Error Detection Reliable Architectures of Camellia Block Cipher Applicable to Different Variants of its Substitution Boxes

Mehran Mozaffari Kermani
Department of Electrical
and Microelectronic Engineering
Rochester Institute of Technology
Rochester, NY
Email: m.mozaffari@rit.edu

Reza Azarderakhsh
Department of Computer
and Electrical Engineering
and Computer Science
Florida Atlantic University
Boca Raton, FL
Email: razarderakhsh@fau.edu

Jiafeng Xie
Department of Electrical Engineering
Wright State University
Fairborn, OH
Email: jiafeng.xie@wright.edu

***Abstract*—Different security properties are provided by cryptographic architectures to protect sensitive usage models such as implantable and wearable medical devices and nano-sensor nodes. Nevertheless, the way such algorithms are implemented could undermine the needed security and reliability aims. Unless the reliability of architectures is guaranteed, natural or malicious faults can undermine such objectives. Noting this, in this paper, we present error detection approaches for the Camellia block cipher taking into account its linear and non-linear sub-blocks. We also tailor the presented error detection architectures towards the desirability of using different variants of the S-boxes based on the security and reliability objectives. The merit of the proposed approaches is that (a) they can be tailored and applied to look-up table-based and composite field-based S-boxes, (b) their reliability *vs.* overhead can be fine-tuned based on the usage models, and (c) they result in high error coverage and acceptable overheads for performance and implementation metrics. We present the results of error simulations and application-specific integrated circuit (ASIC) implementations to benchmark the efficiency of the presented schemes.**

***Index Terms*—Application-specific integrated circuit (ASIC), block cipher, Camellia, fault detection, reliability.**

I. Introduction

Cryptographic block ciphers are used to efficiently provide different security properties for constrained applications. Sensitivity of such applications in terms of security and implementation metrics calls for lightweight and efficient architectures, and this necessitates having effective error detection schemes which constitute minimal overhead for the original structures, so that the added burden is acceptable. Battery drainage for a protected implantable medical device against security threats, e.g., pacemakers, due to the cryptographic implementation burden is catastrophic or at the very least uncomfortable for patients, needing surgery to replace the battery.

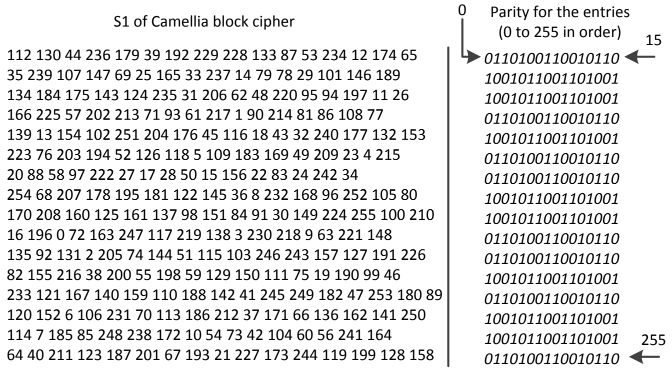
Camellia is a symmetric-key block cipher with block size of 128 bits and three key sizes of 128, 192, and 256 bits [1]. It was jointly developed by Mitsubishi Electric and Nippon Telegraph and Telephone (NTT) of Japan, and has been approved for use by the ISO/IEC, the European Union's New European Schemes for Signatures, Integrity, and Encryption (NESSIE) project, and the Japanese Cryptography Research and Evaluation Committees (CRYPTREC) project.

Block ciphers, e.g., Camellia, provide confidentiality; nevertheless, natural faults, e.g., through exposure to laser and cosmic ray particles such as alpha and gamma rays, and malicious faults undermine their reliability. A number of fault injection mechanisms such as temperature/optical/electromagnetic fault injection have been presented to date. Moreover, concurrent error detection (CED) techniques have been widely presented to account for reliable hardware architectures (including cryptographic block ciphers) [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]. Such schemes are based on hardware/information/time/hybrid redundancy, and have been presented to have a compromise for reliability and overhead tolerance. For instance, hardware redundancy uses extra hardware to process the same input twice to match the two outputs, information redundancy schemes have a number of variants, e.g., robust codes, and time redundancy approaches are capable of detecting both permanent and transient faults if recomputing with encoded operands is used.

