Semiconductor Optoelectronics
ISSN: 1001-5868

RELIABLE CRC BASED ERROR DETECTION TECHNIQUES FOR FINITE FIELD MULTIPLIERS

Dr. A. Gangadhar[1], Dr. G. Appala Naidu[2]
Assistant Professor ECE Department, UCEN-JNTUK, Narasaraopet, Andhra Pradesh, India-522601
Assistant Professor ECE Department, JNTUGV-CEV, Vizianagaram, Andhra Pradesh, India-535003

ABSTRACT:
The study of finite-subject multiplication has garnered significant attention in the literature, primarily due to its relevance in error-detecting codes and cryptography. This mathematical framework is known for its complexity, high computational costs, and time-consuming nature, often necessitating the use of a substantial number of logic gates in cryptographic algorithms. Through our investigation, particularly within the context of the Luov cryptographic algorithm, we propose robust hardware architectures based on cyclic redundancy check (CRC) as effective error- detection mechanisms for post-quantum cryptography (PQC). Luov is a finalist in the second round of the PQC standardization competition conducted by the National Institute of Standards and Technology (NIST).
The chosen CRC polynomials are compatible with both the field widths and the essential error-detection capabilities required for the cryptographic algorithms. Additionally, we have developed verification codes that enable the implementation of the proposed schemes in software, ensuring the accuracy and correctness of the derived formulations. To further validate our findings, we have utilized a Xilinx field- programmable gate array (FPGA) to construct original multipliers incorporating the recommended error-detection techniques in hardware. Our results demonstrate that the suggested systems effectively achieve high error coverage with reasonable resource overhead.
Keywords: CRC, FPGA, SHA3, PQC, MSB,LSB, CMOS.

I INTRODUCTION

CRCs, or cyclic redundancy checks,[5] are built upon the principles of cyclic error-correcting codes. The concept of using systematic cyclic codes, which involve appending a fixed-length check value to messages, for error detection in communication networks, was originally introduced by W. Wesley Peterson in 1961. Cyclic codes offer a straightforward implementation and possess a significant advantage in detecting burst errors, which are consecutive sequences of erroneous data symbols within messages. This capability is particularly valuable because burst errors are prevalent in various communication channels, such as magnetic and optical storage devices.
In practice, when an n-bit CRC[4] is applied to a data block of variable length, it can identify any single error burst that is no longer than n bits. Furthermore, it can detect a proportion of

longer error bursts, with the fraction of detected longer bursts being equal to (1 - 2^(-n)). Specification of a CRC code requires definition of a so-called generator polynomial. This polynomial becomes the divisor in a polynomial long division, which takes the message as the dividend and in which the quotient is discarded and the remainder becomes the result. The important caveat is that the polynomial coefficients are calculated according to the arithmetic of a finite field, so the addition operation can always be performed bitwise- parallel (there is no carry between digits). In practice, all commonly used CRCs employ the Galois field, or more simply a finite field, of two elements, GF(2). The two elements are usually called 0 and 1, comfortably matching computer architecture.

A CRC is called an n-bit CRC when its check value is n bits long. For a given n, multiple CRCs are possible, each with a different polynomial. Such a polynomial has highest degree n, which means it has n + 1 terms. In other words, the polynomial has a length of n + 1; its encoding requires n + 1 bits. Note that most polynomial specifications either drop the MSB or LSB, since they are always 1. The CRC and associated polynomial typically have a name of the form CRC-n-XXX as in the table below. The simplest error- detection system, the parity bit, is in fact a 1-bit CRC: it uses the generator polynomial x + 1 (two terms),[3] and has the name CRC-1. CRC checks (cyclic redundancy checks) are the most Commonly used checks in data communication. In embedded software development, CRC algorithm is often used to verify various data. Therefore, mastering basic CRC algorithms[7] should be a basic skill for embedded programmers.

However, few embedded programmers I know can really master the CRC algorithm. Most of the CRC code that I usually see in a project is a very inefficient implementation. In addition, since most embedded programmers are traveling halfway from home, many people only use C. Therefore, the sample code in this paper is all implemented in C language. As an introductory short article, the code given here focuses more on demonstration and is as readable as possible. Therefore, the code in this article does not seek the most efficient implementation, but is fast enough for general applications.

II RELATED STUDY

"Reliable Hardware Architectures For The Third-Round SHA-3 Finalist Grostl Benchmarked On FPGA Platform"

The third round of the SHA-3 candidate competition is currently underway to determine the selected cryptographic function for 2012. While much attention has been given to assessing the performance and security of these candidates, little has been done to enhance their trustworthiness. In this study, a novel fault detection technique for the SHA-3[2] third-round candidate Grostl, inspired by the approach used for AES Encryption, is introduced for the first time.

