Reliable and Error Detection Architectures of Pomaranch for False-Alarm-Sensitive Cryptographic Applications

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

***Abstract*— Efficient cryptographic architectures are used extensively in sensitive smart infrastructures. Among these architectures are those based on stream ciphers for protection against eavesdropping, especially when these smart and sensitive applications provide life-saving or vital mechanisms. Nevertheless, natural defects call for protection through *design for fault detection and reliability*. In this paper, we present implications of fault detection cryptographic architectures (Pomaranch in the hardware profile of European Network of Excellence for Cryptology) for smart infrastructures. In addition, we present low-power architectures for its nine-to-seven *uneven* substitution box [tower field architectures in GF(3^3)]. Through error simulations, we assess resiliency against false-alarms which might not be tolerated in sensitive intelligent infrastructures as one of our contributions. We further benchmark the feasibility of the proposed approaches through application-specific integrated circuit realizations. Based on the reliability objectives, the proposed architectures are a step-forward toward reaching the desired objective metrics suitable for intelligent, emerging, and sensitive applications.**

***Index Terms*— Application-specific integrated circuit (ASIC), reliability, smart infrastructures.**

I. Introduction

CRYPTOGRAPHIC architectures provide protection for sensitive and smart infrastructures such as secure healthcare, smart grid, fabric, and home [1]–[8]. Nonetheless, the use of cryptographic architectures does not guarantee immunity against faults occurring in these infrastructures. Defects in VLSI systems may cause smart usage models to malfunction. Extensive research has been done for detecting such faults in the cryptographic algorithms such as elliptic curve cryptography and the Advanced Encryption Standard (AES) [9]–[14] (also refer to [15] for reliable architectures for lightweight cryptography).

Design for reliability and fault immunity ensures that with the presence of faults, reliability is provided for the aforementioned sensitive cryptographic architectures. The proposed work presents false-alarm sensitive fault detection schemes for cryptostructures (we note that to have a thorough analysis, we choose the Pomaranch stream cipher also known as a cascade jump controlled sequence generator (CJCSG) [16]–[18]). Such false alarms could be exploited to induce distrust to the user, i.e., repetitive false detections result in either ignoring the alarms by the user or abandoning the devices in which the cryptographic architectures are embedded. From a user's point of view, at the very least, this is uncomfortable; however, false alarms could lead to financial loss if abandoning the crypto-architectures happens. Finally, such a false detection would result in higher dynamic power consumption, resulting in extra energy depletion especially for constrained applications. The uneven architecture of this cipher presents unique challenges, which are motivations to its choice for the proposed work.

We would like to emphasize that the proposed work can be applied to similar ciphers and this paper does not intend to benchmark the algorithmic attacks or the performance efficiency for a certain cipher. Pomaranch is classified in the hardware profile of European Network of Excellence for Cryptology. This stream cipher includes an uneven substitution box (also refer to [19]) and has been the center of attention to achieve efficient hardware architectures. Natural defects, which are inevitable in VLSI systems call for protecting these architectures through detection mechanisms to preserve their reliability.

A. Major Contributions

Assessing the implications of providing fault detection and secure architectures useful for emerging usage models and smart infrastructures is of paramount importance. These architectures need to be feasible to use for different performance and implementation objectives of sensitive smart applications. Moreover, with respect to concurrent error detection architectures, the fault diagnosis approaches proposed for the structures of the nine-to-seven substitution box need to be carefully devised to detect possible false alarms. The main contributions of this paper can be summarized as follows.

1) We present lightweight and low-power architectures for the substitution box of the Pomaranch stream cipher (realized in composite fields). The proposed structures are based on tower field architectures of this substitution box in [19]. Specifically, we present low-power restructured architectures for this uneven substitution box useful for emerging constrained and sensitive usage models.

Manuscript received July 8, 2014; revised October 8, 2014 and December 7, 2014; accepted December 14, 2014. Date of publication January 14, 2015; date of current version November 20, 2015. The work of M. Mozaffari-Kermani and R. Azarderakhsh was supported by the Texas Instruments Faculty Award.

M. Mozaffari-Kermani and A. Aghaie are with the Department of Electrical and Microelectronics, Rochester Institute of Technology, Rochester, NY 14623 USA (e-mail: mmkeme@rit.edu; aa6964@mail.rit.edu).

R. Azarderakhsh is with the Department of Computer Engineering, Rochester Institute of Technology, Rochester, NY 14623 USA (e-mail: rxaeec@rit.edu).

Digital Object Identifier 10.1109/TVLSI.2014.2382715

2) We propose fault diagnosis approaches for the light-weight and low-power architectures of the nine-to-seven substitution box of Pomaranch. The proposed framework can be modified based on the objectives to achieve.
3) Through simulations for various fault models, we benchmark the fault detection capability of the proposed schemes. The occurrence of false alarms is evaluated through simulations and the approaches to avoid them are elaborated.
4) Finally, we synthesize the proposed fault detection architectures on the application-specific integrated circuit (ASIC) platform using 65-nm CMOS technology. Our results show that the proposed efficient fault detection architectures can be feasibly utilized for reliable architectures of the Pomaranch stream cipher making them suitable for the required performance and implementation metrics to achieve.

The organization of this paper is as follows. In Section II, we review the preliminaries. In Section III, the modified architectures for the nine-to-seven substitution box are presented. In Section IV, the proposed fault detection architectures are presented. In Section V, the proposed architectures are benchmarked through fault simulations to assess their effectiveness for various fault models. In Section VI, the efficiencies of the proposed structures are benchmarked through ASIC syntheses. Finally, we conclude in Section VII.

II. Preliminaries

In what follows, preliminaries on the substitution box of the Pomaranch stream cipher, the most complex architecture in the design of this cipher, and also fault diagnosis are presented. The structure of Pomaranch is based on linear feedback shift registers that allow fast implementation and produce sequences with a large period if the feedback polynomial is chosen appropriately (often clock controlled for complexity induction and used in conjunction with jumping to increase the efficiency and reach a CJCSG structure). The CJCSG consists of eight identical sections plus an incomplete ninth section [16]–[18].

The structure of a jump register section includes jump control in (JC_i) and out (JC_o) signals, which are fed into and out of the section. The substitution box is part of this unit which nonlinearly affects the jump control out signal which is used as an input of the following section. Fig. 1 shows the aforementioned sections cascaded nine times to contribute to the key stream of the cipher. As observed in this figure, this accumulated cascade jump control in key stream generation mode combines the outputs of the nine sections to reach to the key stream needed.

