7-20-2017

Efficient Error detection Architectures for for Low-Energy Block Ciphers with the Case Study of Midori Benchmarked on FPGA

Anita Aghaie
aa6964@rit.edu

Graduate Project as Part of Master's of Science Degree

Efficient Error detection Architectures for for Low-Energy Block Ciphers with the Case Study of Midori Benchmarked on FPGA

Anita Aghaie

July 2017

Rochester Institute of Technology
Department of Computer Engineering
College of Engineering

Department Chair
Prof. Shanchieh Jay Yang

Supervisor
Prof. Mehran Mozaffari-Kermani

Date of the graduation
20.July.2017

For my kind parents, Azam and Mohammad;
My awesome siblings, Mandana and Amin;
And my best love, Mahdi.

Acknowledgments

It seems so obvious pursuing cryptography as an academic degree requires a wide range of various areas such as engineering, mathematics, and computer science. Combining of all these interesting research areas motivates me to continue in cryptography and I have had the opportunity to publish a number of IEEE Transactions journal papers and a conference paper to wrap up what I have learned till now.

After an intensive period of two years and one year of remote research, for the total of three year, writing this note of thanks is the finishing touch on my dissertation and my Master's program. It has been a period of intense learning for me, not only in the scientific area, but also on a personal level. I would like to reflect on the people who have supported and helped me so much throughout this period.

I would first like to thank my supervisor Dr. Mehran Mozaffari-Kermani for all of the opportunities I was given to conduct my research and further my dissertation. Without his support and encouragement, this research and my papers would not be successful. Also, I would like to thank my co-adviser Dr. Reza Azarderakhsh for his guidance through my research. I thank Computer Engineering department and Dr. Dhireesha Kudithipudi for their valuable support during my Master's program. Moreover, this thesis was performed under the U.S. federal agency award 60NANB16D245 granted from U.S. Department of Commerce, National Institute of Standards and Technology (NIST).

I would also like to thank my lovely husband, Mahdi Homayouni and my wonderful parents for their wise counsel and sympathetic ear. You are always there for me. Finally, there are my friends especially Mahshad Mahdavi Hezaveh. We were not only able to support each other by deliberating over our problems and findings, but also happily by talking about things other than just our papers.

Abstract

Achieving secure, high performance implementations for constrained applications such as implantable and wearable medical devices is a priority in efficient block ciphers. However, security of these algorithms is not guaranteed in presence of malicious and natural faults. Recently, a new lightweight block cipher, Midori, has been proposed which optimizes the energy consumption besides having low latency and hardware complexity. This algorithm is proposed in two energy-efficient varients, i.e., Midori64 and Midori128, with block sizes equal to 64 and 128 bits. In this thesis, fault diagnosis schemes for variants of Midori are proposed. To the best of the our knowledge, there has been no fault diagnosis scheme presented in the literature for Midori to date. The fault diagnosis schemes are provided for the nonlinear S-box layer and for the round structures with both 64-bit and 128-bit Midori symmetric key ciphers. The proposed schemes are benchmarked on field-programmable gate array (FPGA) and their error coverage is assessed with fault-injection simulations. These proposed error detection architectures make the implementations of this new low-energy lightweight block cipher more reliable.

List of Figures

List of Tables

Contents

1 Introduction

1.1 Overview of Cryptography

Regardless of the situation and applications, for many years, information security has played a generally crucial role in transmission and different forms of conveyance of information. Cryptography uses secure methods, e.g., utilizing block ciphers (for encryption) and hash function (for integrity checking), to boost information security against the presence of third parties through data transmission. Updated and various cryptographic mechanisms such as the recent post-quantum algorithms can be applied to increase the security level of embedded systems. Several requirements for information security should be considered in the utilized protocols and mechanisms of communication systems, such as data integrity, confidentiality, non-repudiation, validation, timestamping, and authentication. Conceptually, the goals of cryptography, including data integrity, confidentiality, and authentication, have not changed dramatically over time and researchers have spent many years ameliorating them using different and efficient approaches [1]-[2]. However, designing new algorithms and applying various techniques do not completely guarantee the information security in the present electronic security.

The data integrity goal provides the unauthorized alteration of data which be possi-

ble by detecting data manipulation such as insertion and substitution. Confidentiality gives the service of encryption and decryption to make a secure information as well as authentication makes the identification service for the content of information. In addition, non-repudiation is also important due to preventing components from denying previous commitments or actions through the data transmission progress [1].

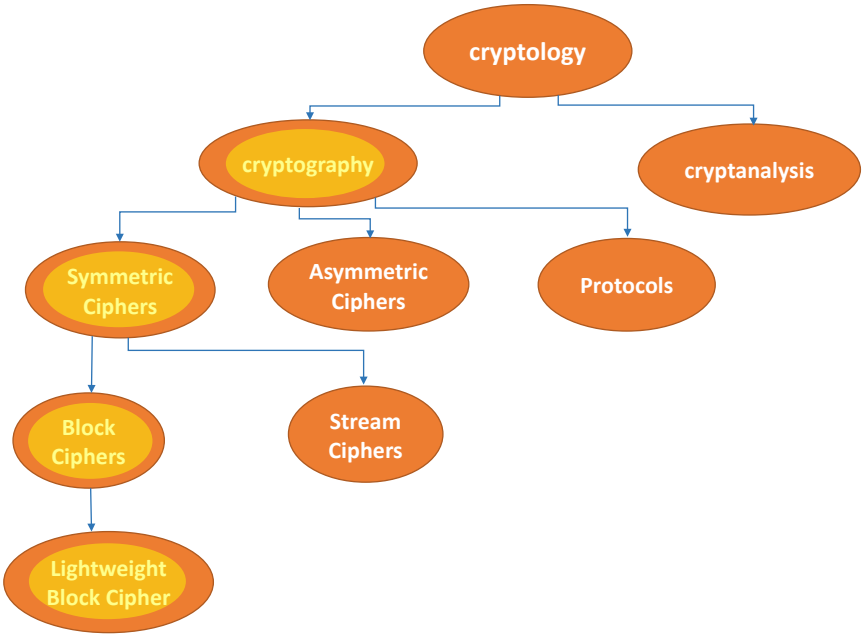

Figure 1.1: The classification of cryptology.

To meet all of the aforementioned necessary goals, cryptography science is divided into various parts: Fig. 1.1 depicts the main cryptographic primitives. As shown in Fig. 1.1, cryptology more generally consists of cryptography for designers who make security for safe data transmitting, and the latter part, cryptanalysis, gives a chance to ill-intentioned people to break the security provided by cryptographers. However, we will focus on cryptography, which splits into three branches: symmetric ciphers, asymmetric ciphers, and protocols. Symmetric algorithms such as the Data Encryption Standard (DES) and Advanced Encryption Standard (AES) use the

inverse operation with the same key for encryption and decryption of the data to perform a secure communication. This key must be transmitted by a secure channel among senders and receivers. On the other hand, in the asymmetric ciphers or public key algorithms, there are two keys: one is a symmetric ciphers key (secret key), and the other is used just for authentication of the sender or receiver (public key). As indicated by their name, public keys are published in the public sphere, such as on the internet, while the private key is unique for each side of communication and it is infeasible to obtain the private key from the encrypt or decrypt function or the public key. The third branch of cryptography is protocols that apply various algorithms of the symmetric and asymmetric ciphers to develop applications such as security of internet communication or Transport Layer Security (TLS) schemes in web browsers [2]-[3].

This thesis focuses on symmetric cryptography in the form of lightweight block ciphers. In detail, a novel lightweight block cipher, Midori, uses the same key for encryption and decryption as a symmetric cipher with the block of inputs/outputs, and keys in the form of a block cipher. This block cipher counts as a lightweight one, discussed in the following section 1.2.

1.2 Lightweight Cryptography

Lightweight cryptography has had an essential role in achieving high security with low area and low energy consumption in many sensitive applications, such as secure embedded systems, wireless nano-sensors, radio-frequency identification (RFID) tags, and implantable and wearable medical devices. Such efficiency is more critical in energy-constrained applications such as implantable medical devices, in which re-

placing discharged batteries with power-inefficient architectures is a burden due to the required surgeries to do so [4]. In addition, tiny computing devices such as RFID tags and sensors need efficient block ciphers because of their small area and limited source power [5].

In these sensitive devices, low energy consumption is among the most significant concerns – especially with the advent of the Internet of Things, in which, through multi-channel connectivity, data are exchanged in embedded systems [6]. This challenge is answered by lightweight block ciphers, which provide a high level of security, low energy consumption, and low hardware complexity, such as PRESENT [7], CLEFIA [8], PRINCE [9], SIMON and SPECK [10], and LED [11]. The presented lightweight block ciphers are solutions to reach reasonable confidentiality with low area in different networks, e.g., Mobile Ad hoc NETworks (MANETs) that have no physical layer security [12]. It is noted that AES is the current symmetric key cryptography standard in encryption and decryption operations which has been optimized in terms of area and power consumption [13]. Nevertheless, Nevertheless, prominent efforts have been made to considerably reduce area and power through AES, e.g., 2400 gate equivalent for 128-bit AES [6]; the optimized AES has still large overheads in terms of area and power consumption for highly-constrained environments.

Standardization of lightweight block ciphers has been a concern of designers and consumers, and there are no specific criteria for lightweight designers. However, several standards such as ISO 29192-2 specify two lightweight block ciphers, CLEFIA and PRESENT [7]-[8]. The first one is a Feistel-cipher with 128-bit structure and can provide a high security level along with good hardware and software implementation capabilities, such as high-speed performance on a wide range of processors. On the other hand, PRESENT has a substitution-permutation network (SPN) structure and has more optimized performance than AES. This lightweight cipher applies 4-

bit S-boxes as a compact design to create low power consumption and high efficiency.

Efficiency of lightweight block ciphers is dependent on many metrics, such as area, latency, power, and energy; indeed, power and energy consumption are correlated metrics, but energy is a more relevant metric to determine a productive design than the latter is [14]-[17]. Having high confidentiality without adding overhead in terms of energy consumption and hardware complexity is the reason to design new and more efficient lightweight symmetric key block ciphers like Midori.

