# Fault Diagnosis Schemes for Secure Lightweight Cryptographic Block Cipher RECTANGLE Benchmarked on FPGA

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

***Abstract*—The security and reliability of cryptosystems are endangered with natural occurring and malicious injected faults by leakage of the information. Efficient trade off among minimum performance and implementation metrics and high level of security for cryptosystems in constrained applications has led to proposed error detection schemes for lightweight block ciphers. These ciphers provide low-cost confidentiality in terms of low hardware complexity and fast implementation. In this paper, we propose fault diagnosis schemes for an efficient lightweight block cipher, RECTANGLE, to ensure high level of security with low hardware overhead incorporated for error detection. This cipher offers efficient performance in both hardware and software implementation using bit-slice techniques. To the best of authors' knowledge, no prior error detection scheme has been presented in literature for RECTANGLE to date. The proposed error detection schemes that are provided for the S-box layer, P-layer, and for the round structures with 80-bit or 128-bit key sizes, are benchmarked on field-programmable gate array (FPGA) hardware platform to assess their suitability. The error coverage of these schemes is close to 100% (assessed with fault injection simulation) and the induced overheads are low, increasing the reliability of the hardware architectures of RECTANGLE.**

***Index Terms*—Bit-slice lightweight block cipher, field-programmable gate array (FPGA), reliability.**

## I. Introduction

Security properties with low area and low energy consumption are satisfied with lightweight cryptography for sensitive applications (including smart cards, implantable and wearable medical devices, and wireless nano-sensors) [1], [2]. Cryptographic entities need to have small area (for hardware platforms) and small code size (for software aspect), see, for instance, various architectures of lightweight block ciphers such as LED [3], LBlock [4], PRINCE [5], SIMON and SPECK [6], and PRESENT [7].

Efficient software implementations accompanied by acceptable hardware metrics are required for lightweight block ciphers. In recent past, lightweight ciphers, such as LED, have been claimed to reasonably answer this criterion. However, the recent proposed cipher, RECTANGLE, through using the bit-slice technique, was designed to speed up the performance and complexity for constrained applications [8], [9], [10]. This lightweight block cipher which has a S-layer and P-layer structure, entails two parts, data processing and key scheduling with key sizes of 80 or 128 bits modules. The plaintext input, the intermediate result, and the final ciphertext output have 64 bits in length.

Error detection for deliberate fault injection assists to boost the security level of cryptosystems in both hardware and software implementations [11], [12], [13], [14], [15], [16], [17]. However, the significance of applying lightweight fault diagnosis schemes with low overheads is more prominent in lightweight block ciphers, such as RECTANGLE, in sensitive applications. In this paper, we propose error detection schemes for linear and non-linear sub-blocks of RECTANGLE such as the S-boxes and the key schedule unit. It is noted that there is no prior work on error detection of this bit-sliced lightweight cipher.

In addition, we propose using signature-based and recomputing with rotated operands (RERO) schemes with high error coverage, based on the objectives of hardware reliability requirements and overhead tolerance. Furthermore, through error simulations and hardware implementations on field-programmable gate array (FPGA) using Xilinx Virtex-7 family, it is shown that the presented architectures are acceptable for reliable, resource-constrained applications.

## II. Proposed Error Detection Schemes

In this section, the error detection approaches of the RECTANGLE encryption are proposed. In what follows, the 16-bit words are described in two-dimensional form for the sake of simplicity, i.e.,

$$2D - Form \rightarrow \begin{pmatrix} a_{0,15} & \cdots & a_{0,2} & a_{0,1} & a_{0,0} \\ a_{1,15} & \cdots & a_{1,2} & a_{1,1} & a_{1,0} \\ a_{2,15} & \cdots & a_{2,2} & a_{2,1} & a_{2,0} \\ a_{3,15} & \cdots & a_{3,2} & a_{3,1} & a_{3,0} \end{pmatrix}.$$

The first part of RECTANGLE, data processing, consists of three operations in each round including AddRoundKey, SubColumn, and ShiftRow. Moreover, the last 25th round has just the AddRoundKey operation to derive the final ciphertext.

