the S-boxes SS_0–SS_3 using logic gates, noting their relatively small sizes, for instance, compared to those for the AES. Lemma 1 is used for the linear multiplication layer of the S-box S_0. We propose the following theorem for the parity prediction of the 16×4 S-boxes SS_0–SS_3.

Theorem 1: Let $\mu, \lambda, \psi, \chi \in GF(2^4)$ be considered as the inputs of the S-boxes SS_0–SS_3, respectively, corresponding to their respective 4-b vectors, e.g., $\mu = (\mu_3, \mu_2, \mu_1, \mu_0)$. The predicted parities of these S-boxes are derived as follows:

$$\begin{aligned}
\hat{P}_{SS_0} &= \mu_3\mu_1\overline{\mu_0} \vee \overline{\mu_1}\left((\mu_3 + \mu_2) \vee \overline{\mu_3\mu_0}\right) \\
\hat{P}_{SS_1} &= \lambda_2(\overline{\lambda_1} \vee \lambda_3\lambda_0) \vee \overline{\lambda_2}(\overline{\lambda_3}\lambda_0 \vee \lambda_3\lambda_1\overline{\lambda_0}) \\
\hat{P}_{SS_2} &= \overline{\psi_3\psi_2}\psi_1\overline{\psi_0} + \overline{\psi_2}(\overline{\psi_3} \vee \overline{\psi_1}\psi_0) \vee \overline{\psi_0}(\psi_3\psi_2 \vee \psi_1) \\
\hat{P}_{SS_3} &= \chi_0\left(\overline{\chi_3} \vee (\overline{\chi_2 + \chi_1})\right) \vee \chi_3\overline{\chi_0}(\chi_2 + \chi_1)
\end{aligned} \tag{2}$$

where $\vee$, $+$, and overline represent OR, XOR, and NOT operations, respectively.

Proof: Based on the CLEFIA's algorithm, the 16×4 S-boxes SS_0–SS_3 are defined in Table II, where the predicted parity of each entity has been derived and shown in the parenthesis next to the hexadecimal value of the entity. For each S-box, based on the 16 predicted parities of the 16 entries (shown in the parentheses in the rows of Table II), one can derive the formulations in (2). For example, for SS_0, one can derive $\hat{P}_{SS_0}$ based on the parities in the parentheses of the row of SS_0 as $\hat{P}_{SS_0} = \overline{\mu_3\mu_2\mu_1\mu_0} + \overline{\mu_3}\mu_2\overline{\mu_1\mu_0} + \overline{\mu_3}\mu_2\overline{\mu_1}\mu_0 + \mu_3\overline{\mu_2\mu_1\mu_0} + \mu_3\overline{\mu_2}\mu_1\mu_0 + \mu_3\overline{\mu_2}\mu_1\overline{\mu_0} + \mu_3\mu_2\mu_1\overline{\mu_0}$ which gives $\hat{P}_{SS_0} = \mu_3\mu_1\overline{\mu_0} + \mu_3\overline{\mu_2\mu_1} + \overline{\mu_3}\mu_2\overline{\mu_1} + \overline{\mu_3\mu_1\mu_0}$. Considering that $\mu_i\overline{\mu_j} \vee \mu_j\overline{\mu_i} = \mu_i + \mu_j$, one can obtain $\hat{P}_{SS_0} = \mu_3\mu_1\overline{\mu_0} \vee \overline{\mu_1}((\mu_3 + \mu_2) \vee \overline{\mu_3\mu_0})$. Similar methods can be used for deriving $\hat{P}_{SS_1} - \hat{P}_{SS_3}$ whose details are not presented for the sake of brevity. Therefore, the proof is complete. ■

Depending on the resources available and the platform to be utilized, one can choose the proposed error detection approaches for the S-box S_0. In what follows, the error detection schemes for the other nonlinear S-box of the F-functions are presented.

2) S-Box S_1: Unlike the S-box S_0 which consists of smaller random S-boxes, the S-box S_1 is based on arithmetic operations in $GF(2^8)$, including a multiplicative inversion and two affine transformations. This allows to devise the error detection schemes which are based on the arithmetic properties of this

TABLE II
PREDICTED PARITIES (IN PARENTHESES) OF THE S-BOXES SS_0–SS_3 WITHIN THE S-BOX S_0 (ENTRIES IN HEXADECIMAL FORM)

