12-2016

Reliable Hardware Architectures for Cyrtographic Block Ciphers LED and HIGHT

Srivatsan Subramanian
ss7725@rit.edu

Reliable Hardware Architectures for Cryptographic Block Ciphers LED and HIGHT

by

Srivatsan Subramanian

A Gradute Paper Submitted

in

Partial Fulfillment

of the

Requirements for the Degree of

Master of Science

in

Electrical Engineering

Approved by:

Prof. _____
(Graduate Paper Advisor-Dr. Mehran Mozaffari-Kermani)

Prof. _____
(Department Head-Dr. Sohail A. Dianat)

Department of Electrical and Microelectronic Engineering

Kate Gleason College of Engineering

Rochester Institute of Technology

Rochester, New York

December, 2016

Acknowledgements

I would like to express my sincere gratitude to my advisor, Dr. Mehran Mozaffari Kermani, for his excellent guidance, support, patience and encouragement which helped me in successfully completing my research and graduate study at Rochester Institute of Technology. I also take this opportunity to thank Prof. Mark A. Indovina, Dr. Dorin Patru, Dr. Marcin Lukowiak and Dr. Amlan Ganguly for their immeasurable amount of guidance and assistance they have provided throughout my graduate program. Finally, I extend my deepest gratitude to my parents for their unceasing love, support and encouragement.

This graduate thesis is dedicated to my family.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other University. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specifically indicated in the text.

Srivatsan Subramanian

December 2016

Abstract

Cryptographic architectures provide different security properties to sensitive usage models. However, unless reliability of architectures is guaranteed, such security properties can be undermined through natural or malicious faults. In this thesis, two underlying block ciphers which can be used in authenticated encryption algorithms are considered, i.e., LED and HIGHT block ciphers. The former is of the Advanced Encryption Standard (AES) type and has been considered areaefficient, while the latter constitutes a Feistel network structure and is suitable for low-complexity and low-power embedded security applications. In this thesis, we propose efficient error detection architectures including variants of recomputing with encoded operands and signature-based schemes to detect both transient and permanent faults. Authenticated encryption is applied in cryptography to provide confidentiality, integrity, and authenticity simultaneously to the message sent in a communication channel. In this thesis, we show that the proposed schemes are applicable to the case study of Simple Lightweight CFB (SILC) for providing authenticated encryption with associated data (AEAD). The error simulations are performed using Xilinx ISE tool and the results are benchmarked for the Xilinx FPGA family Virtex-7 to assess the reliability capability and efficiency of the proposed architectures.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

To provide different security properties efficiently, lightweight cryptographic implementations on different hardware platforms have been emerged due to the advancement of constrained devices. These nodes require low-complexity implementations over small chip area and consume low amount of energy. Nevertheless, the Advanced Encryption Standard (AES), the current symmetric-key cryptography standard, may not achieve such tight necessities in terms of performance and implementation metrics. Thus, lightweight security mechanisms through low-complexity implementations of cryptographic algorithms are needed. We note that there have been constant, prominent efforts to realize the AES lightweight, an example for which is a 128-bit AES that was developed over an area of 2,400 gate equivalent [1]. It is noted that the AES architecture in [1] has been towards considerable area reductions; nonetheless, it is still considered a burden for resource-constrained applications such as radio-frequency identification (RFID) tags, nano-sensor nodes, and applications such as implantable and wearable medical devices. Furthermore, the inability of the AES to adapt to the varying level of security needed by different devices might be inefficient in case lower number of bits need to be protected.

In recent years and based on the above motivation, a number of lightweight block ciphers have been proposed, e.g., KATAN and KTANTAN [2], PICCOLO [3], and PRESENT [4], SEA [5], LED [6], Simon and Speck [7–10], Midori [11], HIGHT [12], and the like. Based on such ciphers, an open competition for a new Authenticated Encryption algo-

rithm (CAESAR) [13] has been initiated and will identify a portfolio of authenticated ciphers that offer advantages over AES-Galois Counter Mode (GCM) and are suitable for widespread adoption. These algorithms, e.g., SILC: SImple Lightweight CFB [14] that employs authenticated encryption with associated data, use encryption/decryption blocks as underlying structures in which the aforementioned block ciphers can be used. Authenticated Encryption provides authenticity and privacy to the data by first converting the plaintext to ciphertext and an authentication tag, message authentication code (MAC). Some of the authenticated encryption candidates such as SILC is of Encrypt-then-MAC type. In the Encrypt-then-MAC type, the authentication tag or MAC is created based on the resulting ciphertext. The MAC is provided as an input to the decryption algorithm and it is compared with the actual tag produced in the decryption. If both of the tags do not match, then authentication fails and if they match, then the output of the decryption algorithm will be the original plaintext.

It is imperative to note that although cryptographic algorithms, e.g., authenticated encryption which preserves authenticity and confidentiality of the message sent by the sender, provide different security mechanisms, natural and malicious faults can undermine such purpose. Let us go over different fault models we have considered in this thesis. We consider both single and multiple stuck-at faults because both are relevant with respect to intentional and natural faults, i.e., fault attackers prefer to ideally be able to inject single faults but, in reality, due to lack of technological advancements, multiple faults might occur, whose protection mechanisms are required. Moreover, natural faults can be of single nature, e.g., single event upsets, or multiple defects. Furthermore, we consider both transient and permanent faults. The attackers typically inject transient faults to gain as much information as they desire without breaking the cryptosystems; however, natural defects could be of permanent nature; thus, we consider them as well.

Various types of fault injection mechanisms such as temperature attacks, optical attacks, electromagnetic fault injection, and the respective countermeasures to such attacks are presented to date. Concurrent error detection (CED) techniques have been widely used to architect reliable hardware for the cryptographic algorithms [15–23] (including

a number of schemes, e.g., hardware/information/time/hybrid redundancy). Hardware redundancy makes use of 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 [24]. Time redundancy technique has a number of schemes, e.g., recomputing with shifted operands (RESO) [25, 26], recomputing with rotated operands (RERO) [27], and recomputing with permuted operands (REPO) [28]. The hybrid redundancy scheme is given in [29–31] where different improvements in the architecture have been proposed.

In this thesis, we consider two block ciphers which can be used as part of SILC, i.e., light encryption device (LED) [6], an AES-based block cipher and high security and lightweight (HIGHT) [12]. LED has 64-bit block length and uses 64-bit key length and re-uses the S-box of PRESENT block cipher [4]. HIGHT is of generalized Feistel type network and has 64-bit block length and 128-bit key length. HIGHT is suitable for embedded CPUs that are used in the nano-sensor network systems. SILC uses a combination of both cipher feedback (CFB) and cipher block chaining (CBC) for encryption and decryption.

In this thesis, we propose error detection approaches for block ciphers LED and HIGHT, considering the reliability and performance metrics objectives. Signature-based approaches are used in conjunction with the proposed error detection schemes based on recomputing with encoded operands to achieve high efficiency, while maintaining high error coverage.

1.1 Thesis Outline

The structure of the thesis is as follows:

- Chapter 2: This chapter explains the block ciphers LED and HIGHT.
- Chapter 3: In this chapter, we present the proposed hardware and time redundant error detection approaches for LED and HIGHT block ciphers.
- Chapter 4: The fault model used for testing the proposed designs is discussed in this chapter. The chapter also presents the error coverage results of both hardware and time redundant error detection schemes.
- Chapter 5: The FPGA benchmarks for Xilinx Virtex-7 for the proposed architectures is discussed in this chapter.
- Chapter 6: Conclusions and possible future work are presented in this chapter.

Chapter 2

Preliminaries

In this chapter, we briefly explain the block ciphers LED and HIGHT. Then, in the next chapter, the proposed error detection approaches are presented.

2.1 LED Block Cipher

LED is a 64-bit block cipher. It is of the AES type and is a substitution permutation network (SPN). LED has a nonce length l_N of 8 bytes and tag length τ. It re-uses S-box of PRESENT block cipher and has a 64-bit key (LED block cipher can be used as one of the underlying block ciphers E_K for SILC, as a usage model). The input plaintext which is of 64-bit length is arranged in a 4×4 array matrix called cipher *state* matrix. The cipher *state* matrix of the LED block cipher along with the key matrix are governed by an irreducible polynomial in $GF(2^4)$. The implemented LED block cipher uses a 64-bit key. Both the cipher *state* matrix and the key are arranged in 4×4 matrix in the form of 16 four-bit nibbles. Each entity in the cipher *state* matrix and the key matrix is of 4-bit length.