From the view of energy consumption, the Midori variants have a round base architecture and use less energy than 1.89 pJ/bit encrypted, which is far better than other lightweight block ciphers such as PRINCE and NOEKEON [9, 14, 18]. In other words, Midori provides an acceptable security level with more optimal energy consumption in comparison to previous lightweight symmetric ciphers . The S-boxes of Midori are different from those of the AES and other lightweight block ciphers. Furthermore, Midori has two types of bijective 4-bit S-boxes, which are more energy-efficient than 8-bit ones. It is noted that Midori, like other lightweight block ciphers, accepts optimal cell-permutation matrices and uses the most efficient maximum distance separable (MDS) matrices due to low implementation overheads and increasing immunity against several attacks [14].

1.3 Fault Diagnosis

It has been shown in the state-of-the-art that a large number of side-channels through data transmission provide information that reveals important and compromising details about secret data. Some of these details can be used as new trapdoors to invert a trapdoor one-way function without the secret key. This allows an adversary to break a cryptographic protocol, even if it proved to be secure in the classical,

mathematical sense. The various side-channels include timing measurements, power consumption, sound, electromagnetic emissions, and faults [19].

A fault induced into a system causes a defect while running a program in which the adversary is able to observe the reaction and break the system. Unlike other side channel attacks, which are passive, fault attacks are active ones, where an adversary must tamper with an attacked device to create faults. Depending on the type of physical stress applied, faults may show effects of different duration. As a result, we differentiate between transient faults, permanent faults, and destructive faults.

A destructive fault occurs if an adversary destroys a physical structure on the chip, which causes a certain bit or variable to be fixed at a specific value for all successive runs of the device. Destructive faults cannot be reversed. Permanent and transient faults, called solid (hard) faults and soft faults, respectively, are both faults that do not modify the hardware of an attacked device, thus allowing it to recover from the induced faults after a certain period of time [19]. Permanent faults change an affected variable until that variable is explicitly overwritten, e.g., by a reset at the start of the next run, whereas temporary faults are only short-lived, such that after a given amount of time, the effect ceases to exist and the correct value is present again.

These soft faults are classified as intermittent when they develop into a permanent fault, or transient when they are caused by some external disturbance such as power supply fluctuations. In addition, they are categorized as parametric or logical in terms of fault effect. The first type of fault makes changes through speed, voltage, or current as it alters the circuit parameter magnitude, whereas logical faults make changes in the Boolean function through the circuit. One kind of significant faults through embedded systems is known as a delay fault, which is caused by slow gates and is an effective parametric fault through embedded systems. This kind of fault

can cause problems of critical races. These faults can be either local or distributed. A distributed fault, such as a clock malfunction, affects multiple variables, whereas a local fault affects a single variable. Consequently, increasing the number of components on a single chip because of VLSI developments has led to an increase in fault occurrence probability through different embedded systems. Thus, fault attacks and different types and countermeasures against them are the focus of many studies, as is described in the following.

1.3.1 Type of Faults

As mentioned above, logical faults are a significant type of fault that represent the behavior of the system modeled. Logical faults consist of three important classes.

A) Stuck-at-faults: A single stuck-at-fault happens when either one of the inputs or the outputs of the logic gate is fixed at either a logic 1 (stuck-at-1) or a logic 0 (stuck-at-0). They can be denoted by the abbreviations s-a-1 and s-a-0, respectively. This fault model is a good representation of types of defects, such as open circuits and short circuits. In practice, stuck-at faults are usually considered to be destructive faults, where it is assumed that a destroyed wire, gate, or memory cell will cause the faulty bit to be stuck at the value 0. Moreover, the stuck-at model can represent multiple faults that result when multiple signal lines are stuck at logic 0 or logic 1.

B) Bridging faults: Bridging faults occur when two or more signal lines are accidentally connected together, depending on the logic circuitry employed. They can be classified as follows.

i) *Input Bridging*: This bridging fault happens when a definite number of primary input lines are shorted. It can form wired logic or a voting model.

ii) *Feedback Bridging:* This type of fault can introduce a feedback with existing

a short circuit between an input and an output line. The fault causes the circuit to either oscillate or latch (additional memory). It is noted that this may happen between two or more signal lines or between the terminals of the transistor. In CMOS circuits, depending on the bridging resistance and the physical location, faults can manifest as either stuck-open or stuck-at faults.

iii) *Non-feedback Bridging:* Apart from the above two, all other remaining types of existing bridging faults are in this non-feedback fault group. If two lines happen to be physically close to each other, the probability of them being bridged is higher. In a positive logic, a bridging fault is assumed to behave as wired-AND with the dominant logic value being 0. In a negative logic, a bridging fault is assumed to behave as wired-OR with the dominant value being 1.

C) Delay Faults: Because of statistical variations in manufacturing processes, there is an increased probability of the appearance of smaller defects that cause a circuit to be partially short or open circuit. Due to these defects, the circuit fails to meet the timing specifications without altering its logic function. The transition of the signal might be delayed from 1 to 0 or vice versa due to a small defect. This is called a delay fault. They are of two types of delay faults.

i) *Gate Delay Fault:* This fault helps in modeling defects that cause the propagation delay of the faulty gate to exceed the worst-case value specified. It can be used to model isolated defects but not distributed defects.

ii) *Path Delay Fault:* This fault can be used to model both isolated and distributed defects. It occurs when the propagation delay exceeds its specified limit along a circuit path.

D) Transition and Intermittent Faults: These faults can be classified as temporary faults. The majority of the malfunctioning in digital circuits results from temporary faults, and these are also difficult to detect and isolate. Transient faults

are the non-recurring temporary faults that occur due to the fluctuations of the power supply or the circuit exposure to some external radiation, such as α-particle radiation. As there is no physical damage to the hardware, these faults cannot be repaired and thus are a major source of failures. Intermittent faults are due to poor designs, loose connections, or components that are partially defective. They happen because of the deteriorating or aging of the components, and because of external environmental conditions such as vibration, humidity, temperature, etc. Intermittent faults are based on the protection of the system from the physical environment through cooling, filtering, shielding, etc.

1.3.2 Fault Degradation

In digital systems, errors can happen through various causes including alpha particles from package decay, cosmic rays creating energetic neutrons and protons, and thermal neutrons. In advanced process technologies, errors can occur due to device shrinking, reduced power supply voltages, and higher operating frequencies which increase the probability of transient errors which can significantly affect reliability of computations. In addition, single event upsets and single event transients are generated due to cosmic rays which create energetic protons and neutrons, thermal neutrons, random noise, or signal integrity problems all resulting in device errors.

Degradation in digital circuits can happen in many ways such as:

- *Time-Dependent Dielectric Breakdown* causes the leakage current affecting the transistor gates to increase, it results in short circuit.
- The phenomenon of *Electromigration* causes the metal ions to migrate thus leading to voids and holes in interconnect. These can cause open or short

circuits which can cause faults.

- The *Hot-carrier effect (HCE)* can cause the threshold voltage in CMOS transistors to increase and also results in the degradation of electron mobility.

1.3.3 Fault Detection Techniques

The process of determining whether the circuit contains a fault or not is called fault detection [20]. As it is important to counteract such natural faults in order to achieve fault immunity and reliability, error detection has been an important part of a number of hardware architectures in different domains, including various arithmetic unit sub-components [21]. In previous work, reliable architectures have been devised to counteract natural or malicious faults, e.g., cryptographic architectures immune to faults through concurrent error detection [22]. Different fault detection strategies can be classified as follows:

A) Concurrent Error Detection: This mechanism helps in detecting the faults in the circuit concurrently with the normal operation of the circuit by making use of additional logic. It results in an error if the resulting output is found different than the predicted output by the checker unit [23]. The error coverage can be improved greatly using the methods of duplication or including parity check registers in the circuit. For improving the error coverage, the trade-off with area or latency, or throughput can be made. The errors can be also detected by running the circuit twice, once with the original operands and the second time using encoded operands such that different outputs are obtained. The checker will raise the error indication flag in case of a mismatch between the two outputs. The operands can be encoded using different methods [24].

B) Off-Line Fault Detection: This method helps in identifying faults in FPGAs and ASICs when they are not in operation with the use of additional circuitry. It helps in detecting manufacturing defects. Automated-Test-Pattern-Generator (ATPG) and Built-in-Self-Test (BIST) are some examples of off-line test circuits. The fault detection process does not involve the original circuitry. It connects the device under test between a pattern generator and an output response analyzer. In order to obtain full error coverage, it is important to check the logic and interconnects and the configuration network. For the FPGAs [25], the need of a large number of test configurations has been eliminated as the additional testing circuitry is built into the development boards by most of the recent consumer grade FPGAs [26]. BIST does not interfere with the normal FPGA operation, and also covers clock networks and PLLs which are complicated systems.

C) Roving Fault Detection: This method helps in pointing out the faulty location in the FPGA circuit. It checks for defects in the FPGA by scanning it entirely and replaces those defects with a test function. It basically helps in adapting the BIST techniques with minimum increase in the area. In the roving detection, the entire FPGA is split equally into a number of regions where one region carries out the BIST testing while the others undergo normal operations. The speed of the roving method depends on the speed of the roving cycle as well as on the operation time. It has been reported that the latency of the best roving methods is less than one second.

In this thesis, we focus on error detection in crypto-architectures that has been the center of attention in many previous works. Concurrent error detection techniques have been widely used to architect reliable hardware for the AES and other cryptographic algorithms [27]-[34]. It is well-known that concurrent error detection techniques include a number of schemes, i.e., hardware, information,

time, hybrid redundancy. Hardware redundancy makes use of extra hardware to process the same input twice to match the two outputs; any mismatch will trigger the error flag. Information redundancy schemes have a number of variants, e.g., parity codes [35] and robust codes [36]. Time redundancy technique has a number of schemes, i.e., recomputing with shifted operands (RESO) [37]-[38], recomputing with rotated operands (RERO) [39], and recomputing with permuted operands (REPO) [40]. The hybrid redundancy scheme is given in [41]-[43] where different improve-ments in the architecture have been proposed. The choice of the error detection technique is completely dependent on the requirements in terms of overhead toler-ance, security, and reliability. In the case of Midori, to the best of the our knowledge, there is no prior work. The merit of the proposed approaches in this thesis compared to the approaches presented before for lightweight block ciphers is two-fold. First, we present both logic-gate based and look-up table based error detection schemes for the two types of the S-boxes in Midori which gives freedom to the designers to choose the implementation strategy based on the implementation and performance metrics requirements and the platform to implement. Second, for the MixColumn operation, we have examined to achieve have low-overhead detection approaches, by performing design space explorations before-math not as an afterthought. Such careful investigations to have a combined original implementation and error detection architecture has not been performed in previous state-of-the-art.