As part of its key generation process, Pomaranch uses eight uneven substitution boxes with a 9-bit input and a 7-bit output. Each substitution unit is based on the inverse modulo an irreducible polynomial of degree nine, i.e., $x^9 + x + 1$, whose period is 73. The 9-bit output is then converted into a 7-bit one with deletion of the most significant and least significant bits of the result.

For the hardware implementations of the uneven substitution box of Pomaranch, multiple instances (memories or lookup tables) are needed. In field-programmable gate array (FPGA) platforms, one needs to use block memories or distributed pipelined memories and in ASIC, memory macros or synthesized logic is needed which are not preferable for high-performance and low-complexity applications. Thus, the inverse can be realized in composite fields such that the composite field $GF((2^3)^3)$ is used through which the complexity of the operations needed for realizing the inverse is much reduced.

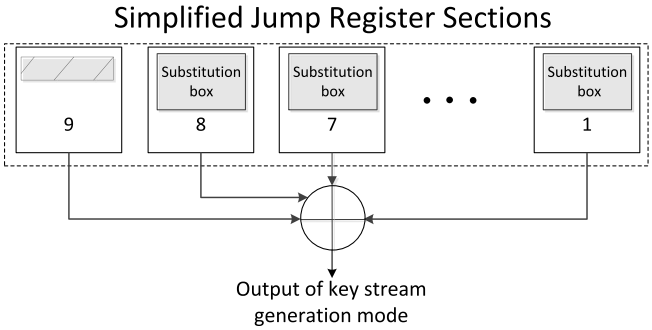

Fig. 1. Simplified accumulated cascade jump control.

The finite field $GF(2^9)$ is represented by elements (in terms of polynomials) of degree eight. The field $GF(2^9)$ can also be represented as $GF((2^3)^3)$, where, here, the elements of this composite field are given as polynomials of degree at most 2 with coefficients from $GF(2^3)$ [19]. We follow the representation in [19] and based on the search performed through the set of primitive polynomials of degree nine over $GF(2)$, it is determined that the polynomial $p(x) = x^9+x^7+x^5+x+1$ is suitable for efficient architectures. Consequently, the polynomials $q(x) = x^3+x+\gamma$ and $r(x) = x^3+x+1$ are used as the tower field polynomials for construction of the composite field operations. One can refer to [19] for detailed information and numeric examples. In this case, $(\alpha^7)^9+\alpha^7+1 = 0$, eventually determining the linear transformation mapping polynomials modulo $x^9 + x + 1$ to polynomials modulo $p(x)$.

In general, time and hardware redundancy are two main methods for fault diagnosis. Hardware redundancy adds hardware to the original structure for diagnosis and time redundancy repeats the operations two times for detection of transient faults. Permanent faults through time redundancy can be detected using various methods which are, generally, denoted as recomputation with encoded operands. The fault diagnosis methods alarm the errors in the architectures; however, even if the overhead is acceptable, there could be a chance for false alarms, i.e., detection of faults that do not result in erroneous outputs. Such false alarms could be exploited to induce distrust to the user, i.e., repetitive, false detections result in either ignoring the alarms by the user or abandoning the devices in which the cryptographic architectures are embedded.

III. Efficient Architectures for the Substitution Box

In this section, we present efficient low-power architectures for the Pomaranch substitution box. Moreover, we perform

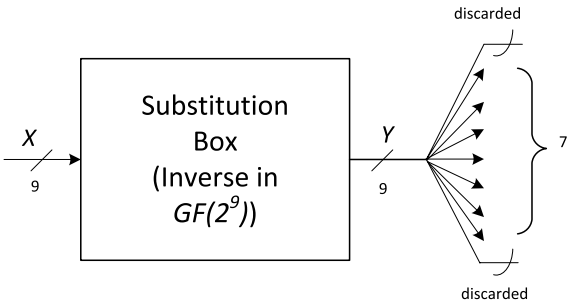

Fig. 2. 9-to-7 substitution box and its uneven structure.

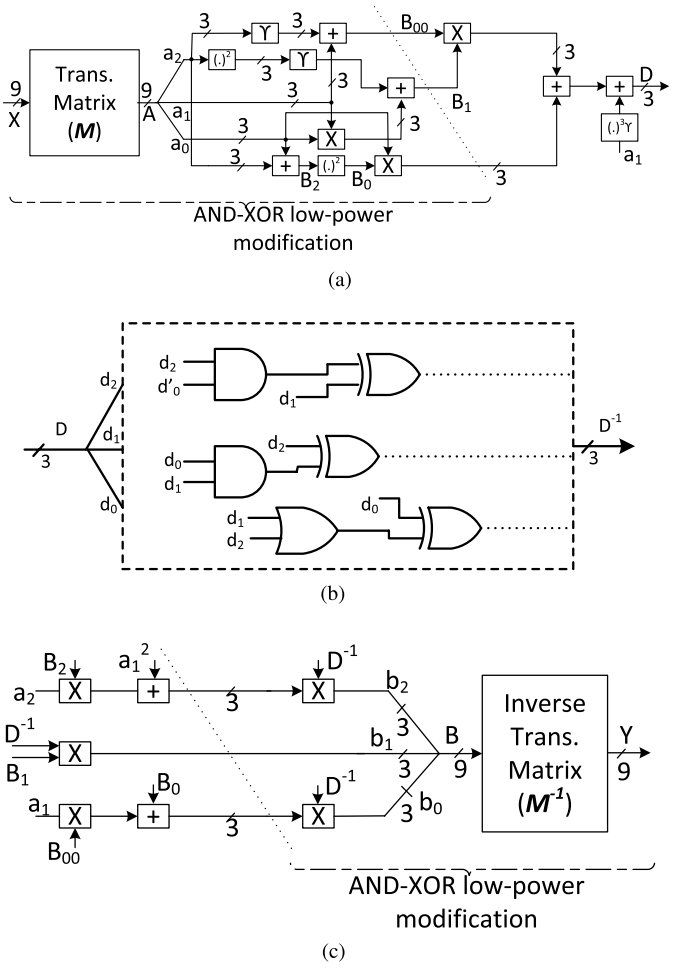

Fig. 3. Architectures for the composite field substitution box and the presented low-power modifications, (a) first subpart, (b) second subpart, and (c) third subpart.

an accurate analysis for power consumption through ASIC implementations and show alleviations, where applicable.

The substitution box is part of a unit in Pomaranch cipher which implements a key-dependent filter function, containing a 9-to-7-bit box and a balanced nonlinear Boolean function of seven variables. The 9-bit output of the substitution box is converted into a 7-bit one with deletion of the most significant and the least significant bits, as shown in Fig. 2.