In this paper, we consider the block cipher Camellia (see [12], [13], [14], [15] for some previous work on its different aspects) and propose both structure-dependent and structure-independent fault diagnosis approaches for its S-boxes and other sub-blocks. The cipher has security levels and processing abilities comparable to the Advanced Encryption Standard (AES). Moreover, in mid-2013, Camellia was selected for adoption in Japan's new e-government recommended ciphers list as the only 128-bit block cipher encryption algorithm developed in Japan. Our main contributions in this paper are summarized as follows:

- We propose error detection approaches for the block cipher Camellia, considering the reliability and performance metrics objectives. Signature-based approaches are used for the linear and non-linear blocks to achieve high efficiency, while maintaining high error coverage.
- The presented error detection approaches for the S-boxes

978-1-5090-5701-6/16/$31.00 ©2016 IEEE



S1 of Camellia block cipher	Parity for the entries (0 to 255 in order)
112 130 44 236 179 39 192 229 228 133 87 53 234 12 174 65	0110100110010110
35 239 107 147 69 25 165 33 237 14 79 78 29 101 146 189	1001011001101001
134 184 175 143 124 235 31 206 62 48 220 95 94 197 11 26	1001011001101001
166 225 57 202 213 71 93 61 217 1 90 214 81 86 108 77	0110100110010110
139 13 154 102 251 204 176 45 116 18 43 32 240 177 132 153	1001011001101001
223 76 203 194 52 126 118 5 109 183 169 49 209 23 4 215	0110100110010110
20 88 58 97 222 27 17 28 50 15 156 22 83 24 242 34	0110100110010110
254 68 207 178 195 181 122 145 36 8 232 168 96 252 105 80	1001011001101001
170 208 160 125 161 137 98 151 84 91 30 149 224 255 100 210	1001011001101001
16 196 0 72 163 247 117 219 138 3 230 218 9 63 221 148	0110100110010110
135 92 131 2 205 74 144 51 115 103 246 243 157 127 191 226	0110100110010110
82 155 216 38 200 55 198 59 129 150 111 75 19 190 99 46	1001011001101001
233 121 167 140 159 110 188 142 41 245 249 182 47 253 180 89	0110100110010110
120 152 6 106 231 70 113 186 212 37 171 66 136 162 141 250	1001011001101001
114 7 185 85 248 238 172 10 54 73 42 104 60 56 241 164	1001011001101001
64 40 211 123 187 201 67 193 21 227 173 244 119 199 128 158	0110100110010110

Fig. 1. The proposed fault diagnosis scheme for the S-box s_1, which can be adopted for s_2, s_3, and s_4 based on the proposed Theorem 2.

within Camellia can be applied to its composite field (tower field, e.g., using polynomial basis, normal basis, and mixed basis) as well as look-up table-based implementations. The former achieves low-area/power and the latter, typically, has higher speed on hardware platforms.

- We benchmark the proposed architectures to assess their ability to detect transient and permanent faults by performing fault injection simulations. Moreover, we implement the proposed error detection architectures on application-specific integrated circuit (ASIC) platform using 65-nm standard-cell library. Our results show that the proposed efficient error detection architectures can be feasibly utilized for Camellia, suitable for the required performance, reliability, and implementation metrics for constrained applications.

II. Proposed Error Detection Approaches

In this section, the error detection schemes used for detecting the transient and permanent faults in the Camellia block cipher are presented.

A. Error Detection Schemes for Non-linear S-boxes

The S-Boxes of Camellia can be realized through two different (and far diverse) approaches in hardware. In the applications where memory macros on ASIC and block memories on field-programmable gate array (FPGA) are utilized, one can realize the S-boxes through look-up tables. Such variants of S-boxes have, typically, high performance but constitute much area and power consumption.

The S-boxes in Camellia are part of the so-called S-function. In the S-function of Camellia, four S-boxes are utilized, i.e., s_1, s_2, s_3, and s_4. The four S-boxes of Camellia are affine equivalent to an inversion function over $GF(2^8)$. In this paper, without loss of generality, we focus on s_1, as the other S-boxes, s_2, s_3, and s_4, are derived by cyclic shifts from s_1, i.e., for the 8-bit input x, we have the outputs of these three S-boxes, respectively, as $s_1(x) \lll 1$, $s_1(x) \ggg 1$, and $s_1(x \lll 1)$. However, when needed, their error detection constructions are also presented. In what follows, a number of schemes are presented for the error detection of the S-boxes.