μ, λ, ψ, χ	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
SS_0	e(1)	6(0)	c(0)	a(0)	8(1)	7(1)	2(0)	f(0)	b(1)	1(1)	4(1)	0(0)	5(0)	9(0)	d(1)	3(0)
SS_1	6(0)	4(1)	0(0)	d(1)	2(1)	b(1)	a(0)	3(0)	9(0)	c(0)	e(1)	f(0)	8(1)	7(1)	5(0)	1(1)
SS_2	b(1)	8(1)	5(0)	e(1)	a(0)	6(0)	4(1)	c(0)	f(0)	7(1)	2(1)	3(0)	1(1)	0(0)	d(1)	9(0)
SS_3	a(0)	2(1)	6(0)	d(1)	3(0)	4(1)	5(0)	e(1)	0(0)	7(1)	8(1)	9(0)	b(1)	f(0)	c(0)	1(1)

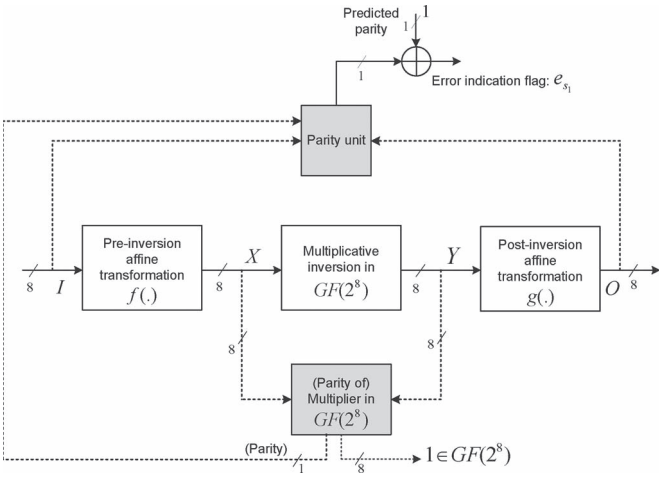

Fig. 2. Error detection structure of the S-box S_1 including pre- and postinversion affine transformations.

S-box rather than storing the predicted parities in the S-boxes. One important benefit of this approach is its applicability to the composite-field S-boxes as well. These S-boxes have lower hardware complexity and can be pipelined for higher performance compared to the ones using lookup tables (see, for instance, [34] for the AES). It is noted that, as seen in the following, a nonlinear multiplication method is used rather than a linear method so that the proposed approach becomes independent of the structure of the CLEFIA's S-box S_1.

Fig. 2 illustrates the obtained error detection scheme for the entire S-box S_1, including its linear and nonlinear blocks. As seen in this figure, one can multiply the input and the output of the multiplicative inversion in $GF(2^8)$ and then compare the result with $1 \in GF(2^8)$ using the irreducible polynomial $P_2(z)$ which is unique for the CLEFIA. Another unique characteristics for the CLEFIA is that it consists of two new different affine transformations before and after the multiplicative inversion in $GF(2^8)$. It is noted that using finite-field multipliers for the input and the output of the inversion [and then comparing the result with $1 \in GF(2^8)$] introduces much overhead to the S-box S_1. Based on the aforementioned observations, a parity-based scheme for the S-box S_1 is proposed considering the following theorem which is adopted from [31].

Theorem 2: Let C be the product of two arbitrary elements A and B of $GF(2^m)$. Let $\hat{P}_{Z^{(j)}} = P_A + \sum_{k=0}^{j-1} Z_{m-1}^{(k)}$ for $1 \le j \le m-1$, where P_A is the parity (modulo-2 addition of bits) of A. Moreover, let $Z_{m-1}^{(k)}$ be the $(m-1)$th coordinate of $Z^{(j)} = A\alpha^j \bmod F(\alpha)$. We note that, for the irreducible polynomial $F(\alpha)$, α^j denotes exponents for α. Then, $\hat{P}_C = \sum_{j=0}^{m-1} b_j \hat{P}_{Z^{(j)}}$.

Based on Theorem 2, the following theorem is presented for the parity prediction of the S-box S_1.

Theorem 3: Considering the S-box S_1 presented in Fig. 2 with the input $I \in GF(2^8)$ and the output $O \in GF(2^8)$, one can obtain the parity prediction of the multiplicative inversion as well as the preinversion $(f(.))$ and postinversion $(g(.))$ affine transformations for the case $f(I) = X \neq 0$ as

$$\begin{aligned}\hat{P}_{S_1} =& i_0(\overline{P_O} + o_5 + o_2 + o_0) + i_1(P_O + o_6 + o_4 + o_2) \\ &+ i_2(o_5 + \overline{o_4}) + i_3(\overline{P_O} + o_7 + o_4 + o_1) \\ &+ i_4(o_7 + o_5 + o_2) + i_5(P_O + o_7 + o_3 + o_2) \\ &+ i_6(o_6 + o_4 + o_2) + i_7(\overline{P_O} + o_6 + o_5 + o_4) \\ &+ o_5 + \overline{o_4} + o_2 + o_1 \end{aligned} \tag{3}$$

where $i = (i_7, \ldots, i_0)$ and $o = (o_7, \ldots, o_0)$ are the vectors corresponding to $I, O \in GF(2^8)$, respectively, and P_O is the parity of O.