### *A. Fault Diagnosis of AddRoundKey and ShiftRow*

AddRoundKey consists of modulo-2 addition of the round subkey to the intermediate results in each round. Moreover,

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

R. Azarderakhsh is with the Department of Computer and Electrical Engineering and Computer Science and is an I-SENSE Fellow, Florida Atlantic University, Boca Raton, FL, USA (e-mail: razarderakhsh@fau.edu).

978-1-5090-6113-6/16/$31.00 ©2016 IEEE

Table I
PARITY BIT OF 4-BIT S-BOX (SUBCOLUMN OF RECTANGLE)

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $S$[x] | 6 (0) | 5 (0) | C (0) | A (0) | 1 (1) | E (1) | 7 (1) | 9 (0) | B (1) | 0 (0) | 3 (0) | D (1) | 8 (1) | F (0) | 4 (1) | 2 (1) |

in ShiftRow, each row of the $4 \times 16$ array is left-rotated with different offsets. For instance, Row 0, $(a_{0,15}...a_{0,1}, a_{0,0})$ has no rotation with 0 offset; however, other rows including Row 1, $(a_{1,15}...a_{1,1}, a_{1,0})$, has 1-bit left rotation, Row 2, $(a_{2,15}...a_{2,1}, a_{2,0})$, is left-rotated with 12 bits, and Row 3, $(a_{3,15}...a_{3,1}, a_{3,0})$, has 13 bits of left rotation.

In this paper, signature-based error detection scheme has been applied for different sub-blocks within RECTANGLE. In this section, we investigate error detection schemes for the first operation of the RECTANGLE encryption, AddRoundKey, that consists of modulo-2 addition of the cipher state in each round with the $i$-th round subkey $K_i$. The proposed scheme provides the signature of output in each round by using the signature of inputs, i.e., modulo-2 addition of cipher state with round subkey in each round. If we denote the output of AddRoundKey as $O$ and inputs (the intermediate results in data processing) as $CS_i$ and $K_i$, $0 \leq i \leq 24$; we have the signatures, denoted as $Sig.$, as follows:

$$\hat{Sig.}_{(O)} = Sig.(CS_i) \oplus Sig.(K_i). \tag{1}$$

Examples of using the above signature is parity and interleaved parity signatures.

In the ShiftRow operation, each 16-bit word of 4-row cipher state has the left rotation of 0, 1, 12, and 13 bits. The signature derivation in this operation would be straightforward, e.g., input's parity and output's parity are equal.

### *B. Proposed Schemes for SubColumn (the S-box Variants)*

In this section, error detection schemes for the nonlinear S-box in the round function of the RECTANGLE encryption are proposed. SubColumn applies parallel S-boxes for each of the 4-bit columns of the two-dimensional matrix in each round (the 4-bit input column of this matrix, as $a_{3,j}a_{2,j}a_{1,j}a_{0,j}$, where $0 \leq j \leq 15$, is transformed to 4-bit output of $b_{3,j}b_{2,j}b_{1,j}b_{0,j}$ by using 4-bit S-boxes $S : F_2^4 \longrightarrow F_2^4$).

Two approaches can be used for applying S-boxes in the hardware implementation of RECTANGLE, i.e., LUT-based and logic gate-based. The former one has high performance but high area and power consumption. On the other hand, the latter one typically has less area and power consumption. Our proposed schemes for the S-boxes are based on signatures but are not confined to a special signature; we describe two examples of parity-based and interleaved parity-based approaches.

*Examples (case studies).* We present the predicted (interleaved) parities and the actual parities for the S-box to compare with each other to detect errors. The parity-based scheme is based on deriving the predicted (interleaved) parities of the S-boxes using LUTs as shown in Tables I and II. For each 4-bit element of inputs in the S-box, we modulo-2 add all bits (or derive the 2-bit interleaved parities for burst faults or adjacent multiple faults), seen in these two tables in parentheses.