Composite fields can be utilized to realize the substitution box to achieve low-complexity architectures. The structure of the substitution box using composite fields is shown in Fig. 3. As shown in Fig. 3(a) and (c), a transformation matrix (M) transforms the elements in the binary field to the composite field $GF((2^3)^3)$. Then, the operations are done in composite fields to achieve the inverse which is then retransformed to binary field using an inverse transformation matrix (M^{-1}). Eventually, the two most and least significant bits are discarded to get to the uneven structure of the substitution box of Pomaranch. The resulting transformation matrix M and its inverse M^{-1} are given in [19] (mapping vectors in $GF(2^9)$ defined by x^9+x+1 to vectors in $GF((2^3)^3)$ defined by $p(x)$ and $\gamma = \alpha^{73}$).

The operations used in composite fields include addition, multiplication (including multiplication with constant γ), squaring, cubing, and inversion in $GF(2^3)$. The architecture of the substitution box in Fig. 3 includes a first subpart [Fig. 3(a)] which contains the transformation matrix M whose input is shown by $X \in GF(2^9)$ to get an output of $A \in GF((2^3)^3)$. This 9-bit element is then divided into 3-bit elements denoted by a_2, a_1, a_0 which are then processed to get the output of this subpart, i.e., $D \in GF(2^3)$. In Fig. 3(b) (second subpart), the inversion operation in $GF(2^3)$ is shown which yields to $D^{-1} \in GF(2^3)$. Finally, as shown in Fig. 3(c) (third subpart), $D^{-1} \in GF(2^3)$ is further modified to obtain 3-bit elements denoted by b_2, b_1, b_0 in Fig. 3(c) and eventually $B \in GF((2^3)^3)$, which is then retransformed by the inverse transformation matrix M^{-1} to get the output $Y \in GF(2^9)$ in the binary field, which is discarded eventually to a 7-bit output.

A. Low-Power Architectures

The substitution box occupies much of the area and consumes much of the power in Pomaranch. Based on our ASIC synthesis, the S-boxes occupy around 91% of the top-level key map and roughly 88% of the top-level power is consumed by the S-boxes.

One needs to carefully pinpoint the approaches for realizing this unit so that the eventual architecture is usable for sensitive applications in various constrained smart infrastructures. To reduce the dynamic hazards in the hardware implementations of the substitution box of the Pomaranch stream cipher for low-power designs, one can base the architectures devised on the propagation probability of signal transitions [20]. One observation is that XOR gates can increase the power consumption due to the fact that such logic gates have the probability of signal propagation of one and thus propagating all the hazards, increasing the power consumed.

To achieve more low-power architectures for the substitution box of the Pomaranch stream cipher, we have restructured it so that a two-level logic, i.e., AND–XOR structure, is obtained for subparts one and three of composite field realization of the substitution box, i.e., Fig. 3(a) and (c), respectively. Specifically, in Fig. 3(a), the combined restructured transformation matrix and part of logic gates in composite fields [specified in Fig. 3(a) by the curly bracket] are modified to achieve an AND–XOR structure. Moreover, in Fig. 3(c), the combined restructured inverse transformation matrix and part of logic gates [shown in Fig. 3(c) by the curly bracket] are transformed into an AND–XOR structure for power preservation.

The original composite field structure and the two modified low-power ones [one with only the AND–XOR structure, as shown in Fig. 3(c), and the other with both of the modified architectures in Fig. 3(a) and (c)] are synthesized in ASIC and the area and power consumptions are derived and compared. We note that composite field realization is of paramount benefit for low-complexity architectures compared with memory-macros or synthesized registers on the ASIC platform. Moreover, power preservation will lead to low-energy solutions for sensitive and constrained, battery-powered embedded systems.

The proposed low-power architectures increase the area with the benefit of much decrease in power consumption. Indeed, based on the synthesis results, the power savings are much higher than the induced area for the structures. Specifically, at a typical working frequency, although the composite field architecture is 7% and 24% more area efficient than the proposed architectures, respectively, its power consumption is 19% and 47% higher compared with the proposed low-power structures, respectively (without much difference in the delay and thus frequency and throughput). Specifically, the power consumption corresponding to the original architecture is 14.5 nW, which is reduced to 11.75 nW (at the expense of a 7% increase in area and a saving of 19% in power) and to 7.69 nW (at the expense of a 7% increase in area and a saving of 19% in power).

IV. Proposed Fault Detection Architectures

In this section, we propose fault detection architectures for the substitution box of Pomaranch considering the vulnerability of such structures to false alarms due to their uneven architectures. Specifically, we propose a framework that can be tailored based on the available resources and the reliability objectives to achieve.

A. Fault Diagnosis Approaches

In what follows, fault diagnosis approaches are provided for the architectures presented in Fig. 3. Multiterm signatures are devised and presented as a fault diagnosis framework that can be used depending on the requirements in smart infrastructures in terms of reliability. We carefully pinpoint the false-alarm vulnerability of such approaches and modifications needed to counteract such instances are presented. These are benchmarked in detail in terms of error coverage and efficiency in the following sections.

We first present two theorems that are used in deriving the signatures needed for our fault diagnosis approaches. Based on the structures in Fig. 3(a) and (c), multiplications in composite fields are used frequently to perform operations in the subfield $GF(2^3)$. Moreover, observing Fig. 3(b), the architecture of inversion in $GF(2^3)$ is shown which is used in each substitution box iteration. Accordingly, the following two theorems are presented to derive the predicted parities of these two important operations in the subfield $GF(2^3)$.

Theorem 1: Let $X \in GF(2^3)$ and $Y \in GF(2^3)$ be two elements in composite fields. Let the vectors $X = (x_2, x_1, x_0)$ and $Y = (y_2, y_1, y_0)$ be their respective vectors representing these elements. The predicted parity of $R = X \times Y \in GF(2^3)$, i.e., $\hat{P_R}$, is derived as

$$\hat{P_R} = x_0(y_0 + y_1 + y_2) + x_1(y_0 + y_1) + x_2 y_0 \tag{1}$$

where $+$ denotes the XOR operation (modulo-2 add).