The first operation is the addition of round key denoted by addRoundKey($state$, K_i). This operation is performed on cipher *state* matrix and key matrix K_i. LED block cipher has a second operation called *step* function, that is responsible for providing enhanced security. This operation comprises of four iterative rounds for encryption in a sequential manner. In LED, each round consists of sequential set of four operations. For the 64-

bit key array matrix, addRoundKey and *step* operations are repeated for eight times. The LED cipher has 32 rounds of iteration for encrypting the plaintext. The sequence of operations that are carried out on the output of the first step are AddConstants, SubCells, ShiftRow and MixColumnsSerial, more details are presented in the next chapter.

2.2 HIGHT Block Cipher

HIGHT is a 64-bit block cipher with 128-bit key length proposed. HIGHT is a variant of Feistel network and it has 32 iterative rounds to complete the encryption process. It has 64-bit plaintext and 128-bit master key as its inputs and a 64-bit ciphertext as its output. It has been claimed that the hardware implementation of HIGHT is more efficient than the AES by justifying that, it had consumed 3,048 gates in $0.25\mu m$ technology.

The plaintext input is represented by $P = P_7||.....P_1||P_0$, where $P_0...P_7$ are each of eight bits length, and $'||'$ denotes concatenation operation and the plaintext P is of 64 bits length. For each round $i = 0, ..., 32$, the 64-bit intermediate values are represented by $X_i = X_{i,7}||....X_{i,1}||X_{i,0}$. The 64-bit ciphertext output is represented as $C = C_7||....||C_1||C_0$ and the master key which is of 128 bits length is denoted as $MK = MK_{15}||...||MK_0$, more details are presented in the next chapter when we go over the proposed error detection schemes.

Chapter 3

Proposed Reliable Architectures for LED and HIGHT

In this chapter, the error detection schemes used for detecting the transient and permanent faults in the LED and HIGHT block ciphers are presented.

3.1 Motivations

In what follows, we present the motivations in presenting the approaches for error detection of LED and HIGHT block ciphers. Then, in the next sub chapter, the proposed methods are presented. The CED techniques have been widely used to architect reliable hardware for the cryptographic algorithms [15–23] (including diverse schemes, e.g., hardware/information/time/hybrid redundancy). Let us review and analyze the existing fault diagnosis schemes here. The use of variants of parity for error detection is effective; nevertheless, one needs to utilize/tailor such approaches so that they are flexibly used, depending on the error detection capability requirements and overhead tolerance (we have proposed such flexibility when applicable in this thesis). Non-uniform error detection due to using parity can be alleviated using robust codes at the expense of relatively higher overhead. If the high area/power overhead is the burden to the usage model, one can alleviate that through the time redundancy approaches, deteriorating the performance metrics. We have considered in this thesis variants of time redundancy mechanisms which

can detect both permanent and transient faults.

The high overhead of hardware redundancy (area/delay overhead of duplication is roughly 100%), the high delay overhead and inability to detect permanent faults of conventional time redundancy (which is usually the case for hardware failures), the need for flexibility in terms of overhead and reliability compromise for information redundancy, and the need for appropriate choice for the specific techniques within hybrid redundancy pool have been motivations for the proposed work.

This is the first work presenting efficient error detection approaches for LED and HIGHT. We have closely considered two main criteria in choosing such approaches: (i) applicability of the approaches to the specific algorithms needs to be considered closely; for instance, while the FPGA realization of the S-boxes through look-up tables is efficient (which also affects the error detection, as presented in this thesis for LED), one may choose to realize them on ASIC through logic-gate approaches (which needs realization of signatures as logic gates instead of storing them in memories), and (ii) the level of granularity of the check points can be dynamically chosen, which is dependent on the overhead tolerance and the reliability requirements for such ciphers. In proposing the error detection techniques here, we have considered both of these. The merit of the proposed approaches is (a) the proposed algorithms are oblivious of the implementation platform (unlike some previous works which use specific resources of platforms), (b) they can be tailored based on the reliability/overhead compromise, e.g., the scheme for MixColumnsSerial of LED, and (c) they are flexible in terms of overhead of metrics, and the reliability goals (alternate approaches are proposed to ensure such requirements are achieved, e.g., the recomputing with encoded operands scheme of HIGHT).

3.2 Error Detection for LED Block Cipher

The overall flow and the top view of the error detection for the LED block cipher is presented here, followed by the details on the approaches. We have chosen using signature-based schemes for LED, especially because of the structures of SubCells and

MixColumnSerial, for which signature-based schemes provide not only flexibility in error detection but also achieve high error coverage. The overall error detection architecture is composed of the predicted/actual signatures and their comparisons to derive the error indication flags (the derived predicted signatures of these four operations are compared with the actual ones to get the error indication flags).

Table 3.1: (Interleaved) predicted parity of the S-Box for the LED block cipher.

x	0	1	2	3	4	5	6	7
$\hat{P}_{s[x]}$	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(1)
$\hat{P_{s[x]}}$ (Int.)	(11)	(00)	(11)	(01)	(11)	(00)	(00)	(10)
x	8	9	A	B	C	D	E	F
$\hat{P_{s[x]}}$	(0)	(1)	(0)	(1)	(1)	(1)	(1)	(1)
$\hat{P_{s[x]}}$ (Int.)	(11)	(01)	(00)	(10)	(01)	(10)	(01)	(10)

3.2.1 AddConstants and ShiftRow

In AddConstants operation, the round constant matrix is given by the round constant matrix of $\begin{bmatrix} 0 & (rc_5||rc_4||rc_3) & 0 & 0 \\ 1 & (rc_2||rc_1||rc_0) & 0 & 0 \\ 2 & (rc_5||rc_4||rc_3) & 0 & 0 \\ 3 & (rc_2||rc_1||rc_0) & 0 & 0 \end{bmatrix}$. The round constants are given by six round-specific bits, i.e., $rc_0, rc_1, rc_2, rc_3, rc_4, rc_5$. For each round, the value of round constants are made zero and the values of the six bits are updated before the AddConstants function. This round constant matrix is bit-wise XORed with the output of the *state* matrix from the addRoundKey operation and the value of the *state* matrix is updated.

The second column of the round constant matrix depends on the round in which the matrix is used. For instance, in Round 1, the value of the round constants are $rc_5 = 0, rc_4 = 0, rc_3 = 0, rc_2 = 0$ and $rc_0 = 1$. For the proposed signature-based approach, we can derive the predicted signatures of such a matrix, e.g., the predicted parity of the last two columns are zero and the one for the second column is also zero. The reason for such derivation is that $(rc_5||rc_4||rc_3) \oplus (rc_2||rc_1||rc_0) \oplus (rc_5||rc_4||rc_3) \oplus (rc_2||rc_1||rc_0)$ leads

to always zero value verifying that the predicted parity is zero. For the first column, modulo-2 adding of the elements to derive the predicted parities also results in zero. Thus, the predicted parity vector for the round constant matrix is $\hat{P} = [0, 0, 0, 0]$ (we use hat notations for predicted signatures). Thus, we can conclude that the derivation of the predicted parity is free in hardware for the AddConstants step.

Interleaved parity for burst faults: One can also derive the interleaved parities of the AddConstants step to account for burst faults. Burst faults are adjacent faults that can affect the output in case of both malicious and natural faults. Let us denote on the round constant matrix the rows through which we derive two interleaved parities, i.e., even/odd rows as $\begin{bmatrix} 0 & (rc_5||rc_4||rc_3) & 0 & 0 \\ \mathbf{1} & (\mathbf{rc_2}||\mathbf{rc_1}||\mathbf{rc_0}) & \mathbf{0} & \mathbf{0} \\ 2 & (rc_5||rc_4||rc_3) & 0 & 0 \\ \mathbf{3} & (\mathbf{rc_2}||\mathbf{rc_1}||\mathbf{rc_0}) & \mathbf{0} & \mathbf{0} \end{bmatrix}$. By modulo-2 adding such rows, we derive the interleaved parity vectors as $\hat{P}_1 = \hat{P}_2 = [2, 0, 0, 0]$. The reason for such derivation is that for the last three columns, the values are canceled once modulo-2 added and for the first column we have $\{0\}_{16} + \{2\}_{16} = \{1\}_{16} + \{3\}_{16}$.