The latter architecture, logic-based approach, for which we have derived the formula for the S-box $S[x]$ is presented here. Suppose, the input to the 4-bit S-box is $(a_0,\, a_1,\, a_2,\, a_3)$ and the 4-bit output is $(b_0,\, b_1,\, b_2,\, b_3)$, noting that $\vee$ represents an OR gate, we have $b_0 = \bar{a_0}a_2\bar{a_3} \vee \bar{a_0}\bar{a_2}a_3 \vee a_1a_2\bar{a_3} \vee a_1\bar{a_2}a_3 \vee a_0\bar{a_1}a_2a_3 \vee a_0\bar{a_1}\bar{a_2}\bar{a_3}$, $b_1 = a_0a_2a_3 \vee a_0\bar{a_1}a_2 \vee \bar{a_0}a_1a_2\bar{a_3}$, $b_2 = \bar{a_0}\bar{a_2}\bar{a_3} \vee \bar{a_1}\bar{a_2}\bar{a_3} \vee \bar{a_0}a_1a_2 \vee a_0\bar{a_1}a_2 \vee a_0a_1\bar{a_2}a_3$, and $b_3 = a_0a_1a_2\bar{a_3} \vee \bar{a_0}\bar{a_1}\bar{a_2}a_3 \vee \bar{a_0}a_1\bar{a_2}\bar{a_3} \vee a_0\bar{a_1}a_2a_3 \vee a_0\bar{a_1}\bar{a_2}\bar{a_3}$.

Here are two examples for parity-based and interleaved parity-based structures.

*Example 1.* We have derived the predicted parity formulation as follows. A "hat" sign represents the predicted parity, e.g., $\hat{P}$.

$$\hat{P_B} = \bar{a_1}\bar{a_2}\bar{a_3} \vee \bar{a_0}a_2\bar{a_3} \vee \bar{a_0}a_1a_3 \vee a_1a_2a_3. \tag{2}$$

*Example 2.* Modulo-2 addition of odd bits and even bits together for the S-box derives the interleaved parities.

$$\hat{P_B^{(0)}} = \bar{a_0}\bar{a_2} \vee \bar{a_0}\bar{a_1}\bar{a_3} \vee \bar{a_0}a_1a_3 \vee a_0a_2\bar{a_3}, \tag{3}$$

$$\hat{P_B^{(1)}} = \bar{a_0}a_1\bar{a_3} \vee a_0\bar{a_1}\bar{a_3} \vee a_0a_1a_2 \vee \bar{a_0}\bar{a_1}\bar{a_2}a_3. \tag{4}$$

### *C. Proposed Schemes for Key Schedule*

The latter part of this cipher is key schedule which is performed with two size of keys 80 or 128 bits. For instance, for the 80-bit key, $(k_{79}...k_1, k_0)$, the key is divided to $5 \times 16$ array of bits similar to the cipher state. In each round, the 64-bit round subkey $K_i$, where $0 \leq i \leq 24$, constructs the first 4 rows of the key matrix as $K_i = k_{3,j}k_{2,j}k_{1,j}k_{0,j}$. The key is updated with 4-bit S-boxes first. Then, a one round Feistel transformation is applied. For example, in the 80-bit key schedule, $Row_0^{'} := (Row_0 \lll 8) \oplus Row_1$, $Row_1^{'} := Row_2$, $Row_2^{'} := Row_3$, $Row_3^{'} := (Row_3 \lll 12) \oplus Row_4$, $Row_4^{'} := Row_0$, where $\lll$ denotes left rotation. Then, the last operation in key schedule is adding a 5-bit round constant $RC[i]$, which is modulo-2 added with the 5-bit key state, i.e., $k_{4,j}^{'}k_{3,j}^{'}k_{2,j}^{'}k_{1,j}^{'}k_{0,j}^{'} := k_{4,j}k_{3,j}k_{2,j}k_{1,j}k_{0,j} \oplus RC[i]$.

For the sake of brevity, we present the error detection for 80-bit seed key; nonetheless, the proposed schemes are applicable to 128-bit key size.

*First-Step Scheme:* Each 80-bit seed key, which is firstly saved in an 80-bit key register and is divided to five 16-bit words, gets modified by applying the S-box for the 4 uppermost rows and the rightmost columns.

*Second-Step Scheme:* In this step, we present two error detection schemes as signature-based scheme and a recomputing approach. The signature of input row, $Row_i$ (for $0 \leq i \leq 24$), is equal to the signature of output row, $Row_i^{'}$. For instance, the parity-based scheme has high error coverage for odd bits which is efficient for some reliability-constrained applications.

