Reliable And Low-Latency And Low-Complexity For Pomaranch For False-Alarm-Sensitive Cryptographic Applications

Jadhav Swathi, Pg Student

Mrs.M.Anusha, Associate Professor

Dept Of Ece

Malla Reddy College of Engineering &Technology,Secunderabad

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(33)]. 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.

INTRODUCTION

The term “false alarm” refers to alarm systems in many different applications being triggered by something other than the expected trigger-event. Examples of this those applications include residential burglar alarms, smoke detectors, industrial alarms, and signal detection theory.

The term “false alarm” may actually be semantically incorrect in some uses. For example, a residential burglar alarm could easily be triggered by the residents of a home accidentally. The alarm is not necessarily false – it was triggered by the expected event – but it is “false” in the sense that the police should not be alerted. Due to this problem, false alarms can also be referred to as “nuisance alarms.”

Residential burglar alarms can be caused by improper arming and disarming of the system, power outages and weak batteries, wandering pets, and unsecured doors and windows. [2] In the U.S. false alarms cost police agencies up 6.5 million personnel hours, according to the International Association of Chiefs of Police. [3] A 2002 study by the U.S. Justice Department estimated the cost of false alarms to be as high as $1.5 billion. [4] Due to this cost, many cities now require permits for burglar alarms, have enacted verified response protocols, or have introduced fines for excessive false alarms. This is typically caused by simple mistakes like entering the wrong passcode or letting too much time pass before entering the code. These types of false alarms can be prevented by taking more time to disarm systems, and entering a home with at least one hand free to properly disarm one's system. 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, wechoose 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. 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.

2) We propose fault diagnosis approaches for the lightweight 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. 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 (JCi) and out (JCo) 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 G F((23)3) is used through which the complexity of the operations needed for realizing the inverse is much reduced. The finite field G F(29) is represented by elements (in terms of polynomials) of degree eight. The field G F(29) can also be represented as G F((23)3), where, here, the elements of this composite field are given as polynomials of degree at most 2 with coefficients from G F(23) [19]. We follow the representation in [19] and based on the search performed through the set of primitive polynomials of degree nine over G F(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 +γ 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, (α7) 9+α7+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.

EXISTING SYSTEM

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 (JCi) and out (JCo) 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 lookuptables) 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 G F((23)3) is used through which the complexity of the operations needed for realizing the inverse is much reduced. The finite field G F(29) is represented by elements (in terms of polynomials) of degree eight. The field G F(29) can also be represented as G F((23)3), where, here, the elements of this composite field are given as polynomials of degree at most 2 with coefficients from G F(23) [19]. We follow the representation in [19] and based on the search performed through the set of primitive polynomials of degree nine over G F(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 +γ 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, (α7) 9+α7+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.

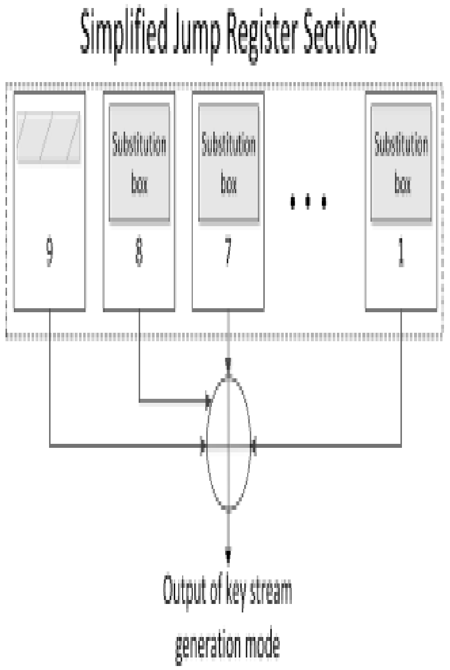

Simplified Jump Register Sections
Substitution box
Substitution box
Substitution box
9
8
7
1
Output of key stream generation mode

Fig. 1. Simplified accumulated cascade jump control.