1.4 Objectives

Through this thesis and for the new lightweight block cipher Midori, we propose error detection schemes with the reasonable area and power consumption overheads for the highly-constrained applications. To the best of our knowledge, research on

developing reliable architectures for Midori have not been reported to date.

The main contributions of this thesis are summarized as follows:

- We propose signature-based schemes for the linear and nonlinear blocks of Midori.
  - For the S-boxes within Midori, we derive and implement both look-up table (LUT) and logic gate S-boxes and we propose fault diagnosis schemes that can be tailored based on the reliability and overheard objectives. This gives the designers freedom in designing the S-boxes based on available resources and performance to achieve.
  - For the non-linear blocks, we present fault diagnosis schemes which are tailored towards design-for-low-overhead fault detection. The MixColumn operation has been examined to achieve a number of schemes and the selection of the matrices within has been carefully done to have low-overhead detection approaches.
- The performed simulation results show high error coverage for the presented schemes. Using the proposed approaches, the error detection structures are capable of detecting the injected faults with high coverage (transient and permanent as well as single, multiple, and adjacent faults).
- Through FPGA implementations using Xilinx Virtex-7 family, it is shown that the overheads of the proposed architectures are acceptable for resource-constrained applications.

1.5 Thesis Outline

In this thesis, we present the error detection constructions of Midori through the following chapters:

- Chapter 2: This chapter explains the brief preliminary of the Midori variants architectures.
- Chapter 3: Through this chapter, our proposed fault detection schemes for Midori are presented.
- Chapter 4: This chapter provides the results of fault injected simulations which determine the capability of our proposed schemes during different faults. Moreover, some of these schemes are implemented on FPGA.
- Chapter 5: This chapter finalizes the thesis by discussing our future scope in this area and summarizing the mentioned research work.

2 Preliminaries

2.1 Midori

Midori consists of two parts, i.e., data processing and key scheduling modules. The plaintext input and the ciphertext output, which are 64 bits or 128 bits in width, are divided to 4-bit and 8-bit cells, respectively. This is also performed for the whitening key (WK) and round keys (RK_i). The round keys are used as input to the main functions of the algorithm, and the whitening keys are modulo-2 added with input and output of the entire encryption or decryption operation. Two variants of Midori, Midori64 and Midori128, are 64-bit block cipher and 128-bit block cipher with the same key length of 128 bits corresponding to 16 and 20 number of rounds, respectively.

The Midori that is based on the SPN structure as shown in Fig. 2.1, uses the following 4×4 array state [14]:

$$S = \begin{pmatrix} s_0 & s_4 & s_8 & s_{12} \\ s_1 & s_5 & s_9 & s_{13} \\ s_2 & s_6 & s_{10} & s_{14} \\ s_3 & s_7 & s_{11} & s_{15} \end{pmatrix}, \tag{2.1}$$

in which the size of each cell is 4 bits and 8 bits for Midori64 and Midori128,

respectively. Midori applies bijective S-boxes, Sb_0 and Sb_1, with 4-bit structure and involution property which are used in Midori64 and Midori128, respectively. Midori128 utilizes four different 8-bit S-boxes, SSb_0, SSb_1, SSb_2, SSb_3. Each of these 8-bit S-boxes consists of two 4-bit $Sb_1$with permutation input and output structures which are described as shown in Fig. 2.1 and in more details in [14].

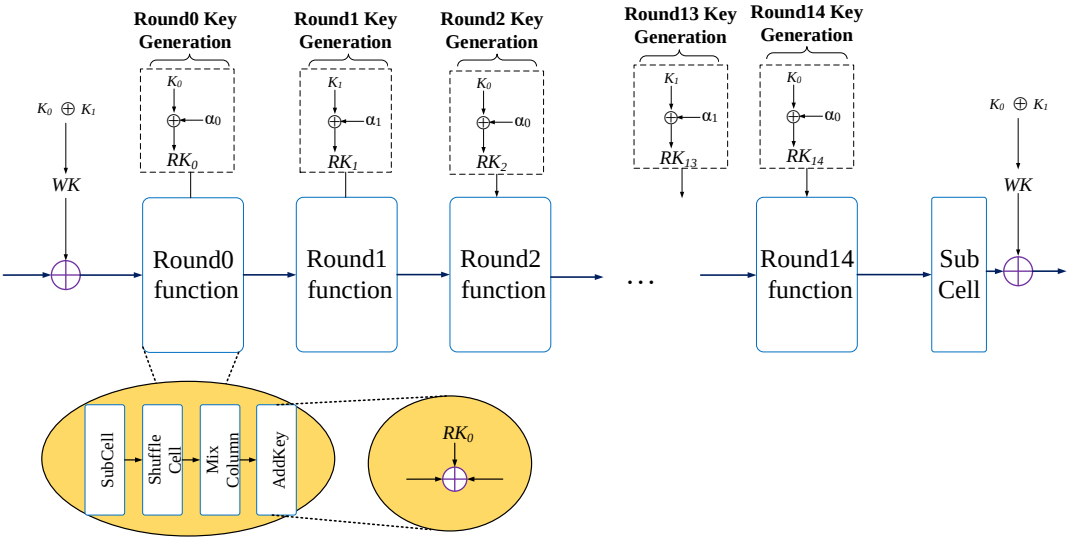

Figure 2.1: Midori lightweight block cipher diagram.

Moreover, Table 3.1 (in the next chapter) represents the structure of bijective S-boxes Sb_0 and Sb_1 that constitute the S-layers to perform round functions. The S-boxes are utilized in each round functions and apply the following four operations to the state matrix:

- SubCell (S): The 4-bit and 8-bit S-boxes are used for each element of the state S in Midori64 and Midori128 in parallel. We have $s_i \leftarrow Sb_0[s_i]$ for Midori64 and $s_i \leftarrow SSb_{(imod4)}[s_i]$ for Midori128, where $0 \leq i \leq 15$.

- ShuffleCell (S): Each byte of the state is derived as follows:

$(s_0, s_1, ..., s_{15}) \leftarrow (s_0, s_{10}, s_5, s_{15}, s_{14}, s_4, s_{11}, s_1, s_9, s_3, s_{12}, s_6, s_7, s_{13}, s_2, s_8)$.

- MixColumn (S): Midori utilizes an involutive binary matrix M, applied to every $4m$-bit column of the state S, i.e., $(s_i,s_{i+1},s_{i+2},\ s_{i+3})^T \leftarrow M\times(s_i,\ s_{i+1},\ s_{i+2},\ s_{i+3})^T$ and $i = 0, 4, 8, 12$.
- KeyAdd (S, RK_i): The i-th n-bit round key RK_i is modulo-2 added to the state S.

Before the first round, an additional KeyAdd operation is applied, and in the last round, the ShuffleCell and MixColumn operations are omitted. The data processing part of the Midori consists of its encryption and decryption that perform the mentioned round function for specific number of rounds except the last round. The decryption is performed through the same sequence of mentioned round function with a difference of the added InvShuffleCell.

3 Proposed Reliable Architectures for Midori

In this chapter, the error detection approaches of subblocks in the Midori encryption and decryption are proposed.

3.1 Proposed Approaches for the S-boxes Variants

In the hardware implementations of Midori, two approaches can be used for realizing the S-boxes, i.e., LUT-based and logic gate-based implementations. The LUT-based S-boxes have advantages such as good performance and disadvantages such as having high area and power consumption. On the other hand, the latter approach typically has less area and power consumption [14].

Our proposed signature-based error detection approach is not confined to a special signature. However, for the sake of clarity, we present two examples, i.e., parity-based and interleaved parity-based approaches.

We can store predicted parities (or interleaved parities) of elements from the S array in LUTs. The scheme for the S-box Sb_0 and Sb_1 is based on deriving the predicted parities of the S-boxes using LUTs as shown in Table 3.1. For each element of

S-boxes, we modulo-2 add all bits. Then, we store the result as parity bit in an extended LUT with five-bit elements (note that one extra bit is added to each 4-bit entry). Thus, the new, protected state would consist of 16 five-bit elements which can be stored in FPGA block memories or pipelined distributed LUTs. An example would be to derive the parity of the first element of Sb_0 which is $\{c\}_{16} = \{1100\}_2$ which is zero.

The other signature-based error detection scheme is based on interleaved parity bits that is proposed in order to protect the non-linear S-boxes. Interleaved parity-based schemes are able to detect burst faults, i.e., adjacent multiple faults. Such faults happen in both natural defects and malicious fault attacks. In this scheme, we compute the interleaved parity bits for 4-bit bijective S-boxes Sb_0 and Sb_1 in hexadecimal form as shown in Table 3.1. We have derived such parities by modulo-2 addition of odd bits and even bits with each other separately. Similarly, these 2-bit interleaved parities along with 4-bit elements of each state are stored as 6-bit elements in memories. An example would be to derive the interleaved parity of the first element of Sb_0 which is $\{c\}_{16} = \{1100\}_2$ which is 11 (modulo-2 adding the odd and even bits separately).

Table 3.1: Interleaved parity and parity of 4-bit bijective S-boxes Sb_0 and Sb_1 in hexadecimal form

x	0	1	2	3	4	5	6	7
Sb_0	C (11,0)	A (00,0)	D (10,1)	3 (11,0)	E (01,1)	B (01,1)	F (00,0)	7 (10,1)
Sb_1	1 (01,1)	0 (00,0)	5 (00,0)	3 (11,0)	E (01,1)	2 (10,1)	F (00,0)	7 (10,1)
x	8	9	A	B	C	D	E	F
Sb_0	8 (10,1)	9 (11,0)	1 (01,1)	5 (00,0)	0 (00,0)	2 (10,1)	4 (01,1)	6 (11,0)
Sb_1	D (10,1)	A (00,0)	9 (11,0)	B (01,1)	C (11,0)	8 (10,1)	4 (01,1)	6 (11,0)

We have also derived the formula for logic-based implementations of the two S-boxes Sb_0 and Sb_1, respectively. Suppose the inputs to the S-boxes are a, b, c, d and the 4-bit outputs are a', b', c', d'.