Proof: According to Fig. 2 and Theorem 2, first, the predicted parity of the multiplicative inversion with the input $X \in GF(2^8)$ and the output $Y \in GF(2^8)$ is derived. Utilizing $P_2(z) = z^8 + z^4 + z^3 + z^2 + 1$ as the polynomial for reductions, one can consecutively first obtain the most significant bits of $X\alpha^j \bmod P_2(\alpha)$ for $1 \le j \le 7$. Then, based on Theorem 2, the following is obtained for the predicted parity of multiplication, i.e., $\hat{P}_M$:

$$\begin{aligned}\hat{P}_M =& x_0 P_Y + x_1(P_Y + y_7) + x_2(P_Y + y_7 + y_6) \\ &+ x_3(P_Y + y_7 + y_6 + y_5) + x_4(y_3 + y_2 + y_1 + y_0) \\ &+ x_5(y_7 + y_2 + y_1 + y_0) + x_6(y_6 + y_1 + y_0) \\ &+ x_7(y_7 + y_5 + y_0). \end{aligned} \tag{4}$$

The affine transformations $f(.)$ and $g(.)$ are presented in [4]. The inverse of the postinversion affine transformation $g(.)$, i.e., $g^{-1}(.)$, has been derived and presented hereinafter, respectively,

$$f(.): \quad \boldsymbol{x} = \begin{pmatrix} 0&0&0&1&1&0&0&0 \\ 0&1&0&1&0&0&0&1 \\ 0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&1&1&0 \\ 0&1&1&0&0&1&0&1 \\ 0&1&0&1&1&1&0&0 \\ 0&1&1&0&0&0&0&0 \\ 1&0&0&0&0&0&0&1 \end{pmatrix} \boldsymbol{i} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

$$g(.): \quad \boldsymbol{o} = \begin{pmatrix} 0&0&0&0&1&0&1&0 \\ 0&1&0&0&0&0&0&1 \\ 0&1&0&1&1&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&1&1&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 1&0&0&1&0&0&0&0 \\ 0&1&0&0&0&1&0&0 \end{pmatrix} \boldsymbol{y} + \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$g^{-1}(.): \quad \boldsymbol{y} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \boldsymbol{o} + \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}.$$

Based on Fig. 2 and considering (4), one can substitute the coordinates of $X, Y \in GF(2^8)$ with those of $I, O \in GF(2^8)$ to derive (3) after reordering and some calculations. This completes the proof. ■

Using subexpression sharing, the hardware complexity of (3) is decreased. As seen in Fig. 2, the obtained predicted parity can be XORed with 1 (or simply inverted) to derive the error indication flag e_{S_1}. It is noted that the case of $X = 0$ is also separately detected to compare the final result with $g(0)$ for error detection. In the following section, the proposed scheme for the two linear operations (4×4 diffusion matrices) in the F-functions is presented.

B. Proposed Scheme for the Diffusion Matrices

In what follows, for every 8 b of the two 4×4 diffusion matrices, i.e., $\boldsymbol{M}_0$ in F_0 and $\boldsymbol{M}_1$ in F_1, predicted parities are derived. The following theorem is presented for these matrices.

Theorem 4: Let $X = X_1|X_2|X_3|X_3$ and $Y = Y_1|Y_2|Y_3|Y_3$ be the 32-b input and output (divided into four 8-b parts each) of the $\boldsymbol{M}_0$ (or $\boldsymbol{M}_1$) matrix, respectively. Then, $\sum_{i=1}^{3} X_i = \sum_{i=1}^{3} Y_i$.

Proof: Matrices $\boldsymbol{M}_0$ and $\boldsymbol{M}_1$ with hexadecimal entries are as follows:

$$\boldsymbol{M}_0 = \begin{pmatrix} 01 & 02 & 04 & 06 \\ 02 & 01 & 06 & 04 \\ 04 & 06 & 01 & 02 \\ 06 & 04 & 02 & 01 \end{pmatrix} \quad \boldsymbol{M}_1 = \begin{pmatrix} 01 & 08 & 02 & 0a \\ 08 & 01 & 0a & 02 \\ 02 & 0a & 01 & 08 \\ 0a & 02 & 08 & 01 \end{pmatrix}.$$

For these matrices, the modulo-2 additions of each column entry is $\{01\}_h$. For instance, for $\boldsymbol{M}_1$, the first column becomes $\{01 + 08 + 02 + 0a\}_h = \{01 + 08 + 02 + 08 + 02\}_h = \{01\}_h$. Thus, $\sum_{i=1}^{3} X_i = \sum_{i=1}^{3} Y_i$, and the proof is complete. ■

Theorem 4 implies that there is no overhead in deriving one predicted parity for these matrices, i.e., $\hat{P}_X = \hat{P}_Y$. Moreover, it gives freedom in utilizing up to eight checkers for each of the matrices.