Proof: Considering the formulas for the multiplications in the subfield $GF(2^3)$ as follows:

$$r_0 = x_0y_0 + x_1y_2 + x_2y_1 \tag{2}$$

$$r_1 = x_0y_1 + x_1y_0 + x_1y_2 + x_2y_1 + x_2y_2 \tag{3}$$

$$r_2 = x_0y_2 + x_2y_0 + x_1y_1 + x_2y_2 \tag{4}$$

we can modulo-2 add the coordinates of the respective vectors representing the result, i.e., through $x_0y_0 + x_1y_2 + x_2y_1 + x_0y_1 + x_1y_0 + x_1y_2 + x_2y_1 + x_2y_2 + x_0y_2 + x_2y_0 + x_1y_1 + x_2y_2$, to reach to the predicted parity derived in (1) and the proof is complete. ∎

Theorem 2: Let $X \in GF(2^3)$ be an element in composite fields. Let the vector $X = (x_2, x_1, x_0)$ be its respective vector representing this element. The predicted parity of the inverse element $X^{-1} \in GF(2^3)$, i.e., $\hat{P}_{X^{-1}}$, is derived as

$$\hat{P}_{X^{-1}} = x_0\overline{x}_1 + (x_1 + \overline{x}_0)x_2. \tag{5}$$

Proof: Considering the formulas for the inversion output bits in the subfield $GF(2^3)$, i.e., $x_0 + (x_1 \vee x_2)$, $x_0x_1 + x_2$, and $x_1 + \overline{x}_0x_2$ (where the symbol $\vee$ represents the OR operation), one can perform a modulo-2 addition to reach $\hat{P}_{X^{-1}}$ in (5), which completes the proof. ∎

In addition to the aforementioned predicted parities, as shown in Fig. 3, one needs to derive the predicted parities for a number of other operations in $GF(2^3)$. These are presented through the following theorem.

Theorem 3: Let $X \in GF(2^3)$ be an element in composite fields. Let $\gamma \in GF(2^3)$ be a constant in composite fields as well. The predicted parities for squaring X^2, i.e., $\hat{P}_{X^2}$, cubing, and multiplication by the constant γ, i.e., $\hat{P}_{X^3\gamma}$, and multiplication by the constant γ, i.e., $\hat{P}_{X\gamma}$, are presented below

$$\hat{P}_{X^2} = x_0 + x_1 \tag{6}$$

$$\hat{P}_{X^3\gamma} = x_0x_1 + (x_0 + x_1)\overline{x}_2 \tag{7}$$

$$\hat{P}_{X\gamma} = x_0 + x_1. \tag{8}$$

Proof: Based on the formulas for the squaring output bits in the subfield $GF(2^3)$, i.e., x_0, x_2, and $x_1 + x_2$, one can perform a modulo-2 addition to reach $\hat{P}_{X^2}$ in (6). Considering the equations for the cubing and multiplication by the constant output bits in the subfield $GF(2^3)$, i.e., $x_0x_1 + x_2$, $(x_0 \vee x_2) + x_1x_2$, and $(x_0 \vee x_2) + x_0\overline{x}_2$, one can perform a modulo-2 addition to reach $\hat{P}_{X^3\gamma}$ in (7). Finally, considering the multiplication by the constant γ output bits in the subfield $GF(2^3)$, i.e., x_2, $x_2 + x_0$, and x_1, one can perform a modulo-2 addition to reach $\hat{P}_{X\gamma}$ in (8). This completes the proof. ∎

Remark 1: The hardware complexities for the predicted parities of $R = X \times Y \in GF(2^3)$, i.e., $\hat{P_R}$, in Theorem 1 and $X^{-1} \in GF(2^3)$, i.e., $\hat{P}_{X^{-1}}$, in Theorem 2 in terms of logic gates and considering the same complexities for different gates are seven and six logic gates, respectively.

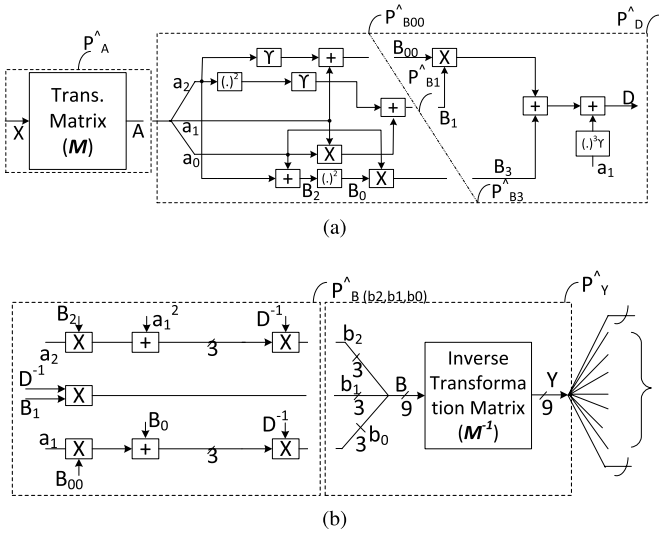

Fig. 4. Predicted signatures of the substitution box for (a) subpart 1 and (b) subpart 3.

Remark 2: The hardware complexities for the predicted parities of squaring X^2, i.e., $\hat{P}_{X^2}$, cubing, and multiplication by the constant γ, i.e., $\hat{P}_{X^3\gamma}$, and multiplication by the constant γ, i.e., $\hat{P}_{X\gamma}$, are one, four, and one logic gate(s), respectively.

We have presented the architectures for different subparts of the substitution box of the Pomaranch stream cipher in Fig. 4. As observed in this figure, different predicted parities within the architectures are shown with the $\hat{P}$ notations. Let us explain in detail how these predicted parities are derived through utilizing Theorems 1–3.

As shown in Fig. 4(a), one can derive the detection signatures for the transformation matrix M whose input is shown by $X \in GF(2^9)$ to get an output of $A \in GF((2^3)^3)$. In this regard, we propose using two different signatures. The first one to derive is the predicted parity of the transformation unit, i.e., $\hat{P}_A$, as shown in Fig. 4(a). This predicted parity is useful for single stuck-at errors as we present in later sections. The second alternate parity is the bit-interleaved parities which are of paramount use in detecting adjacent errors. We present the following theorem and proposition for deriving these predicted parities.

Theorem 4: Let $A \in GF((2^3)^3)$ be an element in composite fields, as shown in Fig. 4. Then, the predicted parity of the transformation matrix, i.e., $\hat{P}_A$, as shown in Fig. 4(a), can be derived as follows based on the bit elements of the input to the unit, i.e., $X \in GF(2^9)$, acting as the row vector $[x_8, \ldots, x_1, x_0]$ multiplied by the transformation matrix to derive the 9-bit row vector representing $A \in GF((2^3)^3)$

$$\begin{aligned}\hat{P}_A &= x_0 + x_1 + x_5 + x_6 + x_7 + x_8 \\ &= P_X + x_2 + x_3 + x_4 \end{aligned} \tag{9}$$

where P_X is the actual parity for the input $X \in GF(2^9)$.