1) Look-up Table-based S-boxes: A viable approach for fault diagnosis of such S-boxes is to store the predicted signatures of the look-up table entries in the S-boxes through expanding the realizations. We have derived the predicted parities as an example for this scheme; nevertheless, it does not confine the proposed approach to only this signature variant. Such an error detection approach is shown in Fig. 1. As seen in this figure, for each entry of s_1, the predicted parities are shown in the right columns. In what follows, we present a theorem through which the error detection architectures for the other S-boxes s_2, s_3, and s_4 are presented.

Theorem 1. *For the S-boxes s_2 and s_3, the same construction as the one in Fig. 1 can be used to store the predicted parities. However, one needs to use interleaved parities for even entries and then odd entries of the S-box s_1 to derive the predicted parities for the S-box s_4 in Camellia.*

Proof: The S-boxes s_2 and s_3 are constructed through the following adoptions from the S-box s_1: For the 8-bit input x, we have the outputs of these two S-boxes, respectively, as $s_1(x) \lll 1$, $s_1(x) \ggg 1$. Cyclic shift does not change the predicted parities as it is just rewiring in hardware; therefore, for the S-boxes s_2 and s_3, the same construction as the one in Fig. 1 can be used to store the predicted parities. Nonetheless, for the other S-box, i.e., s_4, we have the output as $s_1(x \lll 1)$. Thus, the cyclically-shifted input x to the left (multiplication by 2) is applied to s_1 for this output and, thus, the output bytes are "picked" from the left column of Fig. 1 (s_1) in interleaved fashion starting from the zeroth entry $(112)_{10}$. Such an interleaved construction for s_4 results in interleaved construction for the predicted parities as well. This completes the proof. ■

The proposed approach is an efficient construction for look-up table-based S-boxes. Nevertheless, in what follows, we present an architecture-oblivious approach which is viable for different variants of the S-boxes.

2) Architecture-oblivious Approach: Look-up table-based realization of the S-boxes suffers from high power consumption, area, and energy usage. On the other hand, composite field-based architectures are low-complexity but need pipelining to reach high throughput and efficiency. Therefore, in what follows, we present an architecture-oblivious approach for fault diagnosis of the S-boxes in Camellia which is applicable to these and other implementations, and, thus, gives flexibility to reach the desired objectives, as it derives a relation between the input and the output of these structures. Let us present the following theorems to derive signatures for error detection in these architectures.

Theorem 2. *[16] Let $C = \sum_{i=0}^{m-1} c_i \alpha^i$ be the multiplication of A and $B \in GF(2^m)$. Then, the coordinates of C can be obtained from*

$$[c_0, c_1, \cdots, c_{m-1}]^T = (\boldsymbol{L} + \boldsymbol{Q^T U})\boldsymbol{b}, \qquad (1)$$

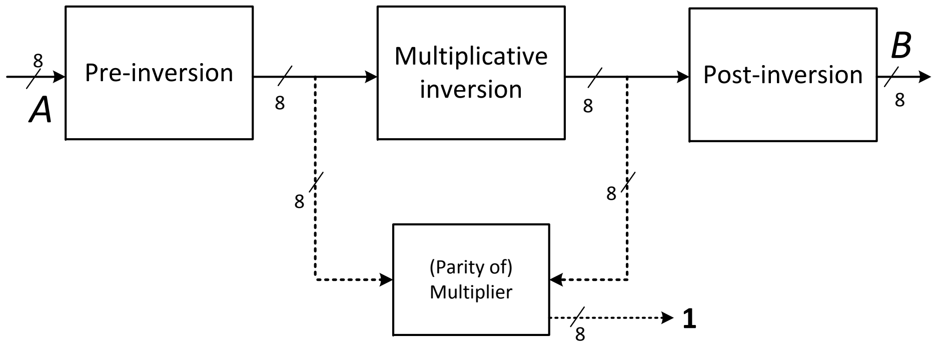


Fig. 2. The proposed architecture-oblivious error detection scheme for Camellia's multiplicative inversion.