The ShiftRow operation shifts the rows of the resulting *state* matrix from the SubCells operation with respect to its row number. If the operation is performed on the first row, then the first row is shifted to the left by 1 position and likewise for the second row, the contents of the *state* matrix are shifted by two positions. For the ShiftRow step, derivation of the predicted signatures is straightforward, e.g., parity prediction is free in hardware as rewiring does not change the predictions.

3.2.2 SubCells

The SubCells operation updates the value of the *state* matrix by replacing the contents of the *state* matrix, according to the S-Box. For detecting the errors in S-Box, we devise a signature-based scheme illustrated through two case studies, i.e., parity prediction and interleaved (*Int.*) parity prediction as shown in Table 3.1. In this method, the output bits of the S-box are XORed (and for the case of interleaved parity, odd and even bits

are modulo-2 added separately) and the output of the S-box is appended by one bit (two bits for the interleaved case). For example, if the output is $\{C\}_{16}$ which is $\{1100\}_2$, that means the predicted parity of is 0 and the interleaved parity pair is 11 (see Table 3.1).

3.2.3 MixColumnsSerial

MixColumnsSerial uses a hardware-friendly matrix known as MDS, that is given by (this matrix has been changed from the original construction in [32]) $M = \begin{bmatrix} 4 & 1 & 2 & 2 \\ 8 & 6 & 5 & 6 \\ B & E & A & 9 \\ 2 & 2 & F & B \end{bmatrix}$. The updated *state* matrix from the previous operation is multiplied with the M matrix and the resulting *state* matrix is updated column-wise.

For presenting our signature-based scheme, let us denote the input and output *state* matrices as $A = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 & a_7 \\ a_8 & a_9 & a_a & a_b \\ a_c & a_d & a_e & a_f \end{bmatrix}, R = \begin{bmatrix} r_0 & r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 & r_7 \\ r_8 & r_9 & r_a & r_b \\ r_c & r_d & r_e & r_f \end{bmatrix}$. Each entity in the input and output *state* matrices is a four-bit nibble. In the MixColumnsSerial step, each column of the output matrix R is the product of the matrix M and the *state* matrix A. As a case study, to compute r_0, the first element of the resultant matrix R, denoting the bits of the elements of A as a_{ij} for ith bit jth element, we have (using the irreducible polynomial utilized for reductions, and not presenting the details for the sake of brevity): $r_0 = 4.a_0 + a_4 + 2.a_8 + 2.a_c = \cdots = x^3[a_{2c} + a_{28} + a_{14} + a_{30}] + x^2.[a_{10} + a_{40} + a_{24} + a_{38} + a_{3c}] + x.[a_{10} + a_{20} + a_{34} + a_{18} + a_{48} + a_{1c} + a_{4c}] + 1.[a_{20} + a_{44} + a_{1c} + a_{18}]$.

One can derive the formulae for the column signatures of the MixColumnsSerial by modulo-2 adding those for each column, e.g., the first column for r_0, r_4, r_8, and r_c, whose details are not presented for the sake of brevity. Adding modulo-2 the first column of matrix M, one can derive the following signature: $r_0 \oplus r_4 \oplus r_8 \oplus r_c = (4 \oplus 8 \oplus B \oplus 2).a_0 + (1 \oplus 6 \oplus E \oplus 2).a_4 + (2 \oplus 5 \oplus A \oplus F).a_8 + (2 \oplus 6 \oplus 9 \oplus B).a_c = 5.a_0 + B.a_4 + 2.a_8 + 6.a_c$.

This can be generalized to other columns and thus we have the followings for the

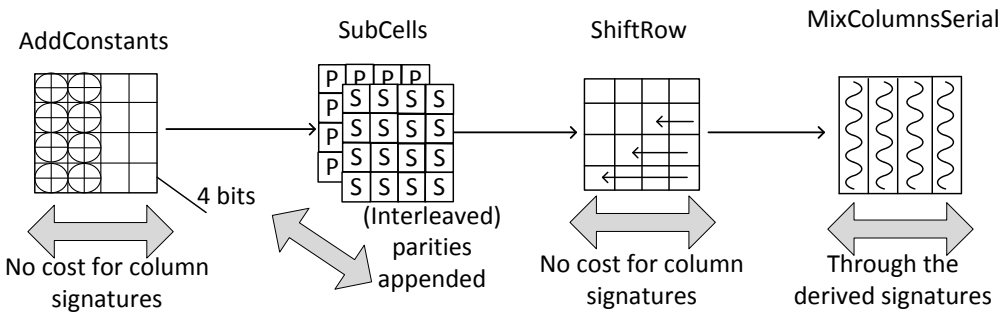

Figure 3.1: The proposed fault diagnosis scheme for the LED block cipher.

second to fourth columns: $r_1 \oplus r_5 \oplus r_9 \oplus r_d = 5.a_1 + B.a_5 + 2.a_9 + 6.a_d, r_2 \oplus r_6 \oplus r_a \oplus r_e = 5.a_2 + B.a_6 + 2.a_a + 6.a_e, r_3 \oplus r_7 \oplus r_b \oplus r_f = 5.a_3 + B.a_7 + 2.a_b + 6.a_f$.

As seen in Fig. 3.1, we have denoted the four operations in LED by the fault diagnosis mechanisms for different sub-blocks. As seen in this figure, the AddConstants parity derivation is cost-free in hardware and that of ShiftRow follows the same details. One can use the derivations for SubCells operations in Table 3.1 as well as those for MixColumnsSerial. The overall error detection architecture is composed of the predicted and actual signatures and their comparisons to derive the error indication flags. In more details, the derived predicted signatures of these four operations are compared (XORed) with the corresponding, actual ones to get the error indication flags (which can be ORed to derive one flag). Based on the reliability requirements and performance and implementation metrics overhead tolerance, one can tailor the proposed approaches.

3.3 Error Detection for HIGHT Block Cipher

In this chapter, we first present the error detection schemes for the "Initial/Final Transformations and Round Function" through Two Pair Two Rail Checker (TPTRC). Then, an alternate method based on recomputation is presented. The TPTRC scheme, inherently, is able to detect both transient and permanent faults. Transient single stuck-at faults are among the ideal cases for fault attackers; however, in practice, multiple (and adjacent) faults occur. These are detected using the presented scheme and more details are presented in the next chapter. The signature-based diagnosis approach, which uses linear codes that can (always) detect random errors of small multiplicity (and can never

detect some other errors); is diverse from an architecture based on robust codes which can detect (with probability) any error. We also go over an alternate approach based on recomputation with encoded operands which has lower area complexity and power consumption, at the expense of lower performance. The choice among these two schemes depends on the overhead tolerance for specific applications. The latter scheme, similar to the former one, detects both transient and permanent faults. We note that permanent faults need to be detected as they might occur through VLSI defects; however, the attackers are not interested to mount the attacks through such damaging faults.

3.3.1 Initial/Final Transformations and Round Function

The 64-bit plaintext is given as the input to the initial transformation function. It uses the plaintext and Whitening Keys WK_3, WK_2, WK_1, WK_0 as its input and generates the output as intermediate values $X_0 = X_{0,0}..X_{0,6}, X_{0,7}$ for the first round. The Whitening Keys WK_3, WK_2, WK_1, WK_0 are fed by the Key Scheduling Algorithm. The Initial Transformation algorithm consists of XOR operations denoted by $\oplus$ and modular addition represented by $\boxplus$ (note that for decryption, this is replaced by subtraction $\boxminus$).

In the final transformation, the input is the output of the last round X_{32} and the other inputs are the Whitening Keys Wk_7, WK_6, WK_5, WK_4. The output generated by this algorithm is the ciphertext, given by C, that is of 64 bits length and it is concatenated and represented as $C_0||....C_6||C_7$.

The round function is an iterative process for the 32 rounds and it plays a vital role in providing enhanced security. The operations carried out in the round function are modular addition and XOR operation. The input of the round function is the output of the previous round X_i and the four SubKeys $SK_{4i+3}, SK_{4i+2}, SK_{4i+1}, SK_{4i}$ generated per each round. The round function algorithm (Algorithm 3.1) generates the 64-bit output X_{i+1} and uses the auxiliary functions F_0 and F_1 to compute the output $X_i = X_{i+1,0}||....X_{i+1,6}||X_{i+1,7}$ concatenation of eight bytes each. The auxiliary functions use the operation $x^{<<<1}$, which is a representation of 1-bit left rotation of the 8-bit value x. The output of the rotated value is bitwise XORed or modulo-2 added depending on the

Algorithm 3.1 Round function in the HIGHT block cipher.