Proof: Considering the transformation matrix presented in the previous section, one can modulo-2 add the rows noting that the bit elements of the input, i.e., $X \in GF(2^9)$, act as the row vector $[x_8, \ldots, x_1, x_0]$ multiplied by the transformation matrix to derive the 9-bit row vector representing $A \in GF((2^3)^3)$. Moreover, noting that $P_X = x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8$, one can derive the predicted parity of the transformation matrix, i.e., $\hat{P}_A$, as shown in Fig. 4(a), and the proof is complete. We note that the latter formulation leads to using only three XOR gates if P_X is available prior to the computations. ■

Proposition 1: The bit-interleaved parities of $A \in GF((2^3)^3)$ (used for detecting burst and adjacent errors) as an element in composite fields can be derived as follows noting that $\hat{P}_{A,0}$ is for even entries (e.g., zeroth, second, fourth, and so on) and $\hat{P}_{A,1}$ is for odd entries (e.g., first, third, fifth, and so on), which are functions of the bit elements of the input to the unit, i.e., $X \in GF(2^9)$

$$\begin{aligned}\hat{P}_{A,0} &= x_0 + x_2 + x_3 + x_4 + x_5 + x_7 \\ &= P_X + x_1 + x_6 + x_8 \end{aligned} \tag{10}$$

$$\begin{aligned}\hat{P}_{A,1} &= x_1 + x_2 + x_4 + x_6 + x_7 + x_8 \\ &= P_X + x_0 + x_3 + x_5 \end{aligned} \tag{11}$$

where P_X is the actual parity for the input $X \in GF(2^9)$.

The predicted signatures for the next four signatures shown in Fig. 4(a) can be derived based on Theorems 1–3. Specifically, for $\hat{P}_{B1}$, $\hat{P}_{B3}$, $\hat{P}_{B00}$, and $\hat{P}_D$, one can use the predicted parities for multiplication and inversion as well as those for the operations in Theorem 3 to obtain the aforementioned predicted parities. We consider $B_{00} = [\theta_2, \theta_1, \theta_0]$ and $B_1 = [\omega_2, \omega_1, \omega_0]$. For the sake of brevity, we just present these as follows using the notations in Fig. 4(a) and the proof is omitted

$$\begin{aligned}\hat{P}_{B1} &= a_0(a_3 + a_4 + a_5) + a_1(a_3 + a_4) + a_2a_5 \\ &\quad + a_8 + a_6 \end{aligned} \tag{12}$$

$$\hat{P}_{B3} = a_0(\overline{a}_7 + a_6) + a_1(a_8 + a_6) + a_2(\overline{a}_7 + a_8) \tag{13}$$

$$\hat{P}_{B00} = a_7 + a_6 + a_5 + a_4 + a_3 \tag{14}$$

$$\begin{aligned}\hat{P}_D &= \omega_0(\theta_0 + \theta_1 + \theta_2) + \omega_1(\theta_0 + \theta_1) + \omega_2\theta_2 \\ &\quad + a_3a_4 + (a_3 + a_4)\overline{a}_5. \end{aligned} \tag{15}$$

The predicted parity of the inversion is already presented through Theorem 2. Therefore, we only need to derive the predicted signatures for the subparts shown in Fig. 4(b). As observed in this figure, we have segmented this architecture through the dotted and dashed lines and the predicted parities are denoted by $\hat{P}_B$ and $\hat{P}_Y$ representing those for the two segments. We would like to point out that depending on the reliability objectives and the overhead tolerated, one can use one to the three predicted parities for the 9-bit output of the composite field operations, i.e., B. These are derived through the presented Theorems 1–3 and are not presented for the sake of brevity.

The inverse transformation matrix converts the elements in composite fields into the corresponding ones in binary field, as shown in Fig. 4(b). Eventually, after discarding two of the 9-bit output, one can obtain the output of the substitution box of Pomaranch which has seven bits. Nonetheless, any devised architecture for the predicted parities of this inverse transformation matrix needs to be carefully obtained to avoid false alarms. In other words, instead of choosing the

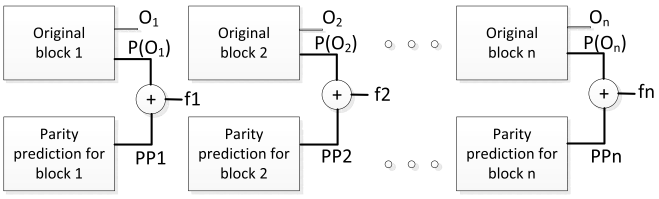

Fig. 5. Parity-based detection mechanism.

9-bit output before discarding to a 7-bit output to obtain the predicted parity (or bit-interleaved signatures as needed), one may consider the final output to avoid detection of the errors at the output which are due to the faults in the substitution box but not affecting the 7-bit final output. These faults show the defects but they do not affect the final output and their detection leads to false-alarms. This is one of the major complications for this very substitution box which has been our motivation in choosing it for our case study. More details are presented in the next section. We finalize this proposed fault diagnosis approach by presenting the predicted parities for the inverse substitution box as follows.

Theorem 5: Let $B \in GF((2^3)^3)$ be an element in composite fields, as shown in Fig. 4(b). Then, the predicted parity of the inverse transformation matrix, i.e., $\hat{P}_Y$, can be derived as follows considering the derived and partly discarded seven-bit row vector representing $Y \in GF(2^9)$

$$\hat{P}_Y = b_3 + b_4 + b_8. \tag{16}$$

Proof: Considering the inverse transformation matrix presented in the previous section, one can modulo-2 add the rows. Finally, one can derive the predicted parity of the inverse transformation matrix, i.e., $\hat{P}_Y$, as shown in Fig. 4(b), and the proof is complete. ■

Proposition 2: The bit-interleaved parities of Y can be derived as follows noting that $\hat{P}_{Y,0}$ is for even entries and $\hat{P}_{Y,1}$ is for odd entries

$$\hat{P}_{Y,0} = b_1 + b_2 + b_4 + b_7 \tag{17}$$

$$\hat{P}_{Y,1} = b_1 + b_2 + b_3 + b_7 + b_8. \tag{18}$$

The merit of these flags is that they alarm the user of the infrastructures-to-become-secure in case of any faults detected. This is in line with the objectives of such infrastructures in terms of fault diagnosis and false alarm resistivity. Specifically, the high coverage of the proposed solutions in this paper will make the crypto-architectures resistant against faults and the false-alarm resistivity ensures that such detections are valid, which is a key point in having reliable solutions.