Table II
INTERLEAVED PARITY OF THE 4-BIT S-BOX FOR BURST FAULTS

| x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $S$[x] | 6 (11) | 5 (00) | C (11) | A (00) | 1 (01) | E (01) | 7 (10) | 9 (11) | B (01) | 0 (00) | 3 (11) | D (10) | 8 (10) | F (00) | 4 (01) | 2 (10) |

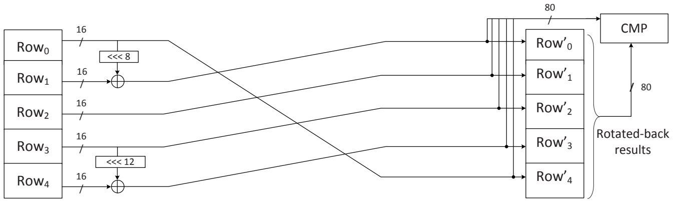

Figure 1. The proposed scheme for the Feistel transformation.

We embed the latter scheme for this step, using modulo-2 addition and left rotation. The proposed RERO scheme, as shown in Fig. 1, applies time redundancy for modulo-2 addition and rotation operations in this step. The first time, the original one which entails five 16-bit words is stored in register $CMP$ in Fig. 1 to compare with rotated back results.If both of the original and back-rotated results are not equal, error has occurred.

*Third-Step Scheme:* The proposed signature-based scheme detects errors in the round constants part as well. Here, we derive one signature for the round key $K_i$ based on modulo-2 addition of the five LSB bits in the first row in the 80-bit keystate matrix considering the round constants. Therefore, we have intact or inverted elements of input key, noting $0 \leq i \leq 24$, $Sig.(K_i) = \sum_{j=0}^{4} k_{0,j} \oplus \sum_{j=0}^{4} RC[i]_{,j}$.

For instance, for the two round constants $RC[0] = 0x01$ and $RC[24] = 0x1D$ in this scheme, we have modulo-2 addition of all bits as '1' and '0', respectively. Therefore, for $K_0$, the input signature is inverted; while, for $K_{24}$, it is intact.

We finalize this section through presenting the overall structures of the presented error detection schemes for the encryption of RECTANGLE-80 which consists of 25 rounds with 64-bit input as depicted in Fig. 2. Each round function of encryption has three operations along with key schedule, for which, we propose fault diagnosis schemes. As seen in Fig. 2, we have shown the respective subsections such as S-layer and P-layer (ShiftRow) in which we have proposed the signature-based schemes in the aforementioned sections. Moreover, the predicted signature-based or RERO schemes are proposed for each of the three parts of key generation process for 25 rounds as shown in Fig. 2.

## III. ERROR SIMULATION AND FPGA BENCHMARK

Throughout this paper, transient and permanent faults as well as including single and multiple stuck-at faults have been considered. The single-bit errors occurring in the cipherstate (the output of each component) are detected by the presented signature-based error detection approaches. Due to the full error coverage of the mentioned schemes for these single stuck-at faults as 100%; we do not simulate them. With respect to fault analysis attacks and technological constraints, flipping exact bits by attackers may not be so applicable to get more information. Therefore, multiple stuck-at faults are simulated in this paper. We use the RECTANGLE cipher with 80-bit key size as reference for our fault injection simulations. In these simulations, error indication flags are counted in RECTANGLE-80 as error representatives through the fault injection experiments, in which, 10,000 faults are injected by an LFSR-based architecture. The results show that the proposed signature-based schemes for data processing part and also for key schedule part are capable of detecting stuck-at faults with very high error coverage (all the cases have at least 9,990 faults detected out of 10,000 samples, i.e., 99.90% detection rate for this case). These comply with the theoretical results using 16 error indication flags in each round. The considered error model (single or multiple stuck-at faults) can be considered as the super-set for differential fault analysis (DFA) of RECTANGLE. It is noted that we do not claim all DFA attacks are detected by the proposed schemes; nevertheless, they make such attacks more difficult to mount.