For Sb_0, we have derived as following:

$$a' = \bar{c}\bar{a} \vee c\bar{b} \vee \bar{a}\bar{d},\ b' = \bar{d}\bar{a} \vee bc \vee acd,\ c' = bd \vee \bar{a}b \vee \bar{a}d,\ d' = c(\bar{a} \vee \bar{b}) \vee d(\bar{a}b \vee a\bar{b}). \quad (3.1)$$

Furthermore, for Sb_1, where $\vee$ represents an OR gate, we have:

$$a' = a\bar{c} \vee a\bar{b} \vee \bar{a}\bar{d}b,\ b' = ad\bar{c} \vee bc \vee \bar{d}b \vee \bar{a}\bar{d}c,\ c' = cd \vee \bar{b}ad \vee \bar{a}b,\ d' = c\bar{a} \vee c\bar{b} \vee \bar{b}\bar{d}. \quad (3.2)$$

We would like to emphasize that other possible signatures can be also utilized. Here are two examples for parity-based and interleaved parity-based approaches.

The following equations show the predicted parity formulations for the two S-boxes Sb_0 and Sb_1, respectively. We have denoted the predicted parities in these formulations by a *"hat"* sign, i.e., $\hat{P}$,

$$\hat{P}_{Sb_0} = \bar{d}ac \vee b\bar{a}\bar{c} \vee \bar{b}\bar{d}(c \vee a) \vee db(\bar{a} \vee \bar{c}),\ \hat{P}_{Sb_1} = \bar{b}(acd \vee \bar{c}\bar{d} \vee \bar{a}\bar{d}) \vee \bar{a}(bd \vee \bar{d}\bar{c}) \vee abc\bar{d}. \quad (3.3)$$

For the interleaved parity-based scheme, we have derived the parities for odd bits and even bits in each of the S-boxes. For Sb_0 and Sb_1, we have:

$$\hat{P}^{(0)}_{Sb_0} = \bar{b}(\bar{a}c \vee \bar{b}\bar{d}) \vee d(cb \vee a\bar{c}),\ \hat{P}^{(1)}_{Sb_0} = \bar{a}\bar{c}(b \vee \bar{d}) \vee \bar{b}d(\bar{a}c \vee a\bar{c}) \vee ac(\bar{d} \vee b), \quad (3.4)$$

$$\hat{P}^{(0)}_{Sb_1} = bd \vee dc\bar{a} \vee a\bar{b}\bar{d} \vee a\bar{c}\bar{d},\ \hat{P}^{(1)}_{Sb_1} = c\bar{b} \vee \bar{a}\bar{c}\bar{d}. \quad (3.5)$$

Different S-boxes are applied in the variants of Midori; for instance, Midori128 applies four different 8-bit S-boxes SSb_i, $0 \leq i \leq 3$. To keep the involution property of S-boxes, each output bit permutation is derived as the inverse of the corresponding input bit permutation. The structure of Midori and the proposed fault diagnosis schemes are presented in Fig. 3.1. This figure shows that four 8-bit outputs of these S-boxes are taken of specific permutation order as shown in Fig. 3.1 (two of the S-boxes are omitted for the sake of brevity). Through the comparison of actual and

predicted parities, we have error indication flags for each Sb_1 in S-boxes of SSb_i as shown in Fig. 3.1 ($e_0 - e_7$). Moreover, both aforementioned parity bits such as single parity and interleaved parity bit have been utilized to create error indication flags. Eventually, one can OR the flags to have a final error indication flag which alters of any faults detected in SSb_i.

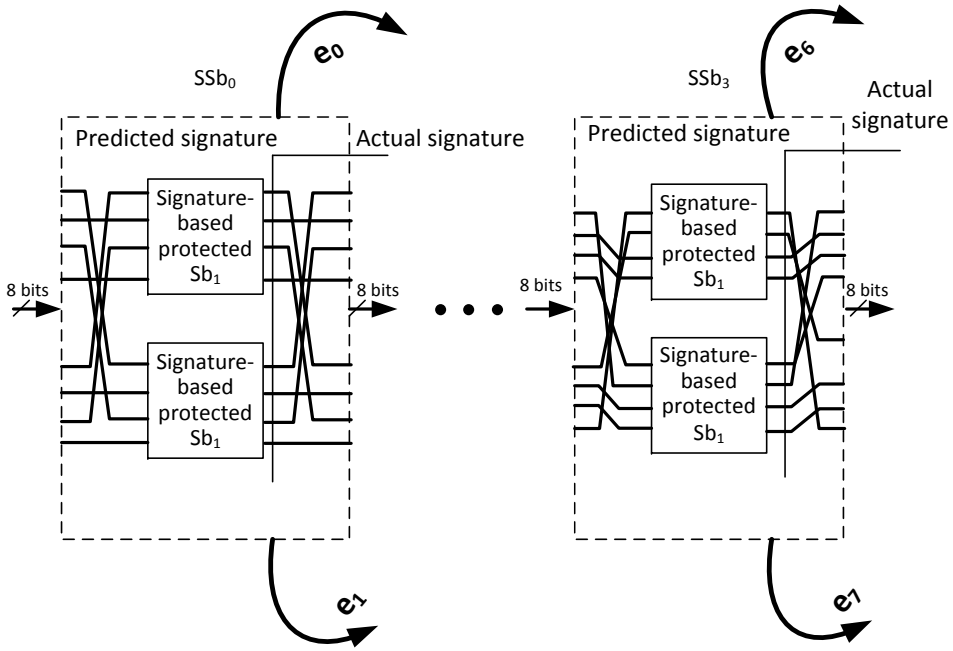

Figure 3.1: Derivation of the error indication flags for the S-boxes in Midori128.

Recomputing with Swapped Inputs (RESI):

We use the method of Recomputing with Swapped Inputs (RESI) as shown in Fig. 3.2 for Midori128 (part of the S-boxes block is shown for the sake of brevity). This method is a subset to the approaches presented in [44]. In this approach, we have swapped the inputs to the S-boxes Sb_1 in each of the four 8-bit S-boxes SSb_i, i.e., the first four inputs are asserted to the second S-box Sb_1 and the next four-bit inputs go to the first one as shown in Fig. 3.2. Then, if the output of each Sb_1 is swapped, it gives the correct results. Finally, we compare the swapped outputs with actual outputs to detect not only transient faults but also permanent faults.

It is noted that the order of permutation of inputs for each SSb_i is different and swapping would be specific for each of 8-bit S-boxes. In the proposed scheme which is based on recomputations, we do not change the original algorithm; nevertheless, we perform the recomputation for detecting the errors; thus, no change is made in the original Midori computation and the overall structure for the original algorithm is intact. Therefore, algorithmic security is not affected in the proposed method as the datapath would use the output of the original Midori algorithm.

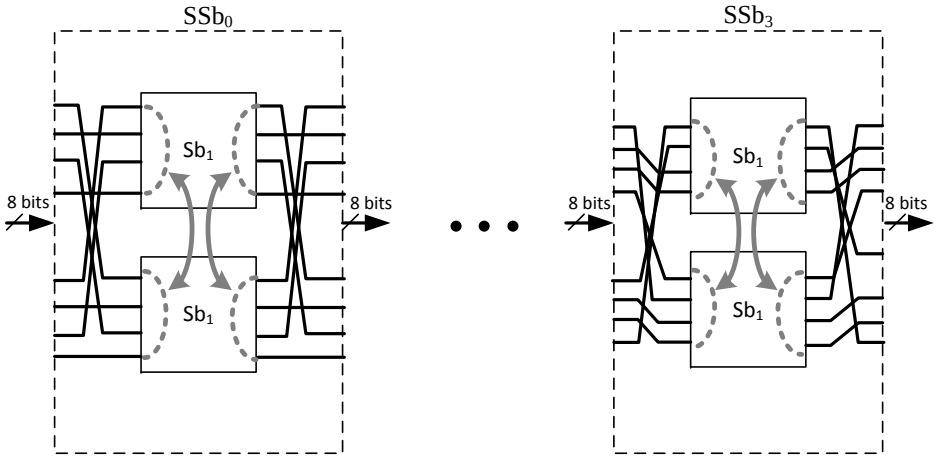

Figure 3.2: The proposed RESI scheme for Midori128.

3.2 Fault Diagnosis of ShuffleCell and KeyAdd

The signature derivation for fault detection in ShuffleCell, such as parity, would be straightforward and can be realized free in hardware due to just rewiring of the elements of 4×4 array state (for instance, parity of inputs is equal to parity of outputs because re-wiring does not affect the computation of parities). We need error detection mechanisms for ShuffleCell (an attacker may try to inject fault by violating setup time for these paths); yet, through using signatures, e.g., parity or interleaved parities, the predicted signatures are equal to the actual signatures of the prior transformation, and that reduces the cost for error detection.

The other operation, KeyAdd, consists of modulo-2 addition of the i-th n-bit round key RK_i with the state S. In this operation, the signature inputs, i.e., state and round-key, are modulo-2 added to derive the signature of output for each round. Suppose the output of KeyAdd is denoted by O and inputs are S and RK_i, $0 \leq i \leq 14$ and $0 \leq i \leq 18$ for Midori64 and Midori128, respectively. Denoting signatures by "$Sig.$", and the predicted signatures of, for instance, the output O which is the function of two inputs by $\hat{Sig.}_{(O)}(S, RK_i)$, we have $\hat{Sig.}_{(O)}(S, RK_i) = Sig.(S) \oplus Sig.(RK_i)$.

3.3 Proposed Design-for-Fault-Detection in MixColumn

Let us denote the input state of MixColumn as S and the output state as S'. Then, we have the following for this operation:

$$S' = M \times S \Longrightarrow \begin{pmatrix} s'_0 & s'_4 & s'_8 & s'_{12} \\ s'_1 & s'_5 & s'_9 & s'_{13} \\ s'_2 & s'_6 & s'_{10} & s'_{14} \\ s'_3 & s'_7 & s'_{11} & s'_{15} \end{pmatrix} =$$

$$\begin{pmatrix} m_0 & m_4 & m_8 & m_{12} \\ m_1 & m_5 & m_9 & m_{13} \\ m_2 & m_6 & m_{10} & m_{14} \\ m_3 & m_7 & m_{11} & m_{15} \end{pmatrix} \times \begin{pmatrix} s_0 & s_4 & s_8 & s_{12} \\ s_1 & s_5 & s_9 & s_{13} \\ s_2 & s_6 & s_{10} & s_{14} \\ s_3 & s_7 & s_{11} & s_{15} \end{pmatrix} \tag{3.6}$$

where each element of input or output state matrix would be 4 bits and 8 bits for Midori64 and Midori128, respectively.