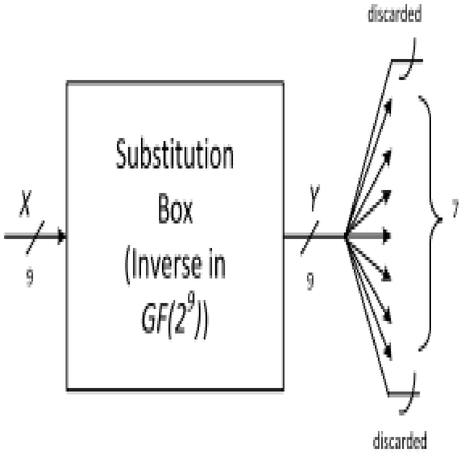
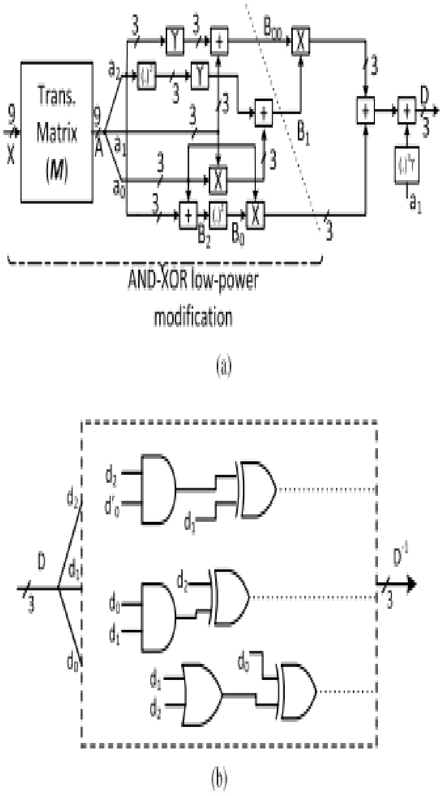


Fig. 2.9-to-7 substitution box and its uneven structure.



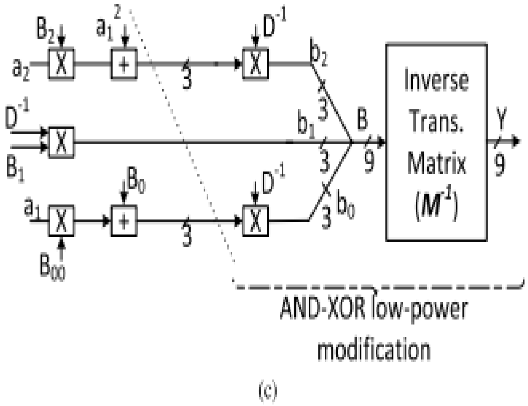

B₂
a₁²
D⁻¹
a₂
X
+
3
X
b₂
Inverse Trans. Matrix (M^{-1})
Y
9
D⁻¹
B₁
X
b₁
3
B
9
3
a₁
X
B₀
+
3
D⁻¹
X
b₀
3
B₀₀
AND-XOR low-power modification
(c)

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.

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 memorymacros 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).

PROPOSED SYSTEM

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.