where, $\boldsymbol{b} = [b_0, b_1, \cdots, b_{m-1}]^T$*,*

$$\boldsymbol{L} = \begin{pmatrix} a_0 & 0 & 0 & 0 & \ldots & 0 \\ a_1 & a_0 & 0 & 0 & \ldots & 0 \\ a_2 & a_1 & a_0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{m-2} & a_{m-3} & \ldots & a_1 & a_0 & 0 \\ a_{m-1} & a_{m-2} & \ldots & a_2 & a_1 & a_0 \end{pmatrix}, \tag{2}$$

$$\boldsymbol{U} = \begin{pmatrix} 0 & a_{m-1} & a_{m-2} & \ldots & a_2 & a_1 \\ 0 & 0 & a_{m-1} & \ldots & a_3 & a_2 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 & a_{m-1} & a_{m-2} \\ 0 & 0 & \ldots & 0 & 0 & a_{m-1} \end{pmatrix}, \tag{3}$$

and the $m-1$ *by* m *binary matrix* $\boldsymbol{Q}$ *is obtained as follows*

$$[\alpha^m, \alpha^{m+1}, \ldots, \alpha^{2m-2}]^T = \boldsymbol{Q}[1, \alpha, \alpha^2, \ldots, \alpha^{m-1}]^T (\text{mod} P(\alpha)). \tag{4}$$

Based on Theorem 2, one can derive the relation between the input and the output of the multiplicative inversion of the Camellia block cipher; nevertheless, the linear blocks need to be also considered to account for the entire S-boxes architectures. The multiplicative inversion on Galois field and the affine transformations of Camellia and the respective field structure are not clearly described in [1]. Nevertheless, the research work in [17], [18] have investigated a number of fields, and found the field extended by using the irreducible polynomial which satisfies the respective look-up table for the S-box.

Camellia utilizes the primitive polynomial of $f(z) = z^8 + z^6 + z^5 + z^3 + 1$. Moreover, for composite field, one can utilize $GF(2^4) : g_0(x) = x^4 + x + 1$ and $GF(2^4)^2 : g_1(x) = x^2 + x + \{1001\}_2$. In addition, the pre- and post-inversion affine transformations are described as follows ($A = (a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7)^T$ and $\lambda^{-1} = (\lambda_0^{-1}, \lambda_1^{-1}, \lambda_2^{-1}, \lambda_3^{-1}, \lambda_4^{-1}, \lambda_5^{-1}, \lambda_6^{-1}, \lambda_7^{-1})^T$ are the input column vectors and $B = (b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7)^T$ and $\lambda = (\lambda_0, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7)^T$ are the output column vectors, respectively):

$$\begin{aligned} \boldsymbol{\lambda} &= \boldsymbol{\Gamma A} + \boldsymbol{\gamma} \\ &= \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix} \boldsymbol{A} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}. \end{aligned} \tag{5}$$

$$\begin{aligned} \boldsymbol{B} &= \boldsymbol{\Gamma}' \boldsymbol{\lambda}^{-1} + \boldsymbol{\gamma}' \\ &= \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \boldsymbol{\lambda}^{-1} + \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}. \end{aligned} \tag{6}$$

Based on the above, we present the following theorem whose proof is not presented for the sake of brevity; yet, it is derived from Theorem 2 and (1)-(6).

Theorem 3. *Let* $\lambda = \lambda_7\alpha^7 + \lambda_6\alpha^6 + \lambda_5\alpha^5 + \lambda_4\alpha^4 + \lambda_3\alpha^3 + \lambda_2\alpha^2 + \lambda_1\alpha + \lambda_0$ *and* $\lambda^{-1} = \lambda_7^{-1}\alpha^7 + \lambda_6^{-1}\alpha^6 + \lambda_5^{-1}\alpha^5 + \lambda_4^{-1}\alpha^4 + \lambda_3^{-1}\alpha^3 + \lambda_2^{-1}\alpha^2 + \lambda_1^{-1}\alpha + \lambda_0^{-1}$ *be the input and output of the multiplicative inversion in the binary field* $GF(2^8)$*, respectively (we have shown the structure of the S-box through composite field in Fig. 2). The result of the multiplication of the input and the output of the multiplicative inversion is, naturally, the unity polynomial* $1 \in GF(2^8)$ *(shown in Fig. 2). Then, we have*

$$\boldsymbol{f}(\lambda) \times \boldsymbol{\lambda}^{-1} = \boldsymbol{1}, \tag{7}$$

where the indices for the elements of $f(\lambda)$ *are shown below (comma indicates modulo-2 addition), for instance,* $5, 7 =$