This section is finalized by briefly presenting the overall structures of the error detection schemes for the CLEFIA encryption and decryption. The error detection structures of two main functions of the CLEFIA are shown in Fig. 3. As seen in this figure and discussed in the previous section, one parity bit is used for every 8 b of the nonlinear S-boxes (see Lemma 1, Figs. 1 and 2, and Theorems 1 and 3) and linear diffusion matrices (see Theorem 4). Using the proposed fault diagnosis approaches in this section for the F-functions, the fault diagnosis of these operations is straightforward. The CLEFIA's encryption and decryption parity-based error detection architectures are also presented in Fig. 4. As seen in this figure, the encryption and the decryption include similar F-functions and modulo-2 additions including those with the whitening keys, whose parity predictions are the additions of the input parities (shown by P_{WK} and P_{RK} in Fig. 4 for whitening or round keys, respectively). Moreover, for the key scheduling part, the F-functions are the major components inside the generalized Feistel network, and thus, the presented scheme is applicable to this part as well.

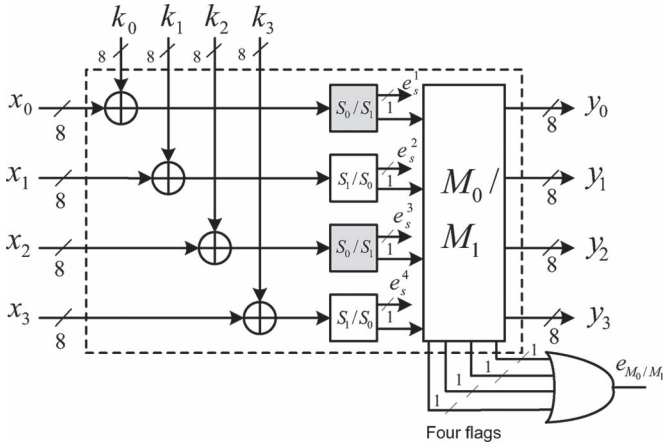

Fig. 3. Error detection structures of the two main functions of the CLEFIA, i.e., F-functions F_0/F_1.

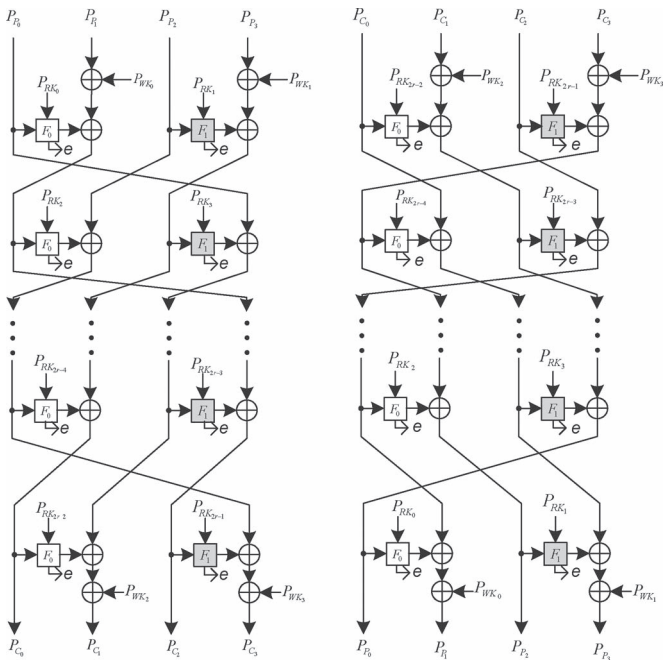

Fig. 4. Error detection structures of the CLEFIA's (left) encryption and (right) decryption.

IV. Analysis of the Proposed Schemes

The error coverage assessments and overhead benchmark of the error detection structures are presented in this section.