We finalize the proposed approach by mentioning that the predicted signatures are compared with the actual ones to reach the error indication flags. For illustrating such a scheme, this has been shown in Fig. 5. As observed in this figure, n subblocks with n outputs (shown by O_i, $1 \leq i \leq n$ in Fig. 5) have n actual parities which are derived by modulo-2 addition of output bits (denoted by $P(O_i)$, $1 \leq i \leq n$ in Fig. 5). The derived actual parities for the subblocks are XORed with the derived predicted parities (denoted by PP_i, $1 \leq i \leq n$ in Fig. 5) and the error detection flags (f_i, $1 \leq i \leq n$) are derived which alert about any detected fault in the structure. Finally, for the bit-interleaved solution, similar schemes are used for deriving the flags used for indicating the faults. A minor modification in Fig. 5, i.e., deriving pairs of bit-interleaved actual/predicted parities for the blocks, results in the detection mechanism using such signatures.

B. Diagnosis Method for Key Map Subparts

Here, we explain other subparts of Pomaranch within key map and mention the respective diagnosis methods for them. Nevertheless, the reason for the emphasis on the S-box is its nonlinearity within Pomaranch compared to other subparts. Such S-boxes occupy most of the area and consume much of the power of the Pomaranch.

Nine least significant bits of the section key are XORed (bit wise) with υ. Fault diagnosis of such XOR operation is performed by hardware or time redundancy depending on the objectives, for instance, recomputing with rotated operands can be used to detect both transient and permanent faults through time redundancy. The 7-bit output of the S-box is again XORed with section key. Boolean function F is the last subpart [16]–[18] and takes seven bits and outputs one bit (JC out bit) of the section, denoted by JC_0. This can be treated as a simple 7-to-1 S-box for which known lookup table fault diagnosis techniques can be used, including duplication and hardware/time redundancy.

V. Error Simulations and Benchmark

Single event upsets in the VLSI systems are also of great importance which lead to erroneous outputs. One may also consider multiple faults for which the locations, types, and numbers of the faults are random. Finally, we consider burst faults affecting adjacent bits in the microchips. VLSI defects can introduce such burst faults. One needs to also note that due to the constraints in fault injection technology, instead of the ideal case of single faults, what actually may happen in practice is occurrence of burst faults.

A. Results

Based on the fault diagnosis schemes presented in the previous section, we have performed simulations starting from single stuck-at faults, we have exhaustively injected faults at the output of the subparts shown in Fig. 4 (this includes the outputs of the respective signature-deriving architectures). Specifically, the top-level fault model is the stuck-at fault model for both the error detection scheme and the original architecture. Using such a stuck-at fault model and utilizing the predicted parities presented, we were able to detect all the injected single faults. We emphasize that using bit-interleaved signatures does not affect the fault detection capabilities of the architectures for this fault model; however, they are avoided in this case because of the slightly higher overhead they introduce when applied to the transformation and inverse

transformation matrices. We have performed RTL simulations (not gate-level netlist simulations). The gate-level simulation with error model can increase the number of unnoticed errors (compared with RTL error simulation) due to nonlinear functionality in substitution boxes.

The second fault model, i.e., multiple faults, has been used and the faults were injected uniformly with random locations, numbers, and types of faults. Using the predicted parities for the aforementioned schemes (not the bit-interleaved ones), more than 99% of the faults were detected. The proposed methodology alleviates the problem of faults but does not make the architectures fully reliable. The experiments were based on injecting 10 000 stuck-at zero and stuck-at one faults and monitoring the cases for which we get erroneous outputs, i.e., eliminating the cases for which the faults are masked. We note that considering eight substitution boxes of Pomaranch, this would be very close to 100%. Multiple faults and, as we cover next, burst faults are also much important when it comes to the practicality of such injections and the probability of occurrence.

Finally, although bit-interleaved signatures slightly increase the hardware complexities of fault diagnosis structures, they are used when adjacent burst faults are of concern. Based on the bit-interleaved signatures obtained in the previous section and considering random locations and types for injected faults, in addition to detecting an odd number of faults, we were able to detect all the two-bit stuck-at zero or stuck-at one burst faults. Two notes need to be considered with respect to these signatures; first, one may use bit-interleaved predicted parities not only for the transformation matrices but also for the inner subparts and second, all the faults of odd number are detected and for the even number, e.g., four, if more than two adjacent bits are of concern, one may use the proposed framework by slightly modifying the formulas to reach, e.g., three-bit, interleaved predicted signatures.

The objective of the proposed approaches in this paper is to provide fault immunity and reliability to smart infrastructures. This framework can be used in conjunction with the presented low-power architectures to provide secure and usable infrastructures for emerging sensitive usage models. The suitability of the proposed solutions stems from their high error coverage (which is crucial in smart infrastructures using cryptographic hardware and embedded systems), as well as their acceptable overhead.

We would also like to present the objectives of the proposed approaches in terms of false-alarm resistance as well as their suitability for smart infrastructures. First, it is emphasized that the false-alarm immunity of crypto-systems also determines the immunity against inducing distrust to the user. Such malicious intents might try to divert the fault diagnosis stream so that without having errors at the output of the crypto-architectures used in smart infrastructures, alarms get falsely initiated, which would eventually cause abandoning the entire system. In short, protecting against such cases would result in reliable and false-alarm immune smart infrastructures that are trustworthy and can be used safely for different usage models.

B. False Alarms

False alarms could have adverse effects on the utilization of cryptographic solutions. Specifically, if such alarms get repetitive, they might hinder the normal operations of cryptographic algorithms by inducing distrust to the user, who may eventually abandon the entire solution.

Let us separate the false-alarm vulnerabilities of such crypto-systems into two streams. The first one deals with those false alarms that are the results of the single stuck-at fault model, which is a probable case with respect to natural faults. We would like to emphasize that for the inverse transformation matrix, the 7-bit output needs to be considered to avoid possible false-alarms. Indeed, through simulations, around 22% false-alarms were observed if this is not carefully followed. This is a clear distinction compared with the substitution boxes with the same number of inputs and outputs (another instance is the uneven 6-to-4 S-box of Data Encryption Standard).