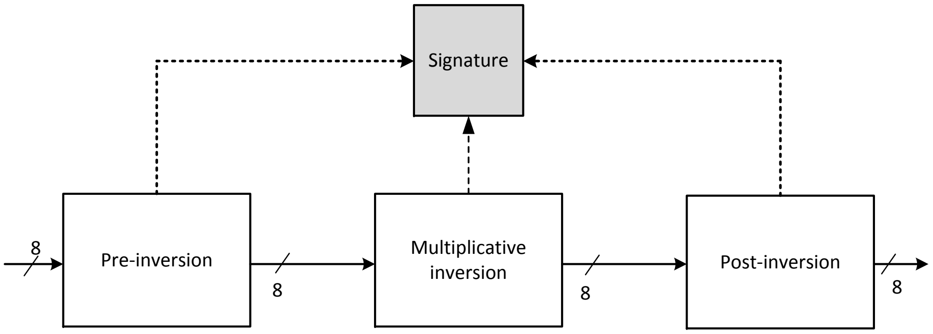


Fig. 3. The proposed signature-based error detection approach for the entire architectures of the S-boxes.

$\lambda_5+\lambda_7$:

$$\begin{pmatrix} 0 & 7 & 6 & 5,7 & 4,6,7 & 3,5,6,7 & 2,4,5,6,7 & 1,3,4,5,6 \\ 1 & 0 & 7 & 6 & 5,7 & 4,6,7 & 3,5,6,7 & 2,4,5,6,7 \\ 2 & 1 & 0 & 7 & 6 & 5,7 & 4,6,7 & 3,5,6,7 \\ 3 & 2,7 & 1,6 & 0,5,7 & 4,6 & 3,5,7 & 2,4,6 & 1,3,5,7 \\ 4 & 3 & 2,7 & 1,6 & 0,5,7 & 4,6 & 3,5,7 & 2,4,6 \\ 5 & 4,7 & 3,6 & 2,5 & 1,4,7 & 0,3,6 & 2,5,7 & 1,4,6,7 \\ 6 & 5,7 & 4,6,7 & 3,5,6,7 & 2,4,5,6,7 & 1,3,4,5,6 & 0,2,3,4,5,7 & 1,2,3,4,6,7 \\ 7 & 6 & 5,7 & 4,6,7 & 3,5,6,7 & 2,4,5,6,7 & 1,3,4,5,6 & 0,2,3,4,5,7 \end{pmatrix}. \tag{8}$$

The above scheme is a general approach for the multiplicative inversion of the S-boxes. We now present the error detection schemes for the entire S-boxes. Considering (5), (6), and (8), let us derive parities and interleaved parities for the pre-inversion, post-inversion, and multiplicative inversion. This is shown in Fig. 3. We present the following two theorems for derivation of the aforementioned signatures. The proofs (not presented for the sake of brevity) are through signature derivation for (5), (6), and (8).

Theorem 4. *One can derive the following predicted parities for the proposed error detection for pre-inversion, post-inversion, and the eight column signatures for $f(\lambda)$ (compared based on (7) in multiplicative inversion), where the "hat" symbol represents the predicted parities, and $P_\lambda=\sum_{i=0}^{i=7}\lambda_i$:*

$$\hat{P}_\lambda = a_7, \tag{9}$$

$$\hat{P}_B = \overline{\lambda_1^{-1}+\lambda_3^{-1}+\lambda_5^{-1}}, \tag{10}$$

$$f(\lambda) \to (P_\lambda, P_\lambda+\lambda_7, P_\lambda+\lambda_{7,6}, P_\lambda+\lambda_{6,5}, P_\lambda+\lambda_{7,5,4} \\ , P_\lambda+\lambda_{6,4,3}, \lambda_{6,4,1,0}, \lambda_{5,3,0}). \tag{11}$$

Theorem 5. *One can derive the following predicted interleaved parities for the proposed error detection for pre-inversion and post-inversion transformations, where $P_A=\sum_{i=0}^{i=7}a_i$:*

$$\hat{P}_\lambda^{(1)} = \overline{P_A}+a_7, \hat{P}_\lambda^{(2)} = \overline{P_A}, \tag{12}$$

$$\hat{P}_B^{(1)} = \overline{P_{\lambda^{-1}}}+\lambda_4+\lambda_6+\lambda_7, \hat{P}_B^{(2)} = \lambda_0^{-1}+\lambda_2^{-1}. \tag{13}$$

B. Error Detection Schemes for Linear Blocks

The linear transformations in Camellia consist of P-function (which together with S-function creates F-function) and also two other functions, i.e., FL-function and FL^{-1}-function which consist of XOR operations with the keys. The errors in the two latter functions can be detected through any signatures as the error detection for XOR operation is straightforward.