RoundFunction(X_i,X_{i+1},SK_{4i+3}, SK_{4i+2},SK_{4i+1}, SK_{4i}) {
$X_{i+1,1} \leftarrow X_{i,0}$; $X_{i+1,3} \leftarrow X_{i,2}$; $X_{i+1,5} \leftarrow X_{i,4}$;
$X_{i+1,7} \leftarrow X_{i,6}$;
$X_{i+1,0} = X_{i,7} \oplus (F_0(X_{i,6}) \boxplus SK_{4i+3})$;
$X_{i+1,2} = X_{i,1} \boxplus (F_1(X_{i,0}) \oplus SK_{4i+2})$;
$X_{i+1,4} = X_{i,3} \oplus (F_0(X_{i,2}) \boxplus SK_{4i+1})$;
$X_{i+1,6} = X_{i,5} \boxplus (F_1(X_{i,4}) \oplus SK_{4i})$;}

input. The output of the last round is fed to the input of the final transformation. We note that $F_0(x) = x^{<<<1} \oplus x^{<<<2} \oplus x^{<<<7}$ and $F_1(x) = x^{<<<3} \oplus x^{<<<4} \oplus x^{<<<6}$.

For modular addition operation, we choose self-checking carry select adder because they are fast and have relatively low complexity (for detecting both permanent and transient stuck-at faults). Generally, it is well known that carry select adders contain two ripple carry adders and multiplexers. In addition to that hardware resource, as shown in the Fig. 3.2, this self-checking carry select adder uses XNOR gates and TPTRC as an additional hardware resource to detect any single stuck at fault [33]. As seen in Fig. 3.2, the size of the adder is arbitrary up to n-bits. Here, for our addition operation, the operands are of 64 bits size and, hence, we use 64-bit adder to perform the operations. The adder consists of cascaded ripple carry adders that can add up to two bits at a time and then the value of the carry out is rippled through a multiplexer. This carry out acts as the value of the actual carry-in to the next set of cascaded ripple carry adders. The carry select adder pre-computes the values of the sum bits before knowing the value of the actual carry-in. Once the value of the actual carry-in is known, the appropriate sum bits and carry-out of the carry select adder is given as the output by the carry select adder.

Self-checking multiplexers and TPTRC are used in the proposed architectures. The TPTRC has two pairs of inputs (x_0, y_0) and (x_1, y_1). In the fault-free condition, the input pairs to the TPTRC are complementary to each other, i.e., $x_0 = y_0'$ and $x_1 = y_1'$. If there is no fault, then the output pair of the TPTRC are also complementary to each other, i.e., $t = t_0'$. Finally, the output pair of TPTRC are given to the XNOR gate and if there is a fault, the error indication flag is raised to high. The valid input code words for the

TPTRC are 10 and 01. In case of a single stuck at fault in any one of the internal path of the ripple carry adder, the input pairs to the TPTRC will be 11 and 00. This will result in a non-valid output and error indication flag is raised. Such an adder can be used to detect faults in the HIGHT block cipher.

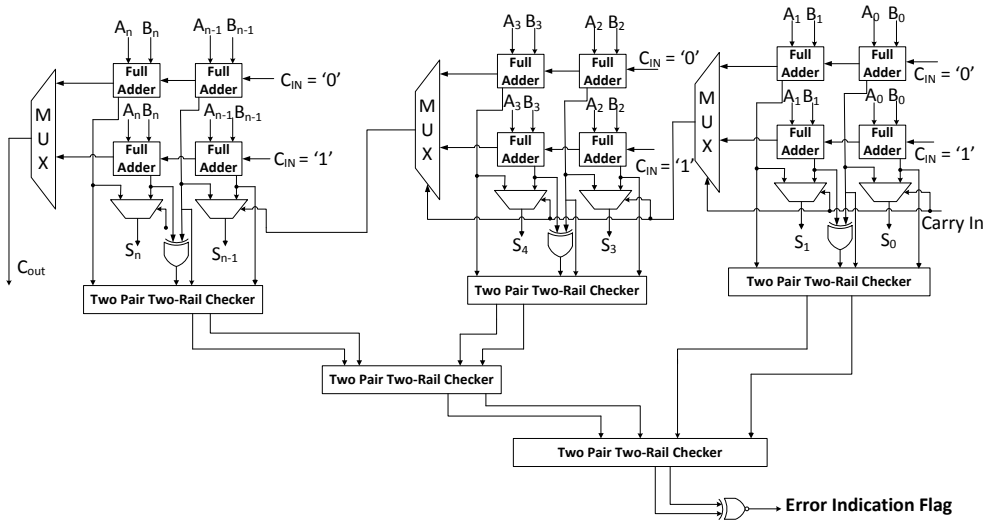

Figure 3.2: Error detection in HIGHT through self-checking carry select adder.

A variant of n-bit model of the self-checking carry select adder based on dual rail encoding has been proposed in [34]. This variant adder includes an additional circuitry of AND gates in addition to the hardware used in Fig. 3.2 To compute the sum bits for the n-bit adder in a dual rail form with valid codewords (10 and 01), AND gates are used in Fig. 3.2. If S_{0n} is the sum bit computed by the full adder for carry-in of "0" and S_{1n} is the sum bit computed by full adder for carry-in of "1", then XNOR operation is performed between sum-bit S_{0n} and all the lower sum-bits. The output of XNOR gate is fed to x_0 in the TPTRC input pair (x_0,y_0), and y_0 is connected to S_{1n}. Similarly, the other input pair (x_1,y_1) to TPTRC is connected in such a way that x_1 is connected from the output of XNOR gate and S_{0n-1}. The sum bit S_{0n-1} is computed by the full adder for $n-1$ inputs with a carry-in of "0". The sum bit S_{1n-1} is connected to y_1 of TPTRC. The input pairs are in dual-rail form, that is, the input pairs will be always complementary to the TPTRC, i.e., $x_0 = y_0'$ and $x_1 = y_1'$.

Finally, as seen in Fig. 3.3, we have shown the fault diagnosis of the round function of

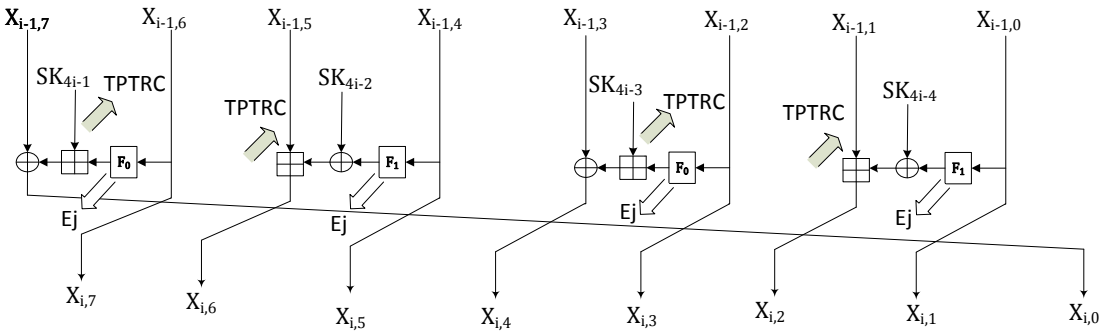

Figure 3.3: Error detection for the round function sub-block for HIGHT.

HIGHT. As seen in this figure, for the auxiliary functions F_0 and F_1 within round function to compute the output $X_i = X_{i+1,0}||....X_{i+1,6}||X_{i+1,7}$, we use signature-based schemes to derive the error indication flags. In more details, the derived predicted signatures of the operations are compared (XORed) with the corresponding, actual ones to get the error indication flags (which can be ORed to derive one flag). Specifically, for the modular addition, the presented TPTRC approach is utilized (see Fig. 3.3); furthermore, for the auxiliary functions F_0 and F_1, we use signature-based schemes. Thus, for the modular addition operation, the proposed scheme based on Fig. 3.3 is utilized. Finally, Fig. 3.4 presents the entire error detection approach for the encryption of HIGHT.

3.3.2 Alternate Approach: Recomputing with Encoded Operands

Concurrent error detection can be performed through recomputing with encoded operands in such a way that the operations are computed twice, one for the normal operands and one for the encoded operands. If an n-bit operand is encoded, e.g., rotated left or right by k bits during the recomputation step, no bit is lost and the scheme utilizes the sizes of adders, ALUs, and registers increased only by 1 bit. Such an alternate approach, similar to the previous one, detects both transient and permanent faults in the HIGHT block cipher. Using such a recomputation, one can efficiently detect $k \ mod \ n$ consecutive logical errors and $k \ mod((n+1)-1)$ in arithmetic operations, where n is the length of arithmetic operations and k bits is the number of bits to be rotated. We have employed such an approach for the HIGHT block cipher. As the operations involved in the HIGHT block cipher are addition mod $GF(2^8)$ and XOR, such an alternate scheme is efficiently

applicable as shown through our error simulations and FPGA implementations.