In the two Midori variants, the linear layers consist of the two mentioned operations, ShuffleCell and MixColumn, that are applied over $GF(2^4)$ and $GF(2^8)$ for the 64-bit Midori and 128-bit Midori, respectively. As mentioned for the MixColumn operation, Midori utilizes an involutive binary matrix M, as defined before. For the matrix M, there could be, typically, three types of 4×4 matrices, i.e., involutive MDS (M_A), non-involutive MDS (M_B), and involutive almost MDS (M_C) matrices [14]:

$$M_A = \begin{pmatrix} 1 & 2 & 6 & 4 \\ 2 & 1 & 4 & 6 \\ 6 & 4 & 1 & 2 \\ 4 & 6 & 2 & 1 \end{pmatrix}, \; M_B = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix}, \; M_C = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}. \tag{3.7}$$

Among these matrices, involutive almost MDS (M_C) has been applied more in various lightweight ciphers such as PRINCE due to its efficiency. Furthermore, M_C has low diffusion speed and small number of active S-boxes in each round and has led to increase in the immunity against linear and non-linear attacks. In the proposed fault detection schemes, the objective is to evaluate these three matrices to possibly add a new aspect in how efficient these are when fault diagnosis approaches are used.

For this operation, we present three error detection schemes as detailed in the following.

Scheme 1 (Column Signatures): In the first scheme, we propose modulo-2 addition of the state elements of each column of the output matrix (S'). The theorem is that the result is equal to modulo-2 addition of the state elements of each column of the input matrix (S). Since, modulo-2 addition of each column of matrix M in all of three types of matrices is equal to $'1'$, fault diagnosis through this approach is efficiently performed for the three matrices. In general, for $0 \leq i \leq 3$, we have

$s'_i = m_i s_0 + m_{i+4} s_1 + m_{i+8} s_2 + m_{i+12} s_3$.

Let us modulo-2 add the first column of the state output matrix as following:

$$s'_0 + s'_1 + s'_2 + s'_3 = (m_0 + m_1 + m_2 + m_3)s_0 + (m_4 + m_5 + m_6 + m_7)s_1$$

$$+(m_8 + m_9 + m_{10} + m_{11})s_2 + (m_{12} + m_{13} + m_{14} + m_{15})s_3. \quad (3.8)$$

Moreover, one can derive that each of the coefficients of the input elements, e.g., m_0+m_1+m_2+m_3 is equal to $'1'$. For example, in the case of M_A and for Midori64 that consists of 4-bit elements in the states, we have: $\{1\}_{16}$+$\{2\}_{16}$+$\{6\}_{16}$+$\{4\}_{16} = \{1\}_{16}$. Consequently, modulo-2 addition of each of the output matrix columns is equal to that of the columns of input matrix, i.e., $s'_0 + s'_1 + s'_2 + s'_3 = s_0 + s_1 + s_2 + s_3$. For both variants of Midori, one can derive four 4-bit (Midori64) or 8-bit (Midori128) signatures which can eventually be compared with the actual ones to derive the respective error indication flags.

Scheme 2 (Low-Overhead Union Signature): The second scheme is through modulo-2 addition of all the elements of the output state (union signature), i.e.,

$$s'_0 + s'_1 + ... + s'_{14} + s'_{15} = (m_0 + m_1 + m_2 + m_3)s_0 + (m_4 + m_5 + m_6 + m_7)s_1$$

$$+(m_8 + m_9 + m_{10} + m_{11})s_2 + (m_{12} + m_{13} + m_{14} + m_{15})s_3+$$

$$(m_0 + m_1 + m_2 + m_3)s_4 + ... + (m_{12} + m_{13} + m_{14} + m_{15})s_{15}. \quad (3.9)$$

It is derived that each of these coefficients, e.g., m_0+$m_1 + m_2 + m_3$, is equal to $'1'$ for the aforementioned matrices. As the proof, the binary values of the following hex entries are all equal to $'1'$:

$$\{1\}_{16} + \{2\}_{16} + \{6\}_{16} + \{4\}_{16} = \{0\}_{16} + \{1\}_{16} + \{1\}_{16} + \{1\}_{16}$$

$$= \{2\}_{16} + \{1\}_{16} + \{1\}_{16} + \{3\}_{16} = \{1\}_{16}. \quad (3.10)$$

Therefore, modulo-2 addition of all output elements in the state is equal to that of all input elements in the state (as a nibble or a byte for Midori64 and Midori128, respectively), i.e., $\Sigma s_i = \Sigma s'_i$; where $0 \leq i \leq 15$. In this approach, we have less number of signatures that leads to lower overhead.

Scheme 3 (Interleaved Signatures): The third scheme is through predicting interleaved signatures. We prove that for each two random rows of M_C this is a viable approach whereas it is not a suitable scheme for the other two matrices presented before. Let us, through an example, detail on this scheme. Let us modulo-2 add two even-row elements of the state output state, i.e., rows 0 and 2, as $s'_0 + s'_2 = (m_0+m_2)s_0+(m_4+m_6)s_1+(m_8+m_{10})s_2+(m_{12}+m_{14})s_3$ and two odd-row elements, i.e., rows 1 and 3, as $s'_1+s'_3 = (m_1+m_3)s_0+(m_5+m_7)s_1+(m_9+m_{11})s_2+(m_{13}+m_{15})s_3$. The derived results are two 4-bit or 8-bit predicted interleaved signatures in each column. As computed for the matrices, just in the case of M_C, the interleaved signatures for each column in the input and output states are equal. It is interesting that modulo-2 addition of each two random rows of M_C leads to coefficients of $'1'$ in the aforementioned discussions, e.g., modulo-2 addition of the first and third rows of M_C is equal to $'1010'$, proving $s'_0 + s'_2 = s_0 + s_2$; moreover, modulo-2 addition of the second and fourth rows is $'0101'$ and thus $s'_1 + s'_3 = s_1 + s_3$. However, the two other matrices, i.e., M_A and M_B, do not have this property.

Midori can utilize three 4×4 MDS matrices for the MixColumn transformation; we have compared these three different variants to motivate that error detection needs to be considered as a design factor. In other words, the inventors of Midori have investigated the matrices used in MixColumn to reach the best efficiency for Midori, when the structures have similar algorithmic security. Different categories of the MixColumn transformations are designed based on a wide pool of criteria that can be made smaller by considering the error detection criterion during the design.

The fact that the proposed error detection scheme through interleaved signatures is merely efficient for M_C and the other two proposed schemes can efficiently be applied to all three types of matrices in MixColumn further motivates the urgency of having error detection as a design factor not an afterthought.

3.4 Proposed Approach for Key Schedule

As mentioned before, for both variants of Midori, a 128-bit secret key (K) is applied; however, in the case of Midori64, the key is denoted as two 64-bit subkeys K_0 and K_1 and the whitening key (WK) is derived through modulo-2 addition of these 64-bit subkeys. In key schedule of Midori64, the round key is derived through $RK_i = K_{(imod2)} \oplus \alpha_i$, where $0 \leq i \leq 14$; while in Midori128, $WK = K$ and $RK_i = K \oplus \beta_i$, $0 \leq i \leq 18$, in which $\beta_i = \alpha_i$ for $0 \leq i \leq 14$. The constants are in the form of 4×4 binary matrices which are modulo-2 added to the LSB of the round key nibble in Midori64 and round key byte in Midori128. Therefore, the signature of the round key matrix is either intact (if round constant element is $'0'$) or inverted (if round constant element is $'1'$): $Sig.(\hat{RK}_i) = Sig.(K) \oplus Sig.(\beta_i/\alpha_i)$.

For the sake of brevity, we go over $\alpha_0 = \beta_0, \alpha_{14} = \beta_{14}$, and β_{18} and we do not analyze all the matrices for the constant values; nonetheless, similar approaches can be used for them:

$$\alpha_0 = \beta_0 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}, \alpha_{14} = \beta_{14} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}, \beta_{18} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (3.11)$$

We present two examples for the error detection approach of round key operation.

First, let us denote the input key as K and the output round key as RK_i. Then, we have the following for this operation in Midori128 with constant β_i, where $0 \leq i \leq 18$, $RK_i = K \oplus \beta_i$, and rk_j, k_j, and β^j are the matrices entries for these states, and each element of input or output key matrix would be 4 bits and 8 bits for Midori64 and Midori128, respectively.

In the first example, we derive 16 signatures, where, each signature is for the round key elements, $rk_j = k_j \oplus \beta^j$ where each k_j is a nibble as $k_j^3 k_j^2 k_j^1 k_j^0$ for Midori64 or a byte as $k_j^7 k_j^6 k_j^5 k_j^4 k_j^3 k_j^2 k_j^1 k_j^0$ for Midori128. One signature is derived for each element, for instance, for Midori128, $Sig.\hat{(}rk_j) = Sig.(k_j^7 k_j^6 k_j^5 k_j^4 k_j^3 k_j^2 k_j^1 (k_j^0 \oplus \beta^j))$; where $0 \leq j \leq 15$ and we have intact or inverted elements of round key matrix.

The second scheme is based on modulo-2 addition of the entire elements in the state matrix to create just one signature for error detection. We have one signature as $Sig.(rk) = \Sigma k_j + \Sigma \beta^j$, where $0 \leq j \leq 15$. For instance, for the three matrices $\alpha_0 = \beta_0, \alpha_{14} = \beta_{14}$, and β_{18} in this scheme, we have the modulo-2 addition of all elements of each matrices as $'0', '1'$, and $'0'$, respectively. Therefore, for RK_0 and RK_{18}, the input signatures are intact; while, for RK_{14}, it is inverted.