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 G F(23). Moreover, observing Fig. 3(b), the architecture of inversion in G F(23) 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 G F(23). Theorem 1: Let X ∈ G F(23) and Y ∈ G F(23) be two elements in composite fields. Let the vectors X = (x2, x1, x0) and Y = (y2, y1, y0) be their respective vectors representing these elements. The predicted parity of R = X × Y ∈ G F(23), i.e., Pˆ R, is derived as Pˆ R = x0(y0 + y1 + y2) + x1(y0 + y1) + x2 y0 (1) where + denotes the XOR operation (modulo-2 add). Proof: Considering the formulas for the multiplications in the subfield G F(23) as follows: r0 = x0 y0 + x1 y2 + x2 y1 (2) r1 = x0 y1 + x1 y0 + x1 y2 + x2 y1 + x2 y2 (3) r2 = x0 y2 + x2 y0 + x1 y1 + x2 y2 (4) we can modulo-2 add the coordinates of the respective vectors representing the result, i.e., through x0 y0 + x1 y2 + x2 y1 + x0 y1 +x1y0 +x1 y2 +x2 y1 +x2 y2 +x0 y2 +x2 y0 +x1 y1 +x2 y2, to reach to the predicted parity derived in (1) and the proof is complete. Theorem 2: Let X ∈ G F(23) be an element in composite fields. Let the vector X = (x2, x1, x0) be its respective vector representing this element. The predicted parity of the inverse element X−1 ∈ G F(23), i.e., Pˆ X−1 , is derived as Pˆ X−1 = x0x 1 + (x1 + x0)x2. (5) Proof: Considering the formulas for the inversion output bits in the subfield G F(23), i.e., x0 + (x1 ∨ x2), x0x1 + x2, and x1 + x0x2 (where the symbol ∨ represents the OR operation), one can perform a modulo-2 addition to reach Pˆ 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 G F(23). These are presented through the following theorem. Theorem 3: Let X ∈ G F(23) be an element in composite fields. Let γ ∈ G F(23) be a constant in composite fields as well. The predicted parities for squaring X2, i.e., Pˆ X2 , cubing, and multiplication by the constant γ , i.e., Pˆ X3γ , and multiplication by the constant γ , i.e., Pˆ Xγ , are presented below Pˆ X2 = x0 + x1 (6) Pˆ X3γ = x0x1 + (x0 + x1)x 2 (7) Pˆ Xγ = x0 + x1. (8) Proof: Based on the formulas for the squaring output bits in the subfield G F(23), i.e., x0, x2, and x1 + x2, one can perform a modulo-2 addition to reach Pˆ X2 in (6). Considering the equations for the cubing and multiplication by the constant output bits in the subfield G F(23), i.e., x0x1 + x2, (x0 ∨ x2) + x1x2, and (x0 ∨ x2) + x0x 2, one can perform a modulo-2 addition to reach Pˆ X3γ in (7). Finally, considering the multiplication by the constant γ output bits in the subfield G F(23), i.e., x2, x2+x0, and x1, one can perform a modulo-2 addition to reach PˆXγ in (8). This completes the proof. Remark 1: The hardware complexities for the predicted parities of R = X × Y ∈ G

F(23), i.e., P^ R, in Theorem 1 and X−1 ∈ G F(23), i.e., 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.

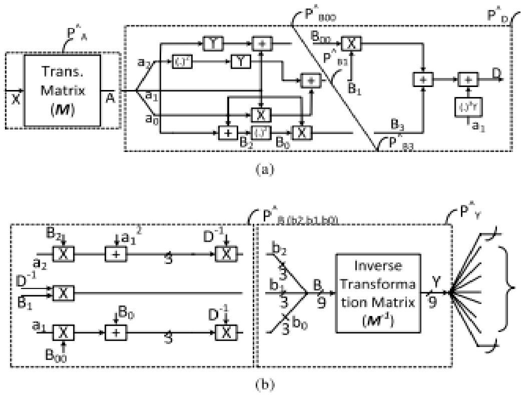


Fig. 4.Predicted signatures of the substitution box for (a) subpart 1 and (b) subpart 3.

The hardware complexities for the predicted parities of squaring X2, i.e., P^ X2 , cubing, and multiplication by the constant γ , i.e., P^ X3γ , and multiplication by the constant γ , i.e., P^ Xγ , 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 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 ∈ G F(29) to get an output of A ∈ G F((23)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., PˆA, 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 ∈ G F((23)3) be an element in composite fields, as shown in Fig. 4. Then, the predicted parity of the transformation matrix, i.e., PˆA, 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 ∈ G F(29), acting as the row vector [x8,..., x1, x0] multiplied by the transformation matrix to derive the 9-bit row vector representing A ∈ G F((23)3) PˆA = x0 + x1 + x5 + x6 + x7 + x8 = PX + x2 + x3 + x4 (9) where PX is the actual parity for the input X ∈ G F(29). 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 ∈ G F(29), act as the row vector [x8,..., x1, x0] multiplied by the transformation matrix to derive the 9-bit row vector representing A ∈ G F((23)3). Moreover, noting that PX = x0 + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8, one can derive the predicted parity of the transformation matrix, i.e., PˆA, 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 PX is available prior to the computations. Proposition 1: The bit-interleaved parities of A ∈ G F((23)3) (used for detecting burst and adjacent errors) as an element in composite fields can be derived as follows noting that PˆA,0 is for even entries (e.g., zeroth, second, fourth, and so on) and PˆA,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 ∈ G F(29) Pˆ A,0 = x0 + x2 + x3 + x4 + x5 + x7 = PX + x1 + x6 + x8 (10) Pˆ A,1 = x1 + x2 + x4 + x6 + x7 + x8 = PX + x0 + x3 + x5 (11) where PX is the actual parity for the input X ∈ G F(29). The predicted signatures for the next four signatures shown in Fig. 4(a) can be derived based on Theorems 1–3. Specifically, for PˆB1, Pˆ B3, Pˆ B00, and Pˆ 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 B00 = [θ2, θ1, θ0] and B1 = [ω2, ω1, ω0]. For the sake of brevity, we just present these as follows using the notations in Fig. 4(a) and the proof is omitted Pˆ B1 = a0(a3 + a4 + a5) + a1(a3 + a4) + a2a5 + a8 + a6 (12) Pˆ B3 = a0(a7 + a6) + a1(a8 + a6) + a2(a7 + a8) (13) Pˆ B00 = a7 + a6 + a5 + a4 + a3 (14) PˆD = ω0(θ0 + θ1 + θ2) + ω1(θ0 + θ1) + ω2θ2 + a3a4 + (a3 + a4)a5. (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 Pˆ B and Pˆ 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 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