This developes a low-overhead flaw detection method by employing closed formulas to calculate the expected signatures of various factors within this SHA-3 second finalist. These low-overhead signatures encompass one or multiple bit parities and ASCII projected signatures. To evaluate the proposed dependable hardware architecture for Grostl, we have implemented it on Xilinx FPGA family hardware, examining its hardware characteristics and temporal performance. The assessments demonstrate that the proposed approach offers a strong

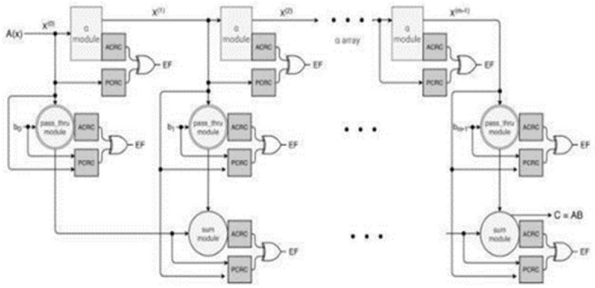
level of error coverage with an acceptable level of overhead, making it a valuable contribution to enhancing the trustworthiness of Grostl in the SHA-3 candidate competition.

"A low-cost S-box for the advanced encryption standard using normal basis"

For the safe transfer of data blocks, the strong cryptographic standard (AES) seems to be a secret based on cryptographic standard that has recently gained widespread acceptance. When implementing the AES in hardware, the Sub Bytes transformation consumes the greatest chip area and power in comparison to the other transformations. S-box hardware optimization iscrucial to a low cost AES because it has 16 of them. We describe a low- cost AES S-box in this paper. For the S-box, digital logic architecture based on a known limited composite field employing normal basis is used. In order to simplify implementations, we then give newformulations for inversion in the S-sub-fields. Our ASIC construction of the suggested S-box employing 0:18mu CMOS technology iscompared to the previous ones after we've studied the new architecture's complexity. According to the results, the provided scheme uses the least amount of power and takes up the smallest amount of space when compared to its competitors in the review papers.

EXISTING DESIGN:

Finite-field multiplication has received prominent attention among many modern, sensitive applications and systems that use finite-field operations in their schemes. Finite-field multipliers perform multiplication modulo, an irreducible polynomial used to define the finitefield. For post quantum cryptography (PQC), the inputs can be very large, and the finite-field multipliers may require millions of logic gates.Therefore, it is a complex task to implement such architectures resilient to natural and malicious faults; consequently, research has focused on waysto eliminate errors and obtain more reliability withacceptable overhead. Moreover, there has been previous work on countering fault attacks and providing reliability for PQC.



III RECOMMENDED SYSTEM

Cyclic Redundancy Check (CRC) codes constitute a well-known special case of checksum functions, which are typically used for packet Fig.1. CR C 16 RTL model.error detection in a wide variety of low-layer protocols. Their main purpose is to validate the integrity of received packets. If an error is detected by such codes, the corrupted packet is normally discarded and a data recovery mechanism can be set, as implemented in protocols such as the Transmission Control Protocol (TCP), where reliability is ensured through retransmission of the corrupted data. In order to avoid systematic retransmission, which would lead to an increased amount of data and extra delays within the network, error correction methods have been proposed at the receiver side. In addition, error detection codes such as CRCs and Checksums have also been

demonstrated to allow error correction. The principle of CRC error detection is based on the computation of a so- called CRC field at the transmitter side. The value of this field is the remainder of the long division of the protected bit sequence, the data, which we will refer to as the payload, denoted d(x) , by a generator polynomial (a binary polynomial of degree n defined by the protocol used, denoted g(x)). The payload is left- shifted by n positions before the division.

The proposed technique utilizes the CRC syndrome value, denoted as s(x), which is computed at the receiver's end. This value is employed to generate a comprehensive list of potential error patterns that can produce the specific syndrome, taking into account the presence of a maximum number of errors. This resultant list may encompass one or multiple entries upon completion. Each entry signifies the positions of bits that need correction to restore a CRC-valid packet, effectively revealing the locations of errors. When the list comprises a solitary element, it permits instant correction of the packet. However, if the list contains multiple entries, additional information becomes necessary to pinpoint the precise error pattern among the potential candidates. The proposed approach offers flexibility by presenting the complete range of conceivable error patterns, accommodating up to N errors, where the parameter N can be adjusted based on the observed channel conditions.

In the following section, we initially introduce the fundamental theoretical concepts of the proposed method within the context of single- error scenarios. Subsequently, we expand upon the method to encompass double-error patterns, followed by the treatment of multiple-error patterns.

OPERATION:

To obtain the exhaustive list of error patterns, we aim at expanding the error range to have it cover the entire length of the protected data. The method we propose is to force a bit to 1 during the process. Forcing a position consists in setting it (or leaving it) to 1 during the single-error search. In other words, it is equivalent to making the hypothesis that a specific position is actually erroneous in the packet. Hence, we force one bit to 1 at position F1 during the process and run the single-error algorithm on the remaining length of the packet. If the bit is already 1, we leave it untouched. Otherwise, setting a bit to 1 is done by applying an XOR operation with g(x) at