3.5 Overall Presented Architecture

This chapter is finalized by presenting the overall structures of the presented error detection schemes. The mentioned error detection structures of encryption of Midori128 which consists of 20 rounds with cell size of 8 bits is depicted in Fig. 3.3. The encryption function of this variant consists of the round function and key generation in which the last round has just the SubCell operation and the whitening key is modulo-2 added just in the first and the last steps. As seen in Fig. 3.3, we have shown the respective sections in which we have proposed the error detection

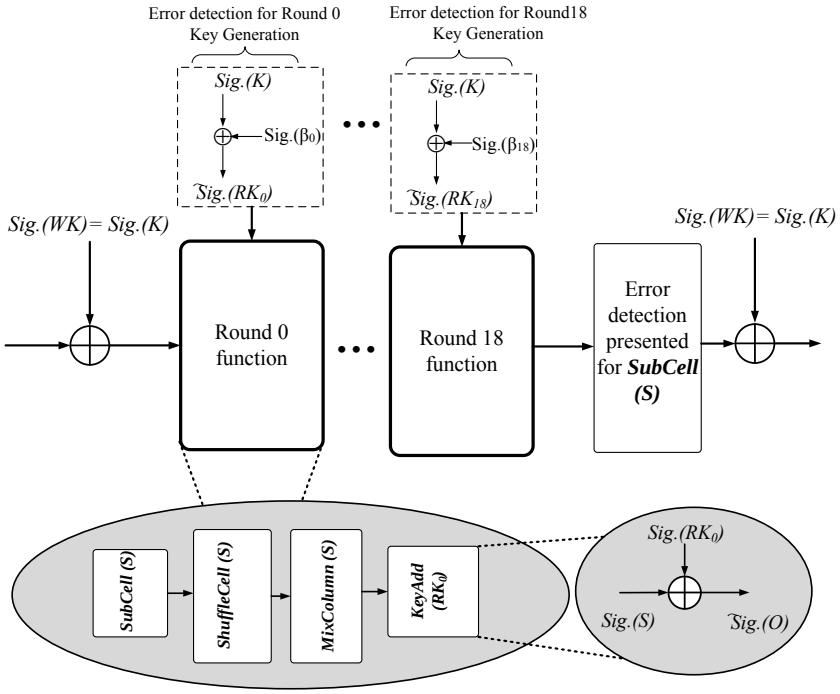

Figure 3.3: The proposed error detection architecture for Midori128.

schemes for different operations. The predicted signatures of each operation in round function are illustrated in Fig. 3.3 as proposed in the aforementioned sections.

Moreover, the Midori's encryption and decryption signature-based error detection architectures are also presented in Fig. 3.3. As seen in this figure, the signatures of the input and the key ("$Sig.$" Input and "$Sig.$" Key) are derived and processed through a loop-like architecture, asserted to a MUX and register (shown in Fig. 3.4). The encryption and decryption include similar operations except for the ShuffleCell (Sh) operation in encryption that is replaced with InvShuffleCell (Sh^{-1}) operation for decryption. The error detection scheme for InvShuffleCell (Sh^{-1}) operation can be adopted based on the explanations in Section 3.2. Moreover, the key generation process is changed in decryption as shown, i.e., $L^{-1}(K)$ instead of K and the corresponding signatures "$Sig.$" ($L^{-1}(K)$) and "$Sig.$" (L), and, correspondingly, the i-th

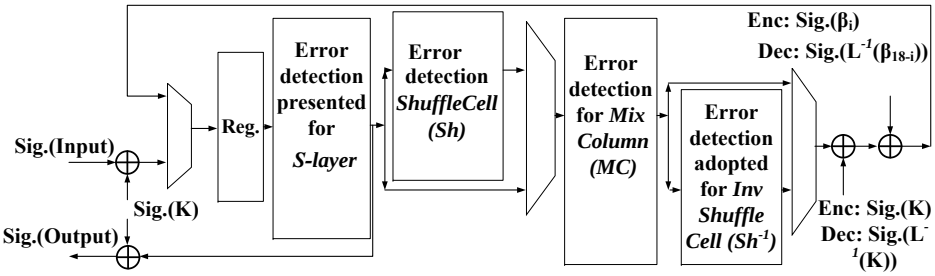

Figure 3.4: Error detection architectures for the combined encryption/decryption unit for Midori128.

round constant is replaced by $L^{-1}(\beta_{18-i})$ instead of β_i, in Fig. 3.4.

We would like to emphasize that the granularity at which the comparisons between the generated and the predicted signatures are to be made has direct effects on reliability, false alarm resiliency, and overhead of both performance and implementation metrics. Let us go over three cases: (a) One can choose to have the check points for the entire encryption/decryption by deriving a "black-box" signature for these processes. In such a case, we have less overhead at the expense of possibility of encountering masking for the error indication flags. The false alarm ratios though are the lowest for this case, as we do not use fine-tuned multiple signatures, (b) one may use the error indication flags of the transformations separately, where the formulations presented in this thesis are used for each of the check points. In such a scheme, error coverage is higher at the expense of more overhead, and the possibility of false alarms, (c) finally, one may choose to have finer granularity, where each of the 4-bit S-boxes of the Midori's 8-bit S-box are checked separately (or, for instance, the columns of the MixColumn transformation are checked individually). At the expense of higher overhead and with higher error coverage, such a scheme may lead to higher false alarms ratios.

To finalize this section, let us provide three examples on the usage models of different signatures. Simple parity codes are capable of detecting odd faults, including

single stuck-at faults which are the ideal cases for fault attacks. Nevertheless, their effectiveness could be limited if fault attacks are mounted intelligently to circumvent such protections. Moreover, VLSI defects could result in burst faults whose detection is not possible through such signatures. Burst faults, e.g., adjacent faults, are detected through interleaved parties at the expense of more overhead. A third usage model would be contrasting 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) which is diverse from an architecture based on robust codes which can detect (with probability) any error. These two solutions have two different goals, the first gives reliability and the second gives hardware security (against fault attacks).

3.6 Proposed Approaches in Presence of Biased Fault Attacks

A subset of fault attacks, differential fault intensity analysis (DFIA), see for instance, [45]-[47] , combines the idea of differential power analysis with fault injection principles to obtain biased fault models; whose merit is that same fault in both the original and redundant computations can be injected, more practically, where not all faults occur with equal probability. Practically, the attacker is interested in using as few faults as possible (preferably single faults with different intensities), to reduce the effort. Previous work argue that the single-bit (more likely in low fault intensity), two-bit, three-bit, and four-bit (more likely in higher intensities) biased fault models can be used to simulate variation of fault intensity. In addition, fault categories presented in [47]-[48] include: Single Bit Upset (SBU), Single Byte Double Bit Upset (SBDBU), Single Byte Triple Bit Upset (SBTBU), Single Byte

Quadruple Bit Upset (SBQBU), Other Single Byte Faults (OSB), and Multiple Byte Faults (MB); the former four corresponding to single/two/three/four-bit models.

The proposed approaches in this thesis based on error detecting codes, column signatures, and RESI are able to thwart a number of such fault models. Specifically, SBUs and SBTBUs are detected fully through the approaches based on error detecting codes and column signatures, using parities. Moreover, through interleaved parities, in addition to burst faults, some SBTBUs, SBQBUs, OSBs, and MBs are detected. Error detecting codes and column signatures (parities) can also detect OSBs and MBs, detailed in the next chapter through simulations. Furthermore, RESI can detect errors with relatively-high error coverage, presented in the next chapter.

Multi-byte faults cannot be used practically for attacking time redundancy countermeasure implementations, e.g., RESI, and single-byte fault models are the only viable option for the attackers [47]. We note that, however, the presented countermeasures based on RESI could fail to detect the occurrence of a fault as long as the adversary could inject the same fault in both the original and redundant computations (biased fault model makes it easier). The proposed RESI architecture (see Fig. 3.2) can be used in conjunction with encoding schemes which nullify the effect of the bias in the fault model by fault space transformation (if two equivalent faults f_0 and f_1 are injected in the output registers, we use a mapping that transforms the fault space), thwarting both these attack schemes, similar the schemes used in [47].

4 Error Injection Simulations and Implementations through FPGA

The error coverage assessment and overhead benchmark of the error detection structures are presented in this chapter. The performed simulation results show high error coverage (the percent of ratio of number of detected errors and the number of injected faults) for the presented schemes. Using the proposed approaches, the error detection structures are capable of detecting the injected faults with high coverage (transient and permanent as well as single, multiple, and adjacent faults). We note that permanent faults, e.g, stuck at faults are caused by VLSI manufacturing defects (and of course if the intention is to break the entire device, such faults can be injected at run-time). There are well established automatic test pattern generation (ATPG)-based testing techniques to identify these faults . On the other hand, "long transient faults" can lead to information leakage . Simple time redundancy cannot detect long transient faults that last for the normal computation and recomputation, and attackers have successfully injected long transient faults to break this countermeasure . Through field-programmable gate array (FPGA) implementations using Xilinx Virtex-7 family, it is shown that the overheads of the proposed architectures are acceptable for resource-constrained applications.

4.1 Error coverage

The performed simulation results show high error coverage (the percent of ratio of number of detected errors and the number of injected faults) for the presented schemes. Using the proposed approaches, the error detection structures are capable of detecting the injected faults with high coverage (transient and permanent as well as single, multiple, and adjacent faults). We note that permanent faults, e.g., stuck at faults are caused by VLSI manufacturing defects (and of course if the intention is to break the entire device, such faults can be injected at run-time). There are well established automatic test pattern generation (ATPG)-based testing techniques to identify these faults. On the other hand, "long transient faults" can lead to information leakage. Simple time redundancy cannot detect long transient faults that last for the normal computation and recomputation, and attackers have successfully injected long transient faults to break this countermeasure. Through FPGA implementations using Xilinx Virtex-7 family, it is shown that the overheads of the proposed architectures are acceptable for resource-constrained applications. Most internal faults are modeled by transient random faults. By relying on simulations, error coverage through multiple stuck-at fault injections is evaluated for the Midori. The results of our performed simulations are for both transient faults and permanent internal faults. We consider both single and multiple stuck-at faults because these model both natural and malicious faults (the reason is that natural faults are usually multiple, and although single stuck-at faults are the ideal cases for the attackers, due to technological constraints, multiple faults occur in reality). The single-bit errors in the nibbles (Midori64) or bytes (Midori128) occurring at the outputs of the linear or nonlinear components of Midori are detected by the presented signature-based error detection approaches. The error coverage of the proposed schemes for these single

stuck-at faults is 100%; therefore, no simulation is required for these cases. The analytic reason is that odd faults are detected using the proposed approaches, as the inherent property of the signatures used, and single stuck-at faults is its subset. We anticipate that the proposed approaches would not detect all of the potential fault attacks but the proposed architectures would make it more difficult to mount, e.g., analytically, more than 99.99% of the faults injected are detected. The reason is that multiple signatures are used for different sub-parts of the architectures which alarm the errors (the error coverage of interleaved parity, for instance, considering different transformations and rounds is: $100 \times (1 - (0.5)^{2m})\%$, where m is the total number of use cases).