TABLE III
ASIC SYNTHESES OF THE ORIGINAL AND THE ERROR DETECTION STRUCTURES FOR THE CLEFIA ENCRYPTION/DECRYPTION USING A 65-nm CMOS STANDARD-CELL LIBRARY

Key-size	Architecture	Area [GE]	Frequency [MHz]	Throughput [Gbps]	Efficiency [Kbps/GE]
CLEFIA-128	Original	9,423	606	4.31	457
	Error detection	11,114 (17.9%)	577 (4.8%)	4.10 (4.8%)	369 (19.3%)
CLEFIA-192	Original	14,388	532	3.09	215
	Error detection	16,809 (16.8%)	511 (3.9%)	2.97 (3.9%)	176 (18.1%)
CLEFIA-256	Original	14,320	532	2.62	183
	Error detection	17,017 (18.8%)	511 (3.9%)	2.51 (3.9%)	147 (19.6%)

A. Error Coverage

Throughout this paper, both single and multiple stuck-at faults are considered. These models cover both natural faults and malicious fault attacks [23]. If exactly 1-b error occurs at the output of the linear or nonlinear blocks of the CLEFIA functions, the presented parity-based error detection approach is able to detect it, and the error coverage of the proposed scheme is 100%; thus, no simulation is needed for this case. Although it is not claimed that all the presented fault attacks are entirely detected, the proposed approach would make it difficult for these attacks to be mounted, e.g., analytically, more than 99.99% of the faults through the attack presented in [17] are detected.

Noting the technological constraints, an attacker may not be able to inject single stuck-at faults to flip exactly 1 b [23]. Thus, multiple bits will actually be flipped. Most internal faults are modeled by transient random faults [23]. Therefore, by relying on simulations, error coverage through multiple stuck-at fault injections is evaluated for the CLEFIA encryption and decryption. The results of the performed simulations are valid for both transient faults and permanent internal faults.

Based on the used fault model presented in this section, stuck-at faults (both stuck-at zero and stuck-at one) are injected in multiple random locations. For the sake of brevity, the CLEFIA-128 encryption and decryption have been considered as reference for the fault-injection simulations. By applying 100 000 random inputs, 1000 multiple and random faults (in terms of the type of the fault, its location, and its count) have been injected for each input. For each injection, error indication flags are monitored, and the detected errors are counted for the encryption and the decryption operations.

The results of the performed simulations show that more than 99.999% of the errors are detected (these comply with the theoretical results using 16 predicted parities per round). Only the instances in which the output is erroneous are evaluated, and the ones in which the injected faults are masked, i.e., the output is correct, are not relevant for calculating the error coverage. Moreover, it is assumed that the comparison units (which are composed of simple XOR gates), comparing the actual and the predicted parities, are fault free. The obtained error coverage of very close to 100% makes the hardware implementations of the CLEFIA more reliable. To theoretically verify the error coverage for multiple errors, let us denote p as the probability of error detection of one parity bit. Therefore, $1-p^n$ is the error detection probability for using n parity bits. Now, let us consider the CLEFIA-128 with 18 rounds (18 F_0s and 18 F_1s), each of which uses eight predicted/actual parities. For $p=0.5$ and $n=36\times 8$, we have the error detection probability as $1-(2\times 10^{-87})\simeq 0.99999$, which complies with our error simulation results.

As it is seen in the next section, the proposed scheme has the area overhead of less than 20% and delay overhead of less than 8% on both ASIC and FPGA (compared to 100% overhead for the redundancy-based approaches). With these acceptable overheads, the error coverage of 100% is achieved in case of ideal fault attacks with 1-b flips, and in other cases, the proposed scheme reaches the error coverage of about 100% [theoretically $1-(2\times 10^{-87})$] which would make it much difficult for these attacks to be mounted.

B. ASIC and FPGA Overhead Benchmark

This section presents the results of the overhead assessments using the ASIC and the FPGA hardware platforms. The analysis has been performed for the original and the error detection structures of the CLEFIA-128, CLEFIA-192, and CLEFIA-256. Through this benchmarking, the overheads (degradations) are derived. A 65-nm standard-cell library [28] has been used for the ASIC results using Design Compiler [35]. Moreover, ISE version 13.4 and Virtex-5 FPGA device xc5vlx50t-3 [29] have been utilized for the FPGA implementations (FPGAs are used in high-performance and cost-effective applications [36]–[39], including FPGA-based control units/systems for fault diagnosis [40]). Verilog has been used as the design entry for the original and the error detection structures.