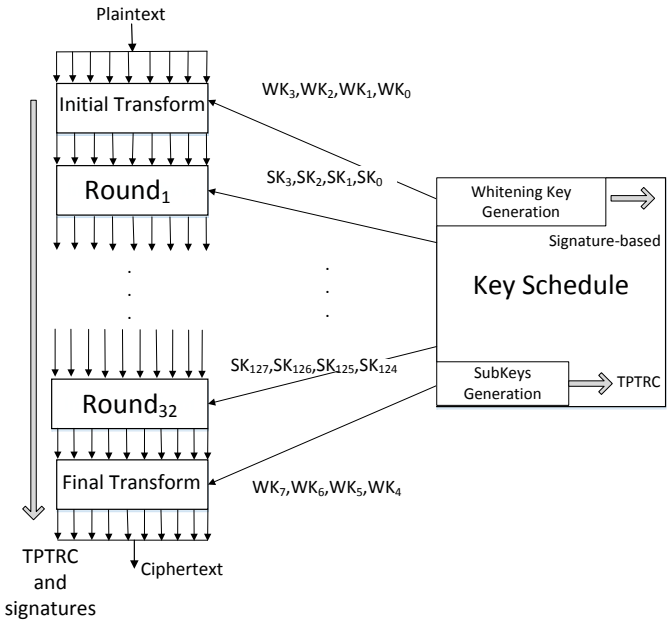

Figure 3.4: Error detection for the entire encryption of HIGHT block cipher.

Let ϕ and ϕ' be the n-bit rotation and "back-rotation" functions, respectively. Let θ be the input to the arithmetic function f, such that $f(\theta)$ is the output of the arithmetic function. The aim is to satisfy $\phi'(f(\phi(\theta)) = f(\theta)$. The operands in the HIGHT block cipher are run twice. For the first run, the original operands are passed and the result is computed and stored in a register. For the second run, rotated operands (right or left cyclic shift) are computed and the result is compared with the first result that is stored in a register. If there is not a match, then error is detected. For addition operations, to ensure the correctness of the carry-in for $k+1$th bit and carry out from $n-1$th bit, we add an extra bit to the most significant bit (*) position using an $n+1$ adder. This bit is always set to "0" (or stuck at zero). Thus, as a result of rotation, the logic $n-1$th bit does not interfere with the logic 0th bit in the rotated position as well as in the normal position. During normal computation, the most significant bit is 0; however, during the recomputation step, the most significant bit is the ith bit. Because the most significant bit is always stuck at zero, during recomputation, the carry-out from the ith bit is given to the carry-in of the $i+1$th bit. In normal computation, as the value of the carry-out is always 0, it does not change the ith bit.

In what follows, we use the case study of SILC to show how the proposed approaches can be utilized; nevertheless, this does not confine the presented schemes. SILC has a fixed block length n and uses a block cipher E such that $\kappa_E \times \{0,1\}^n \rightarrow \{0,1\}^n$. We note that HIGHT and LED can both be used for such a construction. SILC comprises of two algorithms for encryption (SILC-ε_K) and decryption (SILC-D_K). The encryption SILC-ε_K algorithm consists of three subroutines HASH, ENC, and PRF. The subroutines are called in a sequential manner, i.e., the output of one subroutine is fed to the input of the successive subroutine. The decryption comprises of the same subroutines as the encryption algorithm, except that an additional comparison between the tag computed in the decryption algorithm and encryption algorithm is performed. The first step in the decryption algorithm is the computation of V (which is the result of hashing the inputs through the key). The next step is the computation of MAC in the decryption algorithm T^*(which is the PRF, using the key, of the hash result and the ciphertext). The third step is the comparison of the tags, and if they do not match, the algorithm returns the result $\perp$, meaning that the authentication is failed. The final step in the decryption algorithm is retrieving the original plaintext M.

3.3.3 Key Schedule

The key schedule for HIGHT comprises of two algorithms: Whitening Key Generation and SubKeys Generation. The main idea behind the key schedule algorithm is to preserve the master key. Whitening Key Generation algorithm is a subroutine that generates the eight whitening keys necessary for the Initial/Final Transformations. The SubKey Generation subroutine generates 128 SubKeys and supplies four SubKeys per each round. For Round 1, four SubKeys SK_3, SK_2, SK_1, SK_0 and for round 32, the SubKeys $SK_{127}, SK_{126}, SK_{125}, SK_{124}$ are generated by this subroutine. For SubKeys Generation algorithm, the generation of 128 7-bit constants from δ_0 to δ_{127} is done by the constant generation module. All the operations are carried out over $GF(2^8)$ governed by the "connection" polynomial $x^7 + x^3 + 1$. The hardware used for the constant generation is an LFSR (Linear Feedback Shift Register). The initial state of the 7-bit LFSR $'h'$ is fixed

at $(1011010)_2$.

Fault diagnosis of Whitening Key Generation is straightforward as it is rewiring in hardware, e.g., one can use signature-based schemes to have low-complexity error detection. For SubKeys Generation and its internal constant generation module, other that logic operations, modular addition is utilized whose fault diagnosis has been discussed in this chapter (through Fig. 3.3).

One can use the proposed algorithms in this thesis to derive the error indication flags of the underlying block ciphers in SILC. Let us consider two scenarios: If the error indication flags are raised, an incorrect ciphertext and tag is expected. In this case, not only the derived tag in decryption would be faulty, but a faulty ciphertext is transmitted to the receiver side. However, the tags match as authentication is not compromised (same ciphertext is used to derive both tags). In the second scenario, let the HASH or PRF functions are faulty, in which case the tags at the receiver side do not match and the algorithm returns the result $\perp$. In both cases, the error indication flags correctly alarm an incorrect ciphertext.

The work in [35, 36] presents differential fault analysis (DFA) attacks on HIGHT and LED. The authors used data leaked through random byte model (covered in our simulations for HIGHT) or random bit (for LED) to deduce the secret key (transient fault simulations are also presented in the next chapter). For instance, the sketch for attacking HIGHT includes three phases: (a) collection of right ciphertext and faulty ciphertexts, (b) the computation of the candidates of subkeys in select rounds and whitening key in the final transformation, and (c) the recovery of the 128-bit secret key from the candidates of subkeys and whitening key. If such attacks are successful by bypassing, for instance, the RERO scheme, we make a small architectural addition to our proposed scheme in order to detect such type of DFA attacks. Since the fault injections are made at the input of a round, we compare the input sub-cipher in each round (starting from second round) with that generated in previous round. Any discrepancies will be indicated by the error indication flag. Should the attacker try to inject faults in the sub-cipher in the previous round itself, the previously proposed RERO scheme will detect such an attack.

Thus, the RERO and the suggested addition should be able to protect the ciphers against permanent and transient faults and make the DFA attacks more difficult; however, we do not claim that it will be able to detect all types of DFA attacks, for instance, [37, 38].

Chapter 4

Error Simulations

In this chapter, we present the error simulations and FPGA implementations for LED and HIGHT block ciphers to benchmark their effectiveness.

To benchmark the effectiveness of the proposed schemes for both ciphers, we utilize LFSRs to inject stuck-at faults. The purpose of using LFSRs is to generate pseudo-random fault patterns for every clock cycle. The outputs of the LFSRs are XORed with a random bit and sent as the feedback to the input. In this way, random single and multiple stuck-at faults are injected in the design. We use the polynomial $x^6 + x^3 + 1$ for the implementation of the LFSRs. The LFSRs are used to inject 10,000 faults in LED block cipher and the fault coverage is found to be very close to 100% from the simulation results. In HIGHT and for recomputing with encoded operands approach, we have used the LFSRs with the same polynomials to inject the faults. In the first run, the actual operands are sent for computation of the output. In the second run, the operands that are rotated to the right are used to compute the output. In the third run, the output from the last round is rotated left and then compared with the output from the first run. If they do not match, then error indication flags are raised. The modular adders used in the round function module are also injected with single and multiple faults. By the use of LFSRs, around 10,000 single and multiple faults are injected in the HIGHT architecture, where, the results from the simulations for the overall structure shows a very high fault coverage of close to 100%.