Now, let us consider the cases in which the fault model deals with multiple, random stuck-at faults. For such cases, a number of causes may result in having false-alarms in crypto-systems. Because we are dealing with multiple signatures, there might be cases in which we detect faults in an inner subpart which will not be eventually translated into errors in the 7-bit output. This might be due to the masking of such faults or due to the occurrence of such faults only in the most and the least significant bits of the output to be discarded. We emphasize that such a case is due to fault diagnosis approaches in which error detection is expected to reveal the error at the output of the functions; yet, the diagnosis method alarms also the faults affecting the middle subparts which are masked at the output. This second case is due to fault detection methods which could affect any general S-box architecture such as that of the AES (and in general crypto-architectures beyond S-boxes) and is not confined to the uneven S-boxes. If the tower field fault detection architectures of the composite-field S-boxes within the AES (or other transformations within) use multiple signatures for the subparts, for instance, false-alarms could also affect the detection schemes.

Through simulations, considering different numbers of stuck-at zero and one cases, we have identified such cases (the former is a general case among the substitution boxes whereas the latter is specific to this uneven box). The results are shown in Table I. As observed in this table, different number of faults is injected for two types and the number of masked and false alarms are shown. We would like to point out that these presented percentages are higher unless, as used here, we utilize just the (bit interleaved) signature(s) for the inverse transformation matrix considering the 7-bit output. Finally, we note that these false-alarms show that there exists natural defect(s) in the architectures; nevertheless, these do not result in erroneous outputs for that particular simulation instance.

VI. ASIC Implementations

This section presents the results of our ASIC syntheses performed for the original and the error detection structures

TABLE I
FALSE-ALARM ASSESSMENTS FOR MULTIPLE, RANDOM FAULTS IN ONE POMARANCH SUBSTITUTION BOX

Type of faults	Injected faults[1]	Detected faults[2]	Masked faults[3]	Affecting discarded bits only	False alarms	Percent false-alarms
Stuck-at zero	1,000	991 (99.1%)	6	3	9	0.90%
	10,000	9,920 (99.2%)	62	36	98	0.98%
	20,000	19,897 (99.5%)	81	71	152	0.76%
	25,000	24,913 (99.7%)	136	102	238	0.95%
Stuck-at one	1,000	987 (98.7%)	4	4	8	0.8%
	10,000	9,912 (99.1%)	78	21	99	0.99%
	20,000	19,910 (99.6%)	109	73	182	0.91%
	25,000	24,874 (99.5%)	113	119	232	0.93%

1. The number of random, multiple faults injected in different sub-parts of the substitution box.
2. Using the presented formulations for multiple signatures.
3. These are detected by at least one of the signatures but they are not translated into errors eventually.

TABLE II
ASIC SYNTHESES AND OVERHEAD ASSESSMENTS FOR THE SUBSTITUTION BOX OF POMARANCH

Architecture	Area		Frequency [MHz]	Throughput [Gbps]
	[μm^2]	GE		
Original	339.8	242	809	5.66
Signature-based scheme	402.7 (18.6%)	286	703	4.92 (13.0%)
Bit-interleaved scheme	411.5 (21.1%)	292	711	4.98 (12.0%)

of the substitution box of Pomaranch algorithm to benchmark the overheads as the result of the added structures. We note that we have chosen the ASIC platform based on the resources available to us; because our presented schemes are not dependent on the hardware platform, similar overheads are expected for FPGAs. Through the performed ASIC syntheses, the overheads in terms of hardware and timing are derived. We have used the TSMC 65-nm standard-cell library [21] in the Synopsys Design Compiler [22].

We have presented the results of our syntheses in Table II. in Table I, for the original algorithm, i.e., the substitution box of Pomaranch in composite fields, and for the proposed error detection schemes (both the regular and the bit-interleaved signatures), the areas (in terms of μm^2), maximum working frequencies (in terms of MHz), and throughputs (in terms of gigabits per second) have been shown. To make the area results transferable when switching technologies, we have provided the NAND-gate equivalency (in terms of gate equivalent, denoted by Gate Equivalent). This is performed using the area of a NAND gate in the utilized library, which is 1.41 μm^2. Furthermore, the area and throughput degradations are presented in parentheses to benchmark the proposed error detection schemes.

As shown in Table II, the hardware complexity (area) of the original substitution box of Pomaranch is 339.8 μm^2, whereas, for the signature-based and bit-interleaved architectures, the areas are 402.7 μm^2 and 411.5 μm^2 (leading to overheads of 18.6% and 21.1%), respectively. These hardware overheads need to be tolerated to achieve fault detection architectures for reliable usage models. We have also derived the area overheads for four different cases, i.e., for the two proposed schemes (Table II) and considering the effects on the entire top-level with detection for the S-boxes as well as the worst case scenario of duplication for other subparts. For the former case, the area overheads are 16.8% and 19.5%, respectively, and for the latter, they are 20.4% and 23.1% (for the signature-based and bit-interleaved schemes). The maximum frequencies in which these architectures can work are 809, 703, and 711 MHz, respectively, leading to the throughputs of 5.66, 4.92, and 4.98 Gb/s, respectively. These lead to throughput degradations of 13% (for the signature-based approach) and 12% (for the bit-interleaved scheme). We would like to point out that based on the reliability requirements, performance and implementation metrics to achieve, and resources available (overheads tolerated) one can use the presented framework to achieve fault detection and false-alarm-aware architectures.

VII. CONCLUSION

In this paper, reliability and false-alarm sensitivity of sensitive cryptographic applications are benchmarked through a case study, i.e., the uneven substitution box of a stream cipher, to elaborate on the respective effects on smart infrastructures. We have presented low-power architectures for this stream cipher and then proposed a framework to provide fault immunity for infrastructures that need to deal with sensitive information and are smart and ubiquitous. The proposed architectures are benchmarked in terms of error coverage for different fault models and assessed for false-alarm immunity. Moreover, they have been synthesized on an ASIC platform and it is shown that with an acceptable overhead, high error coverage can be achieved for the proposed architectures. Furthermore, we have assessed the benefits and effects of such architectures for smart infrastructures. The benchmark details the smart infrastructure implications and elaborates on the fact that using the proposed framework, smart infrastructures can be more efficiently and reliably utilized.

REFERENCES

[1] K. Fu and J. Blum, "Controlling for cybersecurity risks of medical device software," *Commun. ACM*, vol. 56, no. 10, pp. 35–37, Oct. 2013.

[2] D. Halperin, T. Kohno, T. S. Heydt-Benjamin, K. Fu, and W. H. Maisel, "Security and privacy for implantable medical devices," *IEEE Pervasive Comput.*, vol. 7, no. 1, pp. 30–39, Jan./Mar. 2008.