Let us explain the ASIC and the FPGA architectures whose results are shown in Tables III and IV. We have followed the loop architecture implementation methods for the original CLEFIA-128, CLEFIA-192, and CLEFIA-256, which have been realized in the CLEFIA standard as well. In this architecture which takes one clock cycle per round, both F-functions, i.e., 32-b functions F_0 and F_1 used in the CLEFIA encryption and decryption, are implemented in parallel (not merged). However, we note that our schemes are independent of the methods that these functions are realized. For this method, 18, 22, and 26 clock cycles corresponding to the same number of rounds in these three original CLEFIA structures are needed. For the error detection structures, both the predicted parity architectures and the structures for the actual parities (modulo-2 addition of the respective bits of the corresponding outputs) are synthesized and implemented. Moreover, the comparison units are included in these architectures in order to have complete benchmarks for the entire error detection architectures. All the syntheses

TABLE IV
FPGA IMPLEMENTATION RESULTS FOR THE ORIGINAL CLEFIA ENCRYPTION/DECRYPTION AND ITS PROPOSED ERROR DETECTION SCHEME ON VIRTEX-5 FPGA DEVICE XC5VLX50T-3

Key-size	Architecture	Area [Slices]	Frequency [MHz]	Throughput [Gbps]	Efficiency [Kbps/Slices]
CLEFIA-128	Original	361	180	1.28	3,546
	Error detection	432 (19.7%)	171 (5.0%)	1.22 (5.0%)	2,824 (20.4%)
CLEFIA-192	Original	621	139	0.89	1,433
	Error detection	718 (15.6%)	128 (7.9%)	0.74 (7.9%)	1,031 (28.1%)
CLEFIA-256	Original	621	140	0.69	1,111
	Error detection	718 (15.6%)	130 (7.1%)	0.64 (7.1%)	891 (19.8%)

and implementations are performed using the same settings (medium map and area efforts), tools, devices (for FPGA), and libraries (for ASIC) to have a meaningful benchmark.

We would like to emphasize that these results are for benchmarking purposes and the original architectures could be optimized. However, this does not change the applicability of the proposed architectures. As seen in these two tables, the area, frequency, throughput (considering 18, 22, and 26 rounds for the different key sizes of the CLEFIA, respectively), and efficiency $(throughput/area)$ for the original and the error detection structures of the CLEFIA with different key sizes are presented. The area of the ASIC designs are presented in terms of NAND gate equivalent (GE) in order to make the area results meaningful when switching technologies. Moreover, for the FPGA implementations, the number of occupied slices is reported.

As seen in Tables III and IV, the overheads are presented in parentheses. For the ASIC syntheses in Table III and the FPGA implementations in Table IV, the highest area overheads are 18.8% and 19.7%, respectively. The maximum overheads for the frequencies and the throughputs on ASIC and FPGA are 4.8% and 7.9%, respectively. Finally, the maximum efficiency degradations in Tables III and IV are 19.6% and 28.1%, respectively. Based on the simulation results in this section, these overheads are added for the error coverage of very close to 100%. The proposed effective fault diagnosis approaches provide high error coverage at the expense of the presented acceptable overheads on the hardware platforms, making the hardware architectures of the CLEFIA more reliable.

V. CONCLUSION

In this paper, new fault diagnosis approaches for the recently standardized lightweight block cipher CLEFIA have been presented. These include the nonlinear S-boxes and the linear diffusion matrices of the F-functions, the main components of the CLEFIA encryption and decryption. The presented scheme for the S-box of the CLEFIA, including the finite-field inversion and pre- and postinversion affine transformations, is applicable to the low-complexity composite-field realizations as well as the lookup table ones. The results of the error coverage analysis show very high error coverage (close to 100%) for the proposed error detection structures for the injected stuck-at faults. Moreover, the ASIC and the FPGA analysis results show acceptable overheads for different key sizes of the CLEFIA when the presented schemes are utilized.

The proposed schemes are a step forward in the area of fault-immune lightweight cryptographic hardware, essential for protecting the extremely sensitive applications. The applicability of the schemes presented is not confined to lightweight block ciphers; lightweight hash functions with similar structures can be made more reliable investing on the schemes devised, making the presented schemes suitable for providing reliability to their lightweight security-constrained hardware implementations.

REFERENCES

[1] R. Roman, P. Najera, and J. Lopez, "Securing the Internet of Things," *Computer*, vol. 44, no. 9, pp. 51–58, Sep. 2011.

[2] E. M. Shakshuki, N. Kang, and T. R. Sheltami, "EAACK—A secure intrusion detection system for MANETs," *IEEE Trans. Ind. Electron.*, vol. 60, no. 3, pp. 1089–1098, Mar. 2013.

[3] National Institute of Standards and Technologies, "Announcing the Advanced Encryption Standard (AES)," Federal Information Processing Standards Publication, no. 197, Nov. 2001.

[4] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, "The 128-bit block cipher CLEFIA," in *Proc. FSE*, Mar. 2007, pp. 181–195.