In the key schedule block of Camellia-128, the right-side key is zero and the left-side key is used as the 128-bit key K. The key schedule unit contains XOR gates as well as F-functions (which include S-function and P-function) whose error detection architectures are presented earlier in Section II. In what follows, we present the error detection schemes for P-function.

In the proposed error detection schemes for P-function, we use column byte signatures and interleaved column byte signatures; however, this does not confine the presented schemes to just these two signatures. P-function can be treated as the following matrix multiplied to a column vector with eight 8-bit entries, i.e., z_8-z_1, to derive a column vector with eight 8-bit entries, i.e., $z_8'-z_1'$:

$$M=\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

Based on above, the followings are derived and this completes the error detection architectures for the linear blocks ($Sig.$ denotes column byte signature and $Sig.^{(1/2)}$ denotes column byte interleaved signature): $Sig.z'=z_4+z_3+z_2+z_1, Sig.^{(1)}z'=z_4+z_2, Sig.^{(2)}z'=z_3+z_1$.

III. Error Simulations and ASIC Implementations

In this section, we present the error injection simulations and ASIC implementations for Camellia encryption and decryption including the key schedule unit.

To benchmark the effectiveness of the proposed schemes, we utilize linear feedback shift registers (LFSRs) to inject

TABLE I
ASIC TSMC 65NM IMPLEMENTATIONS OF COMBINED CAMELLIA-128 ENCRYPTION AND DECRYPTION.

Architecture	Area (μm^2)	Area (GE)	Delay (ns)	Throughput ($Gbps$)	Efficiency ($\frac{Kbps}{\mu m^2}$)	Power (mW)
Original-pipelined	268,960	190,751	12.4	10.2	37.9	11.8
Proposed scheme	316,398 (17.6%)	224,395 (17.6%)	13.5 (8.8%)	9.5 (6.8%)	29.9 (21%)	14.3 (21%)

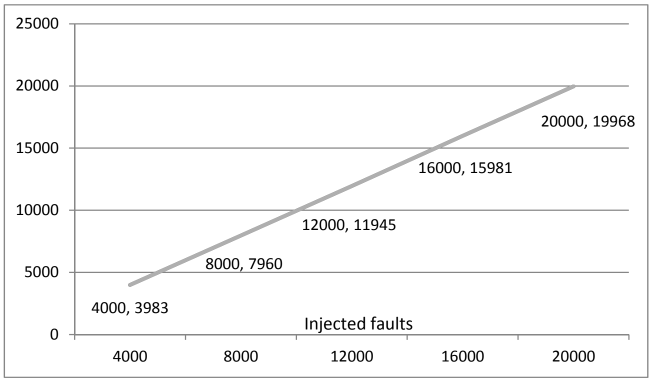


Fig. 4. Error simulations for up to 20,000 injections for composite field S-boxes.

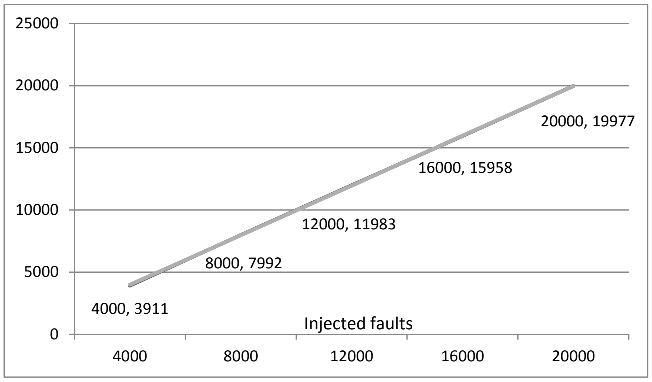


Fig. 6. Error simulations for up to 20,000 injections for LUT-based S-boxes.

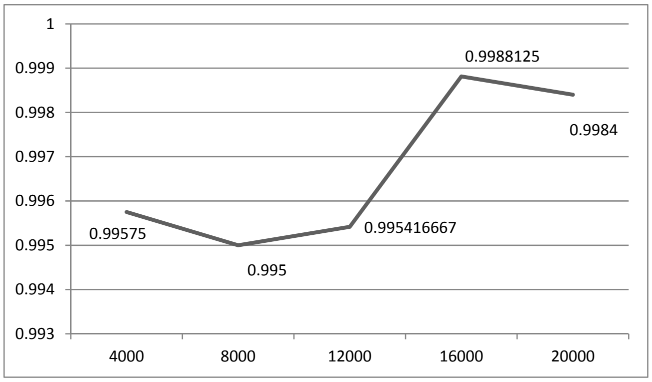


Fig. 5. Error coverage for composite field S-boxes.