The results of our FPGA overhead assessment for the proposed schemes are presented in this section as shown in Table III. The ISE version 14.7 and Virtex-7 FPGA family (device: xc7vx330t) with VHDL as design entry has been used for the original and the error detection structures using LUT-based S-boxes structure. As shown in Table III, the area, delay, and power consumption overheads of the proposed signature-based schemes are 8%, 14.2%, and 1.13%, respectively. The degradations for throughput and efficiency in the mentioned architecture are 12.5 % and 19%, respectively. We anticipate similar overheads for application-specific integrated circuit (ASIC) implementations as well as other FPGA families because our proposed schemes are platform oblivious. With the error coverage of close to 100% and the aforementioned acceptable overheads, the presented schemes are suitable for making the hardware implementations of RECTANGLE more reliable.

## IV. CONCLUSIONS

Through the approaches presented in this paper, reliability of the bit-slice lightweight block cipher, RECTANGLE, has been increased. We have utilized signature-based schemes to provide high error coverage with low overheads for both LUT-based and logic gate-based structures of the S-boxes (S-layer) and P-layer in RECTANGLE-80. These architectures are implemented on Xilinx FPGAs and the derived overheads are acceptable. This variety of designing the error detection architectures for the S-boxes gives the designers more freedom to allocate proper resources to achieve better performance. Furthermore, for the key schedule module, not only have we developed the presented signature-based schemes but also RERO schemes have been utilized to protect the architectures with low-overhead. The merit of the proposed schemes is that

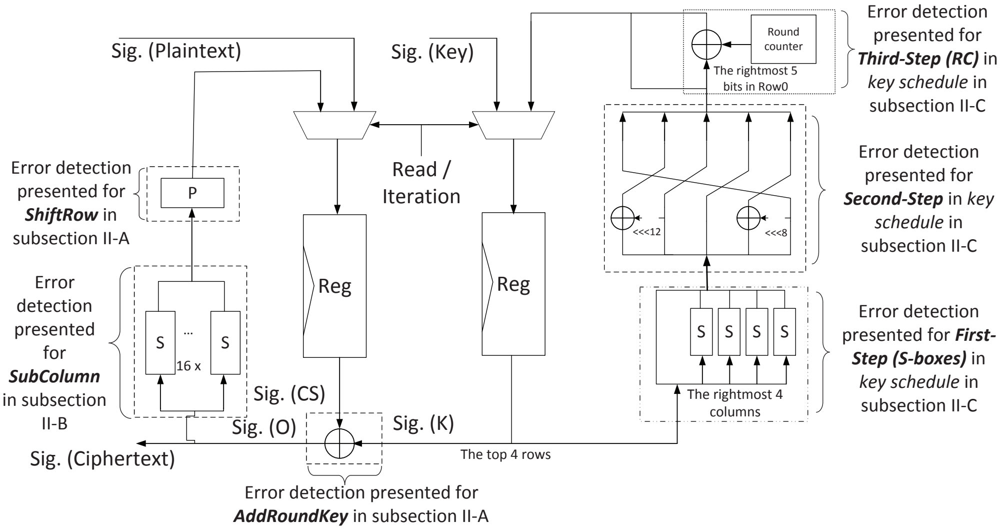
Sig. (Plaintext)
Sig. (Key)
Round counter
The rightmost 5 bits in Row0
Error detection presented for Third-Step (RC) in key schedule in subsection II-C
Read / Iteration
Error detection presented for ShiftRow in subsection II-A
P
Error detection presented for Second-Step in key schedule in subsection II-C
<<<12
<<<8
Reg
Reg
Error detection presented for SubColumn in subsection II-B
S
...
S
16 x
Error detection presented for First-Step (S-boxes) in key schedule in subsection II-C
S
S
S
S
The rightmost 4 columns
Sig. (CS)
Sig. (O)
Sig. (K)
Sig. (Ciphertext)
The top 4 rows
Error detection presented for AddRoundKey in subsection II-A

Figure 2. Error detection architectures for the encryption of RECTANGLE.