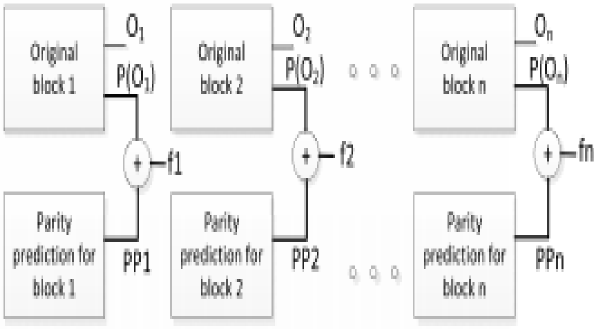
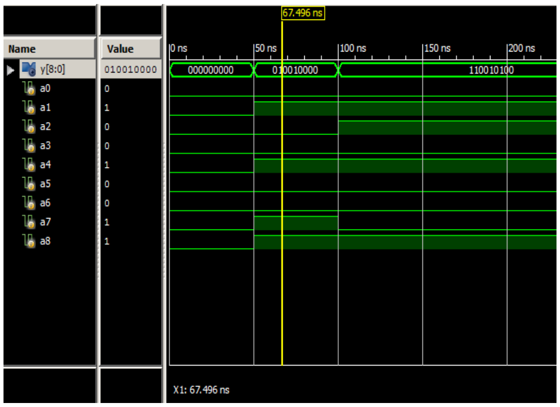


Fig. 5.Parity-based detection mechanism.

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 ∈ G F((23)3) be an element in composite fields, as shown in Fig. 4(b). Then, the predicted parity of the inverse transformation matrix, i.e., Pˆ Y , can be derived as follows considering the derived and partly discarded seven-bit row vector representing Y ∈ G F(29) Pˆ Y = b3 + b4 + b8. (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., Pˆ 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 Pˆ Y,0 is for even entries and Pˆ Y,1 is for odd entries Pˆ Y,0 = b1 + b2 + b4 + b7 (17) Pˆ Y,1 = b1 + b2 + b3 + b7 + b8. (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 Oi , 1 ≤ i ≤ n in Fig. 5) have n actual parities which are derived by modulo-2 addition of output bits (denoted by P(Oi), 1 ≤ i ≤ n in Fig. 5). The derived actual parities for the subblocks are XORed with the derived predicted parities (denoted by P Pi , 1 ≤ i ≤ n in Fig. 5) and the error detection flags (fi , 1 ≤ i ≤ 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 bitinterleavedactual/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 v. 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 JC0. 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

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 stuckat 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 cryptoarchitectures 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.

SIMULATION RESULTS:



Name	Value
y[8:0]	010010000
a0	0
a1	1
a2	0
a3	0
a4	1
a5	0
a6	0
a7	1
a8	1

fig wave form for cryptographic

8.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 structureindependent 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 highperformance 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 highperformance 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.

paper : swatchi