[5] *CLEFIA Standardization in ISO/IEC 29192-2*, accessed in November 2012. [Online]. Available: http://www.sony.net/Products/cryptography/clefia/standard/iso.html

[6] T. Akishita and H. Hiwatari, "Very compact hardware implementations of the blockcipher CLEFIA," in *Proc. SAC*, Aug. 2011, pp. 278–292.

[7] Y. Tsunoo, E. Tsujihara, M. Shigeri, T. Saito, T. Suzuki, and H. Kubo, "Impossible differential cryptanalysis of CLEFIA," in *Proc. LNCS FSE*, Feb. 2008, pp. 398–411.

[8] A. Bogdanov and V. Rijmen, "Zero-correlation linear cryptanalysis of block ciphers," in *Proc. IACR Cryptol. ePrint*, 2011, p. 123.

[9] X. Tang, B. Sun, R. Li, and C. Li, "Impossible differential cryptanalysis of 13-round CLEFIA-128," *J. Syst. Softw.*, vol. 84, no. 7, pp. 1191–1196, Jul. 2011.

[10] A. Yazdani, H. Sepahvand, M. Crow, and M. Ferdowsi, "Fault detection and mitigation in multilevel converter STATCOMs," *IEEE Trans. Ind. Electron.*, vol. 58, no. 4, pp. 1307–1315, Apr. 2011.

[11] M. A. Rodríguez-Blanco, A. Claudio-Sânchez, D. Theilliol, L. G. Vela-Valdês, P. Sibaja-Terân, L. Hernândez-Gonzâlez, and J. Aguayo-Alquicira, "A failure-detection strategy for IGBT based on gate-voltage behavior applied to a motor drive system," *IEEE Trans. Ind. Electron.*, vol. 58, no. 5, pp. 1625–1633, May 2011.

[12] T. A. Najafabadi, F. R. Salmasi, and P. Jabehdar-Maralani, "Detection and isolation of speed-, dc-link voltage-, and current-sensor faults based on an adaptive observer in induction-motor drives," *IEEE Trans. Ind. Electron.*, vol. 58, no. 5, pp. 1662–1672, May 2011.

[13] S. Cruz, M. Ferreira, A. Mendes, and A. J. M. Cardoso, "Analysis and diagnosis of open-circuit faults in matrix converters," *IEEE Trans. Ind. Electron.*, vol. 58, no. 5, pp. 1648–1661, May 2011.

[14] H. Chen, W. Wu, and D. Feng, "Differential fault analysis on CLEFIA," in *Proc. LNCS ICICS*, Dec. 2007, pp. 284–295.

[15] J. Takahashi and T. Fukunaga, "Improved differential fault analysis on CLEFIA," in *Proc. LNCS FDTC*, Aug. 2008, pp. 25–34.

[16] J. Takahashi and T. Fukunaga, "Differential fault analysis on CLEFIA with 128, 192, and 256-bit keys," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E93.A, no. 1, pp. 136–143, Jan. 2010.

[17] S. Ali and D. Mukhopadhyay, "Protecting last four rounds of CLEFIA is not enough against differential fault analysis," in *Proc. IACR Cryptol. ePrint*, May 2012, p. 286.

[18] J. Takahashi and T. Fukunaga, "Differential fault analysis on the AES key schedule," in *Proc. IACR Cryptol. ePrint*, Dec. 2007, p. 480.

[19] G. Sutter, J. P. Deschamps, and J. Imaña, "Modular multiplication and exponentiation architectures for fast RSA cryptosystem based on digit