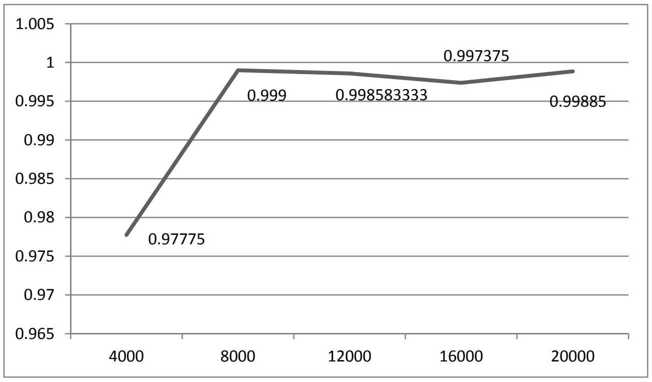


Fig. 7. Error coverage for LUT-based S-boxes.

stuck-at zero and one faults. We incorporate LFSRs to generate pseudo-random fault patterns for every clock cycle. This is useful as we can use LFSRs in conjunction with multiplexers to have random multiple faults whose type (stuck-at zero or one), location, and number are random.

We have considered transient and permanent faults (the former is used in fault attacks and also happens through defects whereas the latter is mainly due to defects) as well as single, burst (detected through the burst error detection schemes presented in Section II), and random faults in the architectures. Maximum length LFSRs with the respective polynomials are used to inject up to 20,000 faults in Camellia encryption and the error detection schemes are benchmarked.

Excluding masked faults, the presented architectures (we have simulated both the look-up table-based S-boxes and the composite field ones along with the presented schemes in Section II) have detected the errors for look-table based S-boxes (and for composite field S-boxes) for the entire Camellia encryption as seen in Figs. 4-5 (Figs. 6-7). Single stuck-at faults are detected using the presented schemes. Finally, for burst faults, we have used interleaved parities (presented in Section II), and error coverage of very close to 100% is achieved.

In this section, we also present the area, power consumption, and delay overhead results as well as throughput and efficiency degradations for the combined encryption and decryption of the Camellia block cipher through ASIC implementations (the decryption of Camellia is performed the same way as the encryption procedure by reversing the order of the subkeys). The benchmarking is done for both the original and fault detection architectures using TSMC 65nm library (shown in Table I).

As seen in Table I, for the pipelined combined encryption/decryption structures, the original (which is an efficient implementation without error detection) and parity-based performance and implementation metrics are shown. The overheads (degradations) are at most 21% for the efficiency measure. We would like to emphasize that as our proposed architectures are independent of the hardware platforms, we expect similar results for other ASIC libraries and also for the FPGA platform. Moreover, the presented results are for Camellia-128 (128-bit key); yet, the architectures for error detection are the same for other variants, i.e., Camellia-192 and Camellia-256.

IV. Conclusions and Discussions

In this paper, signature-based error detection approaches are presented which can be applied to the encryption and decryption of Camellia block cipher. Formulae for the linear and non-linear sub-blocks of the Camellia block cipher are presented, tailoring which one can achieve the intended reliability objectives. Through fault-injection analysis, it has been shown that the error coverage is close to 100%. Furthermore, through ASIC implementations, we have shown that acceptable overheads are achieved. Based on the available resources, one may utilize the proposed error detection schemes for making the hardware implementations of Camellia more reliable.

A subset of fault attacks, differential fault intensity analysis (DFIA), see for instance, [19], [20], [21], combines the idea of differential power analysis with fault injection principles to obtain biased fault models; whose merit is that same fault in both the original and redundant computations can be injected, more practically, where not all faults occur with equal probability. Previous work argue that the single-bit (more likely in low fault intensity), two-bit, three-bit, and four-bit (more likely in higher intensities) biased fault models can be used to simulate variation of fault intensity. While the proposed approaches in this paper (based on error detecting codes) are able to thwart a number of such fault models, recomputing with encoded operands can be paired with them (if the overhead can be tolerated) to thwart such attacks. An intriguing future work would be to investigate combined fault and power analysis attacks countermeasures for Camellia (there has been related prior work, for instance, [22], [23], not considering Camellia). Moreover, compared to [24] which uses duplication of the S-boxes for error detection, the proposed approach here (based on the resources available and reliability objectives) can be tailored to achieve more efficient error detection structures.