The results of our performed simulations are for both transient faults and permanent internal faults. Midori128 encryption has been considered as reference for the fault-injection simulations. Through applying two fault injection experiments, i.e., 10,000 and 100,000 faults (diverse in terms of the type of the fault, its location, and its count), error indication flags are monitored, and the detected errors are counted for the encryption operation. The results of the performed simulations in Table 4.1 show that as the number of injection points is increased, higher error coverage is obtained. We would like to note two points on our experiment methods; (a) we have used linear-feedback shift registers (LFSRs) to inject the faults, when random multiple faults are required, where the location, the type, and the number of faults are chosen by LFSRs with maximum tap polynomials. (b) Starting with injecting 10,000 faults using such a method and through LFSRs for different assertions of the inputs, we have increased the number of injections to get closer to more realistic error coverages. This has been done up to 100,000 injections using the aforementioned method and as seen in Table 4.1, the change in the error coverage, although slight,

Table 4.1: Error coverage of the proposed schemes for Midori128

Type of faults	Injected faults	Detected faults	Error coverage
Stuck-at zero	10,000	9,910	99.10%
	100,000	99,890	99.89%
Stuck-at one	10,000	9,909	99.09%
	100,000	99,782	99.78%

Table 4.2: FPGA implementation results for the original Midori128 encryption and its proposed error detection scheme on Virtex-7 FPGA for xc7vx330t

Architecture	Area (occupied slices)	Delay (ns) / Frequency (MHz)	Power (mW)	Throughput (Gbps)	Efficiency (Mbps/slices)
Midori128 (LUT-based)	155	2.70 (370.37)	340	47.41	305.9
Signature-based error detection for LUTs	161 (3.9%)	2.81 (4.07%) (355.87)	367 (7.9%)	45.55 (3.9%)	282.9 (7.5%)
Midori128 (logic-based)	157	2.79 (358.42)	349	45.88	292.2
Signature-based error detection for logic-based	171 (8.9%)	3.01 (7.88%) (332.22)	396 (13.5%)	42.52 (7.3%)	248.6 (14.9%)

hints to what we realistically would get through exhaustive search.

Midori's S-boxes and MixColumn operations consume much of the area and power consumption of the block cipher (compared to ShuffleCell and KeyAdd). Therefore, these are the prominent operations for protecting against injected faults. We have presented in this thesis three schemes for each of these operations, i.e., parity/interleaved parity/swapping the inputs for the S-boxes, and element-wise, column-wise, and matrix-wise signatures for MixColumn. Alleviating the error coverage of Midori can be performed through using the schemes with higher error coverage, i.e., swapping the inputs for the S-boxes combined with the element-wise error detection approach for MixColumn, at the expense of higher overhead incurred. Such an improvement would be a compromise between the reliability requirements and overhead tolerance.

4.2 Implementation Results

The results of the FPGA overhead assessments of our proposed schemes are pre-

sented in this section. Through this benchmarking, the overheads (degradation) are derived for various metrics of the presented schemes. The ISE version 14.7 and Virtex-7 FPGA family (device: xc7vx330t) have been utilized for the FPGA implementations. VHDL has been used as the design entry for the original and the error detection structures. As seen in Fig. 4.1, we have also shown the structure of the

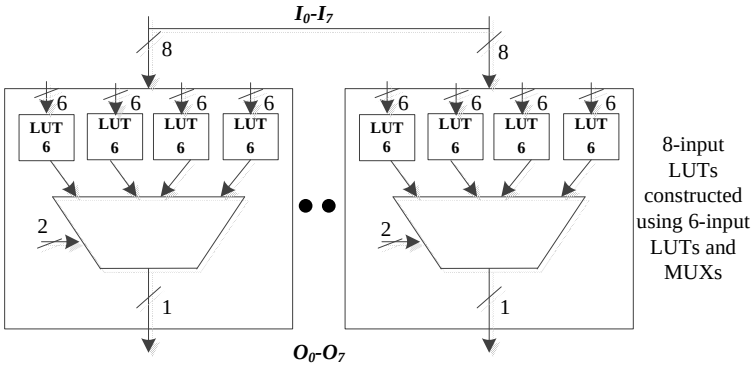

Figure 4.1: The structure of Midori128 8-input S-box implemented using Virtex-7 6-input LUTs.

8-input S-box of Midori128 using 6-input LUTs (6LUT in Fig. 4.1) and multiplexers (MUXs), leading to 32 of 6LUTs.

We present the results for two of the proposed schemes in this thesis tabulated in Table 4.2. Based on the simulation results in this section, these overheads are added for the error coverage of very close to 100%. The proposed fault diagnosis approaches provide high error coverage at the expense of acceptable overheads, making the hardware architectures of Midori more reliable. We would like to emphasize that it is expected to have similar overheads for other FPGA families and also the designs on application-specific integrated circuit (ASIC).

We finalize this section by investigating differential fault attacks on a similar lightweight block cipher to Midori, PRESENT, to study potential fault attacks on this cipher as well. It is noted that many prior works have been done on fault attacks of standardized ciphers such as the AES and the DES [49]-[51]. In such attacks on lightweight

ciphers like PRESENT, the attacker compares the correct ciphertext and the faulty one (caused by faulty operation) to obtain the secret key [52]. Mounting such attacks can be based on single-bit faults, single-nibble faults, or multiple-nibble faults which are adopted into the key schedule algorithm during the encryption operation. According to such attacks on PRESENT with SPN structure (or even for the functions in CLEFIA which is a Feistel network), potential attacks on Midori variants can be mounted (which have round-based architectures similar to PRESENT), to recover the subkeys in the rounds [14], [52]-[54]. In differential fault attacks, the attackers try to inject the faulty nibble in a specific round in order to discover the secret key by examining a group of correct and faulty ciphertexts. After that, the attack is based on solving a number of fault equations which are obtained through the propagation of the injected faults through the remainder of the encryption structure. We anticipate that differential fault attacks, similarly, can be mounted on Midori. Finally, we would like to note that Midori (and other block ciphers) include controller architectures when practically implemented. Such control blocks are also susceptible to natural and malicious faults, see, for instance, [55], and the remedies proposed based on programmable state flipflops.

5 Conclusion

In this thesis, we have presented new error detection schemes for low-energy block ciphers such as Midori. This would be an add-on to the work of our research group on reliability and cryptographic engineering [56]-[101]. To sum up, this thesis presents for the first time, the fault diagnosis approaches for the energy-efficient, lightweight block cipher Midori. For the S-boxes within Midori, we have derived and implemented both LUT-based and logic gate variants, and proposed fault diagnosis schemes that can be tailored based on the reliability and overheard objectives. The MixColumn operation has been examined to achieve a number of schemes, and the selection of the matrices within has been carefully done to have low-overhead detection approaches.

Through FPGA implementations using Xilinx Virtex-7 family, it has been shown that the overheads of the proposed architectures are acceptable for resource-constrained applications. We have also presented sketches of possible fault attacks as the motivation to the presented work as well as biased faults considerations, Virtex-7 specific implementation of Midori128 S-boxes, and usage models for three select signatures. More reliable architectures for Midori are achieved through the proposed detection schemes and they can be tailored based on the objectives in terms of reliability and overhead tolerance.

5.1 Future Works

In this thesis, the proposed schemes are based on signature approach and swapping inputs unlike the traditional time and space redundancy countermeasures to meet the main purpose of this work. The main objective of this thesis is towards achieving a low-area and low-power consuming architecture for lightweight ciphers along with roughly 100% fault detection.

One future scope which can be helpful for this work is that one could analyze combined power and fault analysis attack resiliency. Efficiently-maskable approaches provide viable solutions for considering thwarting power analysis attacks before the design phase.

Countermeasures based on recomputations could fail to detect the occurrence of a fault as long as the adversary could inject the same fault in both the original and redundant computations (biased fault model makes it easier). The countermeasures based on recomputations can be used in conjunction with encoding schemes which nullify the effect of the bias in the fault model by fault space transformation (if two equivalent faults f_0 and f_1 are injected in the output registers, we use a mapping that transforms the fault space), thwarting both these attack schemes, similar the schemes used in [48].

During the implementation, type of used platform which determines the hardware fabric that will be utilized for implementation, is considered as a significant factor for deeply-embedded systems.. These proposed schemes for this cipher were implemented on Virtex FPGA families. However, the proposed design can be implemented on other FPGA families and the results can be benchmarked. Similarly, compared to an FPGA, implementations on ASIC platforms might be suitable in terms of area and power optimizations for some applications. Therefore, ASIC implementation

can be the next part of the proposed work.

Bibliography

[1] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. CRC press, 1996.

[2] C. Paar and J. Pelzl, *Understanding cryptography: a textbook for students and practitioners*. Springer Science & Business Media, 2009.

[3] D. R. Stinson, *Cryptography: theory and practice*. CRC press, 2005.

[4] M. Mozaffari-Kermani, M. Zhang, A. Raghunathan, and N. K. Jha, "Emerging frontiers in embedded security," in *VLSI Design and 2013 12th International Conference on Embedded Systems (VLSID), 2013 26th International Conference on*. IEEE, 2013, pp. 203–208.

[5] T. Eisenbarth and S. Kumar, "A survey of lightweight-cryptography implementations," *IEEE Design & Test of Computers*, vol. 24, no. 6, 2007.

[6] A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang, "Pushing the limits: a very compact and a threshold implementation of AES," in *Eurocrypt*, vol. 6632. Springer, 2011, pp. 69–88.

[7] 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.

[8] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, "The 128-bit blockcipher CLEFIA," in *FSE*, vol. 4593. Springer, 2007, pp. 181–195.

[9] J. Borghoff, A. Canteaut, T. Güneysu, E. B. Kavun, M. Knezevic, L. R. Knudsen, G. Leander, V. Nikov, C. Paar, C. Rechberger *et al.*, "Prince–a low-latency block cipher for pervasive computing applications," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2012, pp. 208–225.

[10] R. Beaulieu, S. Treatman-Clark, D. Shors, B. Weeks, J. Smith, and L. Wingers, "The SIMON and SPECK lightweight block ciphers," in *Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE*. IEEE, 2015, pp. 1–6.