[3] M. Rostami, W. Burleson, A. Jules, and F. Koushanfar, "Balancing security and utility in medical devices?" in *Proc. 50th ACM/EDAC/IEEE Int. Conf. Design Autom.*, May/Jun. 2013, pp. 1–6.

[4] M. Zhang, A. Raghunathan, and N. K. Jha, "Trustworthiness of medical devices and body area networks," *Proc. IEEE*, vol. 102, no. 8, pp. 1174–1188, Aug. 2014.

[5] H. Khurana, M. Hadley, N. Lu, and D. A. Frincke, "Smart-grid security issues," *IEEE Security Privacy*, vol. 8, no. 1, pp. 81–85, Jan./Feb. 2010.

[6] M. Mozaffari-Kermani, M. Zhang, A. Raghunathan, and N. K. Jha, "Emerging frontiers in embedded security," in *Proc. 26th Int. Conf. VLSI Design*, Jan. 2013, pp. 203–208.

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

[8] T. H.-J. Kim, L. Bauer, J. Newsome, A. Perrig, and J. Walker, "Challenges in access right assignment for secure home networks," in *Proc. USENIX Conf. Hot Topics Secur.*, 2010, pp. 1–6.

[9] 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.

[10] 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.

[11] 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 Integr. (VLSI) Syst.*, vol. 19, no. 1, pp. 85–91, Jan. 2011.

[12] A. Satoh, T. Sugawara, N. Homma, and T. Aoki, "High-performance concurrent error detection scheme for AES hardware," in *Proc. 10th Int. Workshop CHES*, Aug. 2008, pp. 100–112.

[13] 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.

[14] X. Guo and R. Karri, "Recomputing with permuted operands: A concurrent error detection approach," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 10, pp. 1595–1608, Oct. 2013.

[15] M. Mozaffari-Kermani and R. Azarderakhsh, "Efficient fault diagnosis schemes for reliable lightweight cryptographic ISO/IEC standard CLEFIA benchmarked on ASIC and FPGA," *IEEE Trans. Ind. Electron.*, vol. 60, no. 12, pp. 5925–5932, Dec. 2013.

[16] C. J. A. Jansen, T. Helleseth, and A. Kholosha, "Cascade jump controlled sequence generator (CJCSG)," in *Proc. Workshop Symmetric Key Encryption*, 2005, pp. 1–16. [Online]. Available: http://www.ecrypt.eu.org/stream/ciphers/pomaranch/pomaranch.pdf

[17] C. J. A. Jansen, T. Helleseth, and A. Kholosha, "Cascade jump controlled sequence generator and Pomaranch stream cipher (version 3)," Dept. Informat., Univ. Bergen, Bergen, Norway, Tech. Rep. 2006/006, 2006. [Online]. Available: http://www.ecrypt.eu.org/stream/papers.html

[18] C. J. A. Jansen, T. Helleseth, and A. Kholosha, "Cascade jump controlled sequence generator and Pomaranch stream cipher," in *Proc. eSTREAM Finalists*, 2008, pp. 224–243.

[19] C. J. A. Jansen, A. Kholosha, and T. Helleseth, "A lightweight implementation of the Pomaranch S-box," in *Proc. eSTREAM*, 2007, pp. 1–6.

[20] S. Morioka and A. Satoh, "An optimized S-box circuit architecture for low power AES design," in *Proc. CHES*, 2003, pp. 172–186.

[21] Taiwan Semiconductor Manufacturing Company. *TSMC Standard-Cell Library*. [Online]. Available: http://www.tsmc.com/, accessed Jan. 2015.

[22] Synopsys, Inc. *Synopsys Design Compiler*. [Online]. Available: http://www.synopsys.com/, accessed Jan. 2015.

Mehran Mozaffari-Kermani (M'11) 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, University of Western Ontario, London, ON, Canada, in 2007 and 2011, respectively.

He joined Advanced Micro Devices, Markham, ON, Canada, as a Senior ASIC/Layout Designer, integrating sophisticated security/cryptographic capabilities into a single accelerated processing unit. He joined the Department of Electrical Engineering, Princeton University, Princeton, NJ, USA, as a Natural Sciences and Engineering Research Council of Canada (NSERC) Post-Doctoral Research Fellow, in 2012. He is currently with the Department of Electrical and Microelectronic Engineering, Rochester Institute of Technology, Rochester, NY, USA. His current research interests include emerging security/privacy measures for deeply embedded systems, cryptographic hardware systems, fault diagnosis and tolerance in cryptographic hardware, VLSI reliability, and low-power secure, and efficient field-programmable gate array and application-specified integrated circuit designs.

Dr. Mozaffari-Kermani was a recipient of the prestigious NSERC Post-Doctoral Research Fellowship. He is a Guest Editor of the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING for the special issue of Emerging Security Trends for Deeply-Embedded Computing Systems from 2014 to 2015. He serves as the Technical Committee Member for a number of security/reliability conferences, including the International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, the Fault Diagnosis and Tolerance in Cryptography Conference, the Conference on RFID Security, the International Workshop on Lightweight Cryptography for Security and Privacy, and the International Workshop on the Arithmetic of Finite Fields. His research is funded through the Texas Instruments Faculty Award (Douglas Harvey) from 2014 to 2015.

Reza Azarderakhsh (M'12) received the B.Sc. degree in electrical and electronic engineering and the M.Sc. degree in computer engineering from the Sharif University of Technology, Tehran, Iran, in 2002 and 2005, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Western Ontario, London, ON, Canada, in 2011.

He joined the Department of Electrical and Computer Engineering, University of Western Ontario, as a Limited Duties Instructor, in 2011. He has been a Natural Sciences and Engineering Research Council of Canada (NSERC) Post-Doctoral Research Fellow with the Center for Applied Cryptographic Research and the Department of Combinatorics and Optimization, University of Waterloo, Waterloo, ON, Canada. He is currently with the Department of Computer Engineering, Rochester Institute of Technology, Rochester, NY, USA. His current research interests include finite field and its application, elliptic curve cryptography, and pairing-based cryptography.

Prof. Azarderakhsh was a recipient of the prestigious NSERC Post-Doctoral Research Fellowship in 2012.

Anita Aghaie received the B.S. degree in electrical and computer engineering from the Isfahan University of Technology, Isfahan, Iran, in 2013. She is currently pursuing the Ph.D. degree with the Rochester Institute of Technology, Rochester, NY, USA, under the co-supervision of Prof. M. Mozaffari-Kermani and Prof. R. Azarderakhsh.

Her current research interests include fault diagnosis and tolerance in digital systems, field-programmable gate array, and VLSI design.