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), awarded to Mehran Mozaffari Kermani and Reza Azarderakhsh.

References

[1] Specification of Camellia, a 128-bit block cipher, https://info.isl.ntt.co.jp/crypt/eng/camellia/dl/01espec.pdf.

[2] 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.

[3] G. Di Natale, M. Doulcier, M. L. Flottes, and B. Rarchitectures, “A reliable architecture for parallel implementations of the Advanced Encryption Standard,” *J. Electronic Testing: Theory and Applications*, vol. 25, no. 4, pp. 269–278, Aug. 2009.

[4] 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.

[5] 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.

[6] X. Guo, D. Mukhopadhyay, C. Jin, and R. Karri, “Security analysis of concurrent error detection against differential fault analysis,” *J. Cryptographic Engineering,* vol. 5, no. 3, pp. 153-169, 2015.

[7] 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.

[8] 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.

[9] M. Karpovsky, K. J. Kulikowski, and A. Taubin, “Robust protection against fault-injection attacks on smart cards implementing the Advanced Encryption Standard,” in *Proc. Dependable Systems and Networks,* 2004, pp. 93-101.

[10] K. Wu and R. Karri, “Algorithm-level recomputing with shifted operands-A register transfer level concurrent error detection technique,” *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,* vol. 25, no. 3, pp. 413-422, 2006.

[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. 1595-1608, 2013.

[12] J. Lu, Y. Wei, P. A. Fouque, and J. Kim, “Cryptanalysis of reduced versions of the Camellia block cipher,” *IET Information Security*, vol. 6, no. 3, pp. 228-238, Sep. 2012.

[13] M. Sugita, K. Kobara, and H. Imai, “Security of reduced version of the block cipher Camellia against truncated and impossible differential cryptanalysis,” in *Proc. Asiacrypt*, 2001, pp. 193-207.

[14] A. Biryukov and I. Nikolic, “Acutomatic search for related-key differential characteristics in byte-oriented block ciphers: Application to AES, Camellia, Khazad and others,” in *Proc. Eurocrypt*, 2010, pp. 322-344.

[15] J. Lu, Y. Wei, E. Pasalic, and P.-A. Fouque, “Meet-in-the-middle attack on reduced versions of the Camellia block cipher,” in *Proc. IWSEC*, 2012, pp. 197-215.

[16] A. Reyhani-Masoleh and M. Hasan, “Low complexity bit parallel architectures for polynomial basis multiplication over $GF(2^m)$,” *IEEE Trans. Comput.*, vol. 53, no. 8, pp. 945-959, 2004.

[17] A. Satoh and S. Morioka, “Unified hardware architecture for 128-bit block ciphers AES and Camellia,” in *Proc. Cryptographic Hardware and Embedded Systems*, 2003, pp. 304-318.

[18] A. F. Martinez-Herrera, J. C. Mex-Perera, and J. A. Nolazco-Flores, “Some representations of the S-Box of Camellia in $GF(((2^2)^2)^2)$,” in *Proc. CANS,* 2012, pp. 296-309.

[19] N. Farhady Ghalaty, B. Yuce, and P. Schaumont, “Analyzing the efficiency of biased-fault based attacks,” *Embedded Systems Letters,* vol. 8, no. 2, pp. 33-36, 2016.

[20] N. Farhady Ghalaty, B. Yuce, M. M. I. Taha, and P. Schaumont, “Differential fault intensity analysis,” in *Proc. FDTC,* 2014, pp. 49-58.

[21] S. Patranabis, A. Chakraborty, P. Ha Nguyen, and D. Mukhopadhyay, “A biased fault attack on the time redundancy countermeasure for AES,” in *Proc. COSADE,* 2015, pp. 189-203.

[22] J. Dofe, H. Pahlevanzadeh, and Q. Yu, “A comprehensive FPGA-based assessment on fault-resistant AES against correlation power analysis attack,” *J. Electronic Testing*, DOI 10.1007/s10836-016-5598-9, June 2016.

[23] 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.

[24] C. Huiju and H. M. Heys, “Compact hardware implementation of the block cipher Camellia with concurrent error detection,” in *Proc. CCECE*, pp. 1129-1132, 2007.