The proposed methods, being for reliability, can deal with permanent and transient faults. To make sure we cover a good number of fault models, through injecting 10,000 faults in the architecture of HIGHT, we have also investigated transient faults, one/two/three-bit faults, byte faults, and two-byte adjacent faults. The simulations have been performed separately and the results show that (a) for transient faults, we have 9,989 detected faults leading to the error coverage of around 99.9%, (b) for one-bit faults, we have 9,854 detected faults leading to the error coverage of around 98.5%, (c) for two-bit faults, we have 9,890 detected faults leading to the error coverage of 98.9%, (d) for three-bit faults, we have 9,988 detected faults leading to the error coverage of around 99.9%, (e) for one-byte faults, we have 9,976 detected faults leading to the error coverage of around 99.8%, and finally, (f) for two-byte faults, we have 9,980 detected faults leading to the error coverage of 99.8%. We note that although these figures are very high, tailoring the proposed error detection schemes, one may achieve higher error coverage if the overhead is tolerated.

Chapter 5

FPGA Implementations and Benchmark

In this chapter, we present the area, power consumption, and delay overhead results for LED and HIGHT block ciphers through the FPGA implementations. The benchmarking is done for both the original and fault detection architectures of LED and HIGHT block ciphers. The FPGA implementations are carried out on Xilinx family Virtex-7 with target device 7vx330tffg1157-3.

The error detection schemes for LED are shown as LED-signature in Table 5.1. The area in terms of number of slices, delay, and power consumptions are derived for Virtex-7 by implementing the design at 100 MHz frequency. The overheads of the error detection architectures are shown in the parentheses in this table (using Xilinx Integrated Synthesis Environment (ISE) 14.7 version).

The results from the FPGA implementations of HIGHT-original and fault detection are shown in Table 5.2. The area overhead for the fault detection design is due to the inclusion of error detecting adders used for the recomputed operation. From the results, we can infer that there are negligible power and delay overheads for the architecture as

Table 5.1: FPGA implementations of LED on Virtex-7 (target device: 7vx330tffg1157-3).

Architecture	Area (slices)	Delay (ns)	Power (mW)
LED-original	178	5.841	2.93
LED-signature	217 (21.9%)	5.914 (1.2%)	3.67 (25.2%)

Table 5.2: FPGA implementations of HIGHT on Virtex-7 (target device: 7vx330tffg1157-3).

Architecture	Area (slices)	Delay (ns)	Power (mW)
HIGHT-original	191	2.686	2.15
HIGHT-proposed	252 (31.9%)	∼2.686	2.17 (0.01%)

well as acceptable area overhead (we note that such overheads can be adjusted based on the objectives, for instance, avoiding sub-pipelining could be a compromise between area/power and throughput).

The throughput degradations are also derived for the original and fault detection architectures. For the signature-based schemes, the throughput degradation is negligible, i.e., for LED we have 342.407 $Mbps$ for the original and 338.180 $Mbps$ for the fault detection structures leading to the degradation of 1.23. For the HIGHT algorithm, one can perform sub-pipelining to account for the inherent reduction in throughput. Utilizing one-stage sub-pipelining, at the cost of registers added, similar throughput to the original architecture is derived.

There has not been any prior work done on error detection methods for these ciphers to the best of our knowledge. In [39], the authors present fault diagnosis of Pomaranch cipher. They have used bit-interleaved scheme for error detection. We compare the overheads of Pomaranch with the proposed scheme. The combined area and throughput overhead for Pomaranch is 35.5%. The proposed schemes have combined area and delay overheads of 23% for LED and 31.9% for HIGHT, respectively. Since the architecture of Pomaranch and presented fault detection scheme is a lot different than the proposed method, the differences in the overheads are reasonably justified.

Chapter 6

Conclusion

Our research group has extensive experience in cryptographic engineering and fault diagnosis as well as editing IEEE Transactions journals [45-95]. In this thesis, signature-based and recomputing with encoded operand-based approaches are presented for the LED and HIGHT block ciphers. Formulae for the linear and non-linear sub-blocks of the LED block cipher are presented, tailoring which one can achieve the required reliability and overheads. We have also applied sub-pipelining to overcome the inherent throughput degradation of the proposed approaches for the HIGHT block cipher [90-92]. Such SPN-based and Feistel network-based block ciphers can be used as part of the authenticated encryption mechanisms such as SILC. Through fault-injection analysis, it has been shown that the error coverage is close to 100%. Moreover, through FPGA implementations, we have shown that acceptable overheads are achieved for both ciphers. Based on the reliability requirements and available resources, one may utilize the proposed error detection schemes for making the hardware implementations of LED and HIGHT algorithms more reliable.

6.1 Future Work

The proposed work is about the error detection of permanent and transient faults in LED and HIGHT block ciphers for the SILC. The time redundancy approach uses RERO for detecting the faults. However, the future work can be done using REPO, which is one

of the types that expliots time redundancy method. The efficiency of the fault detection architectures such as REPO can be also be tested using fault injection simulation schemes such as BIST. The future work can be done by implementing the cipher designs in ASIC platforms to see if the fault tolerant architectures can be suitable for low area or low power applications. The work can be further considered to test for transition faults (slow rise (0 to 1) and slow fall (1 to 0) faults) and bridging faults.

References

[1] A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang, "Pushing the limits: A very compact and a threshold implementation of AES," *Proc. Advances in Cryptology*, pp. 69 – 88, 2011.

[2] C. De Canniere, O. Dunkelman, and M. Knežević, "KATAN and KTANTAN - a family of small and efficient hardware-oriented block ciphers," *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 272–288, 2009.

[3] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai, "Piccolo: An ultra-lightweight blockcipher," *Proc. Cryptographic Hardware and Embedded Systems*, pp. 342 – 357, 2011.

[4] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoel, "PRESENT: An ultra-lightweight block-cipher," *Proc. Cryptographic Hardware and Embedded Systems*, pp. 450 – 466, 2007.

[5] F. X. Standaert, G. Piret, N. Gershenfeld, and J. J. Quisquater, "SEA: A scalable encryption algorithm for small embedded applications," *Proc. Smart Card Research and Advanced Applications*, pp. 222 – 236, 2006.

[6] J. Guo, T. Peyrin, A. Poschmann, and M. J. B. Robshaw, "The LED block cipher," *Proc. Cryptographic Hardware and Embedded Systems*, pp. 326 – 341, 2011.

[7] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "SIMON and SPECK: Block ciphers for the internet of things," *NIST Lightweight Cryptography Workshop*, 2015.

[8] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "The SIMON and SPECK families of block ciphers," *Proceedings of the 52nd Annual Design Automation Conference*, p. 175, 2015.

[9] A. Biryukov, A. Roy, and V. Velichkov, "Differential analysis of block ciphers SIMON and SPECK," *Proc. Fast Software Encryption*, pp. 546 – 570, 2014.

[10] S. Sun, L. Hu, P. Wang, K. Qiao, X. Ma, and L. Song, "Automatic security evaluation and (related-key) differential characteristic search: Application to SIMON, PRESENT, LBLOCK, DES(L) and other bit-oriented block ciphers," *Proc. Advances in Cryptology*, pp. 158–178, 2014.

[11] S. Banik, A. Bogdanov, T. Isobe, K. Shibutani, H. Hiwatari, T. Akishita, and F. Regazzoni, "Midori: A block cipher for low energy (extended version)," *Cryptology ePrint Archive*, p. 2015/1142, 2015. [Online]. Available: http://eprint.iacr.org

[12] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B.-S. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, and S. Chee, "Hight: A new block cipher suitable for low-resource device," *Proc. CHES*, pp. 46–59, 2006.

[13] "CAESAR:Competition for Authenticated Encryption: Security, Applicability, and Robustness." [Online]. Available: https://competitions.cr.yp.to/caesar.html.

[14] "SILC: Simple lightweight CFB." [Online]. Available: https://competitions.cr.yp.to/round2/silcv2.pdf.

[15] C. H. Yen and B. F. Wu, "Simple error detection methods for hardware implementation of advanced encryption standard," *IEEE Trans. Comput,*, vol. 55, no. 6, pp. 720–731, Dec. 2006.

[16] G. Di Natale, M. Doulcier, M.-L. Flottes, and B. Rouzeyre, "A reliable architecture for parallel implementations of the Advanced Encryption Standard," *J. Electronic Testing: Theory and Applications*, vol. 25, no. 5, pp. 269 – 278, Aug. 2009.