[11] 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.

[12] E. M. Shakshuki, N. Kang, and T. R. Sheltami, "EAACKâa secure intrusion-detection system for MANETs," *IEEE transactions on Industrial Electronics*, vol. 60, no. 3, pp. 1089–1098, 2013.

[13] N. F. Pub, "197: Advanced encryption standard (AES), federal information processing standards publication 197, US department of commerce/NIST, november 26, 2001. available from the NIST website."

[14] 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, Report 2015/1142, 2015. http://eprint. iacr. org, Tech. Rep.

[15] J. Guo, J. Jean, I. Nikolic, K. Qiao, Y. Sasaki, and S. M. Sim, "Invariant subspace attack against full Midori64." *IACR Cryptology ePrint Archive*, vol. 2015, p. 1189, 2015.

[16] L. Lin and W. Wu, "Meet-in-the-middle attacks on reduced-round Midori64," *IACR Transactions on Symmetric Cryptology*, vol. 2017, no. 1, pp. 215–239, 2017.

[17] A. Moradi and T. Schneider, "Side-channel analysis protection and low-latency in action: –case study of PRINCE and Midori–," in *Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22*. Springer, 2016, pp. 517–547.

[18] J. Daemen, M. Peeters, G. Van Assche, and V. Rijmen, "Nessie proposal: Noekeon," in *First Open NESSIE Workshop*, 2000, pp. 213–230.

[19] M. Otto, "Fault attacks and countermeasures." Ph.D. dissertation, University of Paderborn, Germany, 2005.

[20] M. Karpovsky, K. Kulikowski, and A. Taubin, *Differential Fault Analysis Attack Resistant Architectures for the Advanced Encryption Standard*. Springer US, 2004.

[21] 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.

[22] 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.

[23] A. Tenca and M. Ercegovac, "A variable long-precision arithmetic unit design for reconfigurable coprocessor architectures," *Proc. IEEE Symp. FPGAs for Custom Computing Machines*, pp. 216 – 225, April 1998.

[24] G. Xiaofei 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, Sep. 2013.

[25] S. Durand and C. Piguet, "FPGA with self repair capabilities," *Int. ACM/SIGDA Workshop on Field Programmable Gate Arrays*, pp. 1 – 10, Feb. 1994.

[26] J. Emmert, C. Stroud, and M. Abramovici, "Online fault tolerance for FPGA logic blocks," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 2, pp. 216 – 226, Feb. 2007.

[27] 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, June 2006.

[28] G. D. 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. 4, pp. 269 – 278, Aug. 2009.

[29] 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.

[30] 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.

[31] 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, 2008.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

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

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] A. Satoh, T. Sugawara, N. Homma, and T. Aoki, "High-performance concurrent error detection scheme for AES hardware," *Proc. CHES*, pp. 110–112, 2008.

[43] 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.

[44] X. Guo and R. Karri, "Invariance-based concurrent error detection for Advanced Encryption Standard," in *Proceedings of the 49th Annual Design Automation Conference*. ACM, 2012, pp. 573–578.

[45] N. F. Ghalaty, B. Yuce, and P. Schaumont, "Analyzing the efficiency of biased-fault based attacks," *IEEE Embedded Systems Letters*, vol. 8, no. 2, pp. 33–36, 2016.

[46] N. F. Ghalaty, B. Yuce, M. Taha, and P. Schaumont, "Differential fault intensity analysis," in *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2014 Workshop on*. IEEE, 2014, pp. 49–58.

[47] S. Patranabis, A. Chakraborty, P. H. Nguyen, and D. Mukhopadhyay, "A biased fault attack on the time redundancy countermeasure for AES," in *International Workshop on Constructive Side-Channel Analysis and Secure Design*. Springer, 2015, pp. 189–203.

[48] S. Patranabis, A. Chakraborty, D. Mukhopadhyay, and P. Chakrabarti, "Using state space encoding to counter biased fault attacks on AES countermeasures." *IACR Cryptology ePrint Archive*, vol. 2015, p. 806, 2015.

[49] E. Biham and A. Shamir, "Differential fault analysis of secret key cryptosystems," *Advances in Cryptology CRYPTO 97*, pp. 513–525, 1997.

[50] G. Piret and J.-J. Quisquater, "A differential fault attack technique against SPN structures, with application to the AES and KHAZAD," in *CHES*, vol. 2779. Springer, 2003, pp. 77–88.

[51] M. Tunstall, D. Mukhopadhyay, and S. Ali, "Differential fault analysis of the Advanced Encryption Standard using a single fault." *WISTP*, vol. 6633, pp. 224–233, 2011.

[52] G. Wang and S. Wang, "Differential fault analysis on PRESENT key schedule," in *Computational Intelligence and Security (CIS), 2010 International Conference on*. IEEE, 2010, pp. 362–366.

[53] S. S. Ali and D. Mukhopadhyay, "Improved differential fault analysis of CLEFIA," in *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2013 Workshop on*. IEEE, 2013, pp. 60–70.

[54] J. Krämer, A. Stüber, and Á. Kiss, "On the optimality of differential fault analyses on CLEFIA," *IACR Cryptology ePrint Archive*, vol. 2014, p. 572, 2014.

[55] A. Nahiyan, K. Xiao, K. Yang, Y. Jin, D. Forte, and M. Tehranipoor, "AVFSM: a framework for identifying and mitigating vulnerabilities in FSMs," in *Design Automation Conference (DAC), 2016 53nd ACM/EDAC/IEEE*. IEEE, 2016, pp. 1–6.

[56] 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.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] 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.

[67] 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.

[68] 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.

[69] 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.

[70] 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.

[71] 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.

[72] 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.

[73] 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.

[74] 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.

[75] 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.

[76] 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.

[77] 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.

[78] 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.

[79] 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.

[80] 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.

[81] 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.

[82] A. Aghaie, M. Mozaffari-Kermani, and R. Azarderakhsh, "Reliable and error detection architectures of Pomaranch for false-alarm-sensitive cryptographic applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 12, pp. 2804 – 2812, Dec. 2015.

[83] 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.

[84] Y. Yoo, R. Azarderakhsh, A. Jalali, D. Jao, and V. Soukharev, "A post-quantum digital signature scheme based on supersingular isogenies." *IACR Cryptology ePrint Archive*, vol. 2017, p. 186, 2017.

[85] A. Aghaie, M. Kermani-Mozaffari, and R. Azarderakhsh, "Fault diagnosis schemes for low-energy block cipher Midori benchmarked on FPGA," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 4, pp. 1528–1536, 2017.

[86] A. Jalali, R. Azarderakhsh, M. Mozaffari-Kermani, and D. Jao, "Supersingular isogeny Diffie-Hellman key exchange on 64-bit ARM," *IEEE Transactions on Dependable and Secure Computing*, 2017.

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

[88] 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.

[89] B. Koziel, A. Jalali, R. Azarderakhsh, D. Jao, and M. Mozaffari-Kermani, "NEON-SIDH: efficient implementation of supersingular isogeny Diffie-Hellman key exchange protocol on ARM," in *International Conference on Cryptology and Network Security*. Springer, 2016, pp. 88–103.

[90] M. Mozaffari-Kermani, R. Azarderakhsh, and J. Xie, "Error detection reliable architectures of camellia block cipher applicable to different variants of its substitution boxes," in *Hardware-Oriented Security and Trust (AsianHOST), IEEE Asian*. IEEE, 2016, pp. 1–6.

[91] B. Koziel, R. Azarderakhsh, and M. Mozaffari-Kermani, "Fast hardware architectures for supersingular isogeny diffie-hellman key exchange on FPGA," in *Progress in Cryptology–INDOCRYPT 2016: 17th International Conference on Cryptology in India, Kolkata, India, December 11-14, 2016, Proceedings 17*. Springer, 2016, pp. 191–206.

[92] M. Mozaffari-Kermani, R. Azarderakhsh, and A. Aghaie, "Fault detection architectures for post-quantum cryptographic stateless hash-based secure signatures benchmarked on ASIC," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 2, p. 59, 2016.

[93] A. Aghaie, M. Mozaffari-Kermani, and R. Azarderakhsh, "Fault diagnosis schemes for secure lightweight cryptographic block cipher RECTANGLE

benchmarked on FPGA," in *Electronics, Circuits and Systems (ICECS), 2016 IEEE International Conference on.* IEEE, 2016, pp. 768–771.

[94] M. Mozaffari-Kermani and R. Azarderakhsh, "Lightweight hardware architectures for fault diagnosis schemes of efficiently-maskable cryptographic substitution boxes," in *Electronics, Circuits and Systems (ICECS), 2016 IEEE International Conference on.* IEEE, 2016, pp. 764–767.

[95] M. Mozaffari-Kermani, R. Azarderakhsh, and M. Mirakhorli, "Multidisciplinary approaches and challenges in integrating emerging medical devices security research and education," 2016.

[96] B. Koziel, R. Azarderakhsh, M. Mozaffari Kermani, and D. Jao, "Post-quantum cryp-tography on FPGA based on isogenies on elliptic curves," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 1, pp. 86–99, 2017.

[97] M. Mozaffari-Kermani, V. Singh, and R. Azarderakhsh, "Reliable low-latency viterbi algorithm architectures benchmarked on ASIC and FPGA," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 1, pp. 208–216, 2017.

[98] A. Jalali, R. Azarderakhsh, and M. Mozaffari-Kermani, "Efficient post-quantum undeniable signature on 64-bit ARM," *Conf. Selected Areas in Cryptography (SAC)*, to appear in 2017.

[99] P. Ahir, M. Mozaffari-Kermani, and R. Azarderakhsh, "Lightweight architectures for reliable and fault detection Simon and Speck cryptographic algorithms on FPGA," *ACM Trans. Embedded Comput. Syst.*, to appear in 2017-2018.

[100] M. Mozaffari-Kermani, A. Jalali, R. Azarderakhsh, J. Xie, and R. Choo, "Reliable inversion in GF 2‘8 with redundant arithmetic for secure error detection of cryptographic architectures." *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, to appear in 2017-2018.

[101] S. Subramanian, M. Mozaffari Kermani, R. Azarderakhsh, and M. Nojoumian, "Reliable hardware architectures for cryptographic block ciphers LED and HIGHT," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, to appear in 2017-2018.