Table III
FPGA implementation results for the original RECTANGLE-80 encryption and our proposed error detection scheme on Virtex-7 FPGA

| Architecture | Area (occupied slices) | Delay (ns) | Power (mW) | Throughput (Gbps) | Efficiency (Mbps/slices) |
|---|---|---|---|---|---|
| RECTANGLE (LUT-based) | 100 | 1.05 | 177 | 60.95 | 609.52 |
| Signature-based error detection for LUTs | 108 (8%) | 1.20 (14.2%) | 179 (1.13%) | 53.33 (12.5%) | 493.83 (19%) |

they are a step-forward towards reliability and fault attack immunity of RECTANGLE lightweight block cipher.

## Acknowledgments

This work was performed under the U.S. federal agency award 60NANB16D245 granted from U.S. Department of Commerce, National Institute of Standards and Technology (NIST).

## References

[1] T. Eisenbarth, S. Kumar, C. Paar, and L. Uhsadel, “A survey of lightweight-cryptography implementations,” *IEEE Design & Test of Computers*, pp. 522-533, 2007.

[2] M. Mozaffari Kermani, M. Zhang, A. Raghunathan, and N. K Jha, “Emerging frontiers in embedded security,” in *Proc. Conf. VLSI Design*, Jan. 2013, pp. 203-208.

[3] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, “The LED block cipher,” in *Proc. CHES*, 2011, pp. 326-341.

[4] W. Wu, and L. Zhang, “LBlock: A lightweight block cipher,” in *Proc. ACNS*, Jun. 2011, pp. 327-344.

[5] J. Borghoff, A. Canteaut, T. Güneysu, E. B. Kavun, M. Knezevic, L. R. Knudsen, G. Leander, V. Nikov, C. Paar, and C. Rechberger, “PRINCE– A low-latency block cipher for pervasive computing applications,” in *Proc. Advances in Cryptology*, Springer, 2012, pp. 208-225.

[6] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, “The SIMON and SPECK lightweight block ciphers,” in *Proc. Design Automation Conf.,* 2015, pp. 1-6.

[7] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, and C. Vikkelsoe, “PRESENT: An ultra-lightweight block cipher,” in *Proc. CHES*, 2007, pp. 450-466.

[8] W. Zhang, Z. Bao, D. Lin, V. Rijmen, B. Yang, and I. Verbauwhede, “RECTANGLE: A bit-slice lightweight block cipher suitable for multiple platforms,” *Sci. China Inf. Sci.*, vol. 58, no. 12, pp.1-15, 2015.

[9] W. Zhang, Z. Bao, V. Rijmen, and M. Liu, “A new classification of 4-bit optimal S-boxes and its application to PRESENT, RECTANGLE and SPONGENT,” in *Proc. Fast Software Encryption*, Mar. 2015, pp. 494-515.

[10] R. Selvam, D. Shanmugam, and S. Annadurai, “Side channel attacks: Vulnerability analysis of PRINCE and RECTANGLE using DPA,” in *Proc. IACR Cryptology ePrint Archive*, 2014, p. 1-15.

[11] X. Guo, and R. Karri, “Recomputing with permuted operands: A concurrent error detection approach,” *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 10, pp. 1595-608, Oct. 2013.

[12] H. Pahlevanzadeh, J. Dofe, and Q. Yu, “Assessing CPA resistance of AES with different fault tolerance mechanisms,” in *Proc. Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2016, pp. 661-666.

[13] M. Mozaffari Kermani, R. Azarderakhsh, C.-Y. Lee, and S. Bayat-Sarmadi, “Reliable concurrent error detection architectures for extended Euclidean-based division over $GF(2^m)$,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 5, pp. 995-1003, 2014.

[14] M. Mozaffari Kermani, R. Azarderakhsh, and A. Aghaie, “Fault detection architectures for post-quantum cryptographic stateless hash-based secure signatures benchmarked on ASIC,” *ACM Trans. Embedded Computing Syst.* (special issue on Embedded Device Forensics and Security: State of the Art Advances), to appear in 2016.

[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] M. Mozaffari-Kermani, R. Azarderakhsh, and A. Aghaie, “Reliable and error detection architectures of Pomaranch for false-alarm-sensitive cryptographic applications,” *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 23, pp. 2804-2812, 2015.

[17] 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.