[17] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "Concurrent structure-independent fault detection schemes for the Advanced Encryption Standard," *IEEE Transactions on Computers*, vol. 59, no. 5, pp. 608–622, 2010.

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

[19] 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.

[20] 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.

[21] M. Yasin, B. Mazumdar, S. S. Ali, and O. Sinanoglu, "Security analysis of logic encryption against the most effective side-channel attack: DPA," *Proc. DFTS*, pp. 97–102, 2015.

[22] D. Karaklajic, J.-M. Schmidt, and I. Verbauwhede, "Hardware designer's guide to fault attacks," *IEEE Trans.Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 12, pp. 2295–2306, 2013.

[23] R. Karri, G. Kuznetsov, and M. Goessel, "Parity-based concurrent error detection of substitution-permutation network block ciphers," *Proc. Cryptographic Hardware and Embedded Systems*, pp. 113–124, 2003.

[24] J. Li and E. E. Swartzlander, "Concurrent error detection in ALUs by recomputing with rotated operands," *Proc. Defect and Fault Tolerance in VLSI Systems*, pp. 109–116, 1992.

[25] 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.

[26] J. H. Patel and L. Y. Fung, "Concurrent error detection in alus by recomputing with shifted operands," *IEEE Trans.Comput.*, vol. C-31, no. 7, pp. 589–595, 1982.

[27] J. Li and E. E. Swartzlander, "Concurrent error detection in alus by recomputing with rotated operands," *Proc. Defect and Fault Tolerance in VLSI Systems.*, pp. 109–116, 1992.

[28] 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.

[29] R. Karri, K. Wu, P. Mishra, and Y. Kim, "Concurrent error detection schemes of fault based side-channel cryptanalysis of symmetric block ciphers," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 12, pp. 1509–1517, 2002.

[30] A. Satoh, T. Sugawara, N. Homma, and T. Aoki, "High-performance concurrent error detection scheme for aes hardware," *Proc. CHES*, pp. 100–112, 2008.

[31] J. Rajendran, H. Borad, S. Mantravadi, and R. Karri, "SLICED: slide based concurrent error detection technique for symmetric block cipher," *Proc. HOST*, pp. 70–75, 2010.

[32] J. Guo, T. Peyrin, A. Poschmann, and M. J. B. Robshaw, "The LED block cipher," 2012. [Online]. Available: https://eprint.iacr.org/2012/600.pdf

[33] D. Vasudevan, P. Lala, and J. Parkerson, "Self-checking carry-select adder design based on two-rail encoding," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 12, pp. 2696 – 2705, Dec. 2007.

[34] M. Akbar and J. Lee, "Comments on self-checking carry-select adder design based on two-rail encoding," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 7, pp. 2212–2214, Jul. 2014.

[35] Y. Lee, J. Kim, J. H. Park, and S. Hong, "Differential fault analysis on the block cipher hight," *Proc. Future Information Technology, Application, and Service*, pp. 407–416, 2012.

[36] P. Jovanovic, M. Kreuzer, and I. Polian, "A fault attack on the led block cipher," *Proc. Int. Workshop,COSADE*, pp. 120–134, 2012.

[37] N. F. Ghalaty, B. Yuce, and P. Schaumont, "Differential fault intensity analysis on present and led block ciphers," *Proc. Int. Workshop COSADE*, pp. 174–188, 2015.

[38] S. Patranabis, A. Chakraborty, P. H. Nguyen, and D. Mukhopadhyay, "A biased fault attack on the time redundancy countermeasure for AES," *Proc. Int. Workshop, COSADE*, pp. 189–203, 2015.

[39] M. M. Kermani, R. Azarderakhsh, and A. Aghaie, "Reliable and error detection architectures of pomaranch for false-alarm-sensitive cryptographic applications," *IEEE Trans. VLSI Circuits and Systems*, vol. 23, no. 12, pp. 2804–2812, 2015.

[40] M. Nicolaidis, "Carry checking/parity prediction adders and ALUs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 1, pp. 121 – 128, 2003.

[41] J.-C. Daly, L. Jien-Chung, and M. Nicolaidis, "Design of static cmos self-checking circuits using built-in current sensing," pp. 104–11, 1992.

[42] S. Endo, T. Sugawara, N. Homma, T. Aoki, and A. Satoh, "An on-chip glitchy-clock generator for testing fault injection attacks," *Journal of Cryptographic Engineering*, vol. 1, no. 4, pp. 265–270, 2011.

[43] Y. Sung-Ming, S. Kim, S. Lim, and S. Moon, "A countermeasure against one physical cryptanalysis may benefit another attack," pp. 414–427, 2001.

[44] E. Fujiwara and K. Haruta, "Fault-tolerant arithmetic logic unit using parity-based codes," *IEICE Transactions (1976-1990)*, vol. 64, no. 10, pp. 653–660, 1981.

[45] M. Mozaffari-Kermani, E. Savas, and S. Upadhyaya, "Guest editorial: Introduction to the special issue on emerging security trends for deeply-embedded computing systems," *IEEE Transactions on Emerging Topics in Computing*, 2016.

[46] M. Mozaffari-Kermani, R. Azarderakhsh, K. Ren, and J. Beuchat, "Guest editorial: Introduction to the special issue on emerging security trends for biomedical computations, devices, and infrastructures," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2016.

[47] 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.

[48] B. Koziel, R. Azarderakhsh, and M. Mozaffari-Kermani, "Post-quantum cryptography on FPGA based on isogenies on elliptic curves," *IEEE Trans. Circuits Syst. I, Reg. Papers*, to appear in 2016.

[49] M. Mozaffari-Kermani and R. Azarderakhsh and V. Singh, "Reliable low-latency Viterbi algorithm architectures benchmarked on ASIC and FPGA," *IEEE Trans. Circuits Syst. I, Reg. Papers*, to appear in 2016.

[50] P. Chen, S. N. Basha, M. Mozaffari-Kermani, R. Azarderakhsh, and J. Xie, "FPGA realization of low register systolic all-one-polynomial multipliers over $GF(2^m)$ and their applications in trinomial multipliers," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, to appear in 2016.

[51] M. Mozaffari-Kermani, S. Sur-kolay, A. Raghunathan, and N. K. Jha, "Systematic poisoning attacks on and defenses for machine learning in healthcare," *IEEE J. Biomedical and Health Informatics*, vol. 19, no. 6, pp. 1893 – 1905, Nov. 2015.

[52] R. Azarderakhsh and M. Mozaffari-Kermani, "High-performance two-dimensional finite field multiplication and exponentiation for cryptographic applications," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 10, pp. 1569 – 1576, Oct. 2015.

[53] M. Mozaffari-Kermani, R. Azarderakhsh, S. Bayat-Sarmadi, and C. Lee, "Systolic gaussian normal basis multiplier architectures suitable for high-performance applications," vol. 23, no. 9, pp. 1969–1972, Sept. 2015.

[54] A. Mohsen-nia, M. Mozaffari-Kermani, S. Sur-kolay, A. Raghunathan, and N. K. Jha, "Energy-efficient long-term continuous personal health monitoring," *IEEE Trans. Multi-Scale Computing Systems,*, vol. 1, no. 2, April 2015.

[55] M. Mozaffari-Kermani, N. Manoharan, and R. Azarderakhsh, "Reliable Radix-4 complex division for fault-sensitive applications," *IEEE Trans. Comput. Aided Design Integr. Circuits Syst.*, vol. 34, no. 4, pp. 656 – 667, April 2015.

[56] R. Azarderakhsh, M. Mozaffari-Kermani, and K. U. Jarvinen, "Secure and efficient architectures for single exponentiation in finite field suitable for high-performance cryptographic applications," *IEEE Trans. Comput. Aided Design Integr. Circuits Syst.*, vol. 34, no. 3, pp. 332 – 340, Mar. 2015.

[57] M. Mozaffari-Kermani, S. Bayat-Sarmadi, and R. Azarderakhsh, "Fault-resilient lightweight cryptographic block ciphers for secure embedded systems," *IEEE Embedded Sys.*, vol. 6, no. 4, pp. 89 – 92, Dec. 2014.

[58] S. Bayat-Sarmadi, M. Mozaffari-Kermani, and A. Reyhani-Masoleh, "Efficient and concurrent reliable realization of the secure cryptographic SHA-3 algorithm," *IEEE Trans. Comput. Aided Design Integr. Circuits Syst.*, vol. 33, no. 7, pp. 1105 – 1109, Jul. 2014.