serial computation," *IEEE Trans. Ind. Electron.*, vol. 58, no. 7, pp. 3101–3109, Jul. 2011.
[20] G. Sutter, J. P. Deschamps, and J. Imaña, "Efficient elliptic curve point multiplication using digit serial binary field operations," *IEEE Trans. Ind. Electron.*, vol. 60, no. 1, pp. 217–225, Jan. 2013.
[21] R. Azarderakhsh and A. Reyhani-Masoleh, "Efficient FPGA implementations of point multiplication on binary Edwards and generalized Hessian curves using Gaussian normal basis," *IEEE Trans. VLSI Syst.*, vol. 20, no. 8, pp. 1453–1466, 2012.
[22] C. H. Yen and B. F. Wu, "Simple error detection methods for hardware implementation of advanced encryption standard," *IEEE Trans. Comput.*, vol. 55, no. 6, pp. 720–731, Jun. 2006.
[23] L. Breveglieri, I. Koren, and P. Maistri, "An operation-centered approach to fault detection in symmetric cryptography ciphers," *IEEE Trans. Comput.*, vol. 56, no. 5, pp. 635–649, May 2007.
[24] P. Maistri and R. Leveugle, "Double-data-rate computation as a countermeasure against fault analysis," *IEEE Trans. Comput.*, vol. 57, no. 11, pp. 1528–1539, Nov. 2008.
[25] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "Concurrent structure-independent fault detection schemes for the advanced encryption standard," *IEEE Trans. Comput.*, vol. 59, no. 5, pp. 608–622, May 2010.
[26] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "A low-power high-performance concurrent fault detection approach for the composite field S-box and inverse S-box," *IEEE Trans. Comput.*, vol. 60, no. 9, pp. 1327–1340, Sep. 2011.
[27] R. Karri and X. Guo, "Invariance-based concurrent error detection for advanced encryption standard," in *Proc. DAC*, Jun. 2012, pp. 573–578.
[28] TSMC. [Online]. Available: http://www.tsmc.com/
[29] Xilinx. [Online]. Available: http://www.xilinx.com/
[30] N. A. Touba and E. J. McCluskey, "Logic synthesis of multilevel circuits with concurrent error detection," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 16, no. 7, pp. 783–789, Jul. 1997.
[31] A. Reyhani-Masoleh and M. A. Hasan, "Fault detection architectures for field multiplication using polynomial bases," *IEEE Trans. Comput.—Special Issue Fault Diagnosis Tolerance Cryptogr.*, vol. 55, no. 9, pp. 1089–1103, Sep. 2006.
[32] C.-Y. Lee, P. K. Meher, and J. C. Patra, "Concurrent error detection in bit-serial normal basis multiplication over $GF(2^m)$ using multiple parity prediction schemes," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 8, pp. 1234–1238, Aug. 2010.
[33] T. Good and M. Benaissa, "Very small FPGA application-specific instruction processor for AES," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 7, pp. 1477–1486, Jul. 2006.
[34] 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.
[35] Synopsys. [Online]. Available: http://www.synopsys.com/
[36] J. J. Rodriguez-Andina, M. J. Moure, and M. D. Valdes, "Features, design tools, and application domains of FPGAs," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1810–1823, Aug. 2007.
[37] F. J. Azcondo, A. de Castro, and C. Branas, "Course on digital electronics oriented to describing systems in VHDL," *IEEE Trans. Ind. Electron.*, vol. 57, no. 10, pp. 3308–3316, Oct. 2010.
[38] E. Monmasson and M. N. Cirstea, "FPGA design methodology for industrial control systems—A review," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1824–1842, Aug. 2007.
[39] S.-K. Lam, T. Srikanthan, and C. Clarke, "Selecting profitable custom instructions for area-time-efficient realization on reconfigurable architectures," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 3998–4005, Oct. 2009.
[40] M. Shahbazi, P. Poure, S. Saadate, and M. Zolghadri, "FPGA-based reconfigurable control for fault-tolerant back-to-back converter without redundancy," *IEEE Trans. Ind. Electron.*, vol. 60, no. 8, pp. 3360–3371, Aug. 2013.

Mehran Mozaffari-Kermani (M'12) received the B.Sc. degree in electrical and computer engineering from the University of Tehran, Tehran, Iran, in 2005 and the M.E.Sc. and Ph.D. degrees from the Department of Electrical and Computer Engineering, The University of Western Ontario, London, ON, Canada, in 2007 and 2011, respectively.

After the completion of his Ph.D., he worked with Advanced Micro Devices as a Senior Application-Specific Integrated Circuit (ASIC)/Layout Designer, integrating sophisticated security/cryptographic capabilities into a single chip. He is currently with the Department of Electrical Engineering, Princeton University, Princeton, NJ. His research interests include developing security/privacy measures for emerging technologies, cryptographic systems, fault diagnosis and tolerance in cryptographic hardware and embedded systems, VLSI reliability, and low-power secure and efficient field-programmable gate array and ASIC designs.

Dr. Mozaffari-Kermani was awarded a Natural Sciences and Engineering Research Council of Canada (NSERC) Postdoctoral Fellowship in 2011. Currently, he is an NSERC Postdoctoral Research Fellow in the Department of Electrical Engineering of Princeton University.

Reza Azarderakhsh received the B.Sc. degree in electrical and electronic engineering in 2002, the M.Sc. degree in computer engineering from the Sharif University of Technology, Tehran, Iran, in 2005, and the Ph.D. degree in electrical and computer engineering from The University of Western Ontario, London, ON, Canada, in 2011.

In September 2011, he joined the Department of Electrical and Computer Engineering of The University of Western Ontario as a Limited Duties Instructor. He is currently a Postdoctoral Fellow in the Center for Applied Cryptographic Research and the Department of Combinatorics and Optimization at the University of Waterloo, Waterloo, ON. His current research interests include finite field and its application, elliptic curve cryptography, and pairing-based cryptography.