[59] J. Pan, R. Azarderakhsh, M. Mozaffari-Kermani, C. Lee, C. Chiou, and J. Lin, "Low latency digit-serial systolic double basis multiplier over $GF(2^m)$ using subquadratic

toeplitz matrix-vector product approach," *IEEE Trans. Comput.*, vol. 63, no. 5, pp. 1169 – 1181, May 2014.

[60] K. U. Jarvinen, R. Azarderakhsh, and M. Mozaffari-Kermani, "Efficient algorithm and architecture for elliptic curve cryptography for extremely constrained secure applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 4, pp. 1144 – 1155, Apr. 2014.

[61] S. Bayat-Sarmadi, M. Mozaffari-Kermani, R. Azarderakhsh, and C. Lee, "Dual basis super-serial multipliers for secure applications and lightweight cryptographic architectures," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 61, no. 2, pp. 125 – 129, Feb. 2014.

[62] 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.

[63] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "Efficient and high-performance parallel hardware architectures for the AES-GCM," *IEEE Trans. Comput.*, vol. 61, no. 8, pp. 1165 – 1178, Aug. 2013.

[64] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "A low-power high-performance concurrent fault detection approach for the composite field S-box and inverse S-box," *IEEE Trans. Comput.*, vol. 60, no. 9, pp. 1327 – 1340, Sept. 2011.

[65] 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.

[66] 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.

[67] M. Mozaffari Kermani and A. Reyhani Masoleh, "Fault detection structures of the S-boxes and the inverse S-boxes for the Advanced Encryption Standard," *J. Electronic Testing: Theory and Applications (JETTA)*, vol. 25, no. 4, pp. 225 – 245, Aug. 2009.

[68] M. Mozaffari-Kermani, R. Azarderakhsh, and J. Xie, "Error detection reliable architectures of Camellia block cipher applicable to different variants of its substitution boxes," *Proc. IEEE Asian Hardware Oriented Security and Trust Symposium (HOST)*, to appear in 2016.

[69] B. Koziel, R. Azarderakhsh, A. Jalali, D. Jao, and M. Mozaffari-Kermani, "NEON-SIDH: Efficient implementation of supersingular isogeny diffie-hellman key exchange protocol on ARM," *Proc. Conf. Cryptology and Network Security (CANS)*, to appear in 2016.

[70] B. Koziel, R. Azarderakhsh, and M. Mozaffari-Kermani, "Fast hardware architectures for supersingular isogeny diffie-hellman key exchange on FPGA," *Proc. Int. Conf. (INDOCRYPT)*, to appear in 2016.

[71] M. Mozaffari-Kermani and R. Azarderakhsh, "Lightweight hardware architectures for fault diagnosis schemes of efficiently-maskable cryptographic substitution boxes," *Proc. IEEE Int. Conf. ICECS*, to appear in 2016.

[72] A. Aghaie, M. Mozaffari-Kermani, and R. Azarderakhsh, "Fault diagnosis schemes for secure lightweight cryptographic block cipher RECTANGLE benchmarked on FPGA," *Proc. IEEE Int. Conf. ICECS*, to appear in 2016.

[73] M. Mozaffari-Kermani, R. Azarderakhsh, and M. Mirakhorli, "Multidisciplinary approaches and challenges in integrating emerging medical devices security research and education," *Proc. Conf. American Society for Engineering Education*, to appear in 2016.

[74] R. Ramadoss, M. Mozaffari-Kermani, and R. Azarderakhsh, "Efficient error detection architectures for CORDIC through recomputing with encoded operands," *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, pp. 2154 – 2157, May 2016.

[75] B. Koziel, M. Mozaffari-Kermani, R. Azarderakhsh, and D. Jao, "Post-quantum cryptography on FPGA based on isogenies on elliptic curves," *Proc. eprint 2016/672*, pp. 1 – 18, 2016.

[76] B. Koziel, A. Jalali, M. Mozaffari-Kermani, R. Azarderakhsh, and D. Jao, "NEON-SIDH: Efficient implementation of supersingular isogeny diffie-hellman key-exchange protocol on ARM," *Proc. eprint 2016/669*, pp. 1 – 16, 2016.

[77] B. Koziel, M. Mozaffari-Kermani, and R. Azarderakhsh, "Low-resource and fast binary edwards curves cryptography using gaussian normal basis," *Proc. Int. Conf. (INDOCRYPT)*, pp. 347 – 369, 2015.

[78] M. Mozaffari-Kermani and R. Azarderakhsh, "Reliable hash trees for post-quantum stateless cryptographic hash-based signatures," *Proc. IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems (DFT)*, pp. 103 – 108, Oct. 2015.

[79] M. Mozaffari-Kermani and R. Azarderakhsh, "Integrating emerging cryptographic engineering research and security education," *Proc. Conf. American Society for Engineering Education*, pp. 1 – 13, Jun. 2015.

[80] C. E. Kennedy and M. Mozaffari-Kermani, "Generalized parallel crc computation on FPGA," *Proc. IEEE Conf. Elec. Comput. Eng.*, pp. 107 – 113, May. 2015.

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

[82] M. Zhang, M. Mozaffari-Kermani, A. Raghunathan, and N. K. Jha, "Energy-efficient and secure sensor data transmission using encompression," *Proc. IEEE Int. Conf. VLSI Design*, pp. 31 – 36, Jan. 2013.

[83] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "Reliable hardware architectures for the third-round SHA-3 finalist grostl benchmarked on FPGA platform," *Proc. IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems (DFT)*, pp. 325 – 331, Oct. 2011.

[84] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "A high-performance fault diagnosis approach for the AES SubBytes utilizing mixed bases," *Proc. IEEE Workshop Fault Diagnosis and Tolerance in Cryptography (FDTC)*, pp. 80 – 87, Sep. 2011.

[85] M. Mozaffari Kermani and A. Reyhani Masoleh, "A low-cost S-box for the Advanced Encryption Standard using normal basis," *Proc. IEEE Int. Conf. Electro/Information Technology (EIT)*, pp. 52 – 55, Jun. 2009.

[86] M. Mozaffari-Kermani and A. Reyhani-Masoleh , "A lightweight concurrent fault detection scheme for the AES S-Boxes using normal basis," *Proc. LNCS Cryptographic Hardware and Embedded Systems (CHES)*, pp. 113 – 129, Aug. 2008.

[87] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "A structure-independent approach for fault detection hardware implementations of the Advanced Encryption Standard," *Proc. IEEE Workshop Fault Diagnosis and Tolerance in Cryptography (FDTC)*, pp. 47 – 53, Sep. 2007.

[88] M. Mozaffari Kermani and A. Reyhani Masoleh, "Parity-based fault detection architecture of S-box for Advanced Encryption Standard," *Proc. IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems (DFT)*, pp. 572 – 580, Oct. 2006.

[89] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "Parity prediction of S-box for AES," *Proc. IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 2357 – 2360, May 2006.

[90] B. Koziel, R. Azarderakhsh, D. Jao, and M. Mozaffari-Kermani, "On fast calculation of addition chains for isogeny-based cryptography," *Proc. Inscrypt*, 2016.

[91] B. Koziel, R. Azarderakhsh, D. Jao and M. Mozaffari-Kermani, "On fast calculation of addition chains for isogeny-based cryptography," *Proc. eprint 2016/1045*, pp. 1–20, 2016.

[92] B. Koziel, R. Azarderakhsh, and M. Mozaffari-Kermani, "Fast hardware architectures for supersingular isogeny diffie-hellman key exchange on FPGA," *Progress in*

Cryptology–INDOCRYPT 2016: 17th International Conference on Cryptology in India, Kolkata, India, December 11-14, 2016, Proceedings, pp. 191–206, 2016.

[93] M. Mozaffari Kermani, "Fault detection schemes for high performance vlsi implementations of the Advanced Encryption Standard," Master's thesis, The University of Western Ontario London, April. 2007.

[94] M. Mozaffari-Kermani, "Reliable and high-performance hardware architectures for the Advanced Encryption Standard/Galois Counter Mode," Ph.D. dissertation, The University of Western Ontario, July. 2011.

[95] M. Mozaffari Kermani, R. Azarderakhsh, and M. Mirakhorli, "Education and research integration of emerging multidisciplinary medical devices security," *American Society for Engineering Education (ASEE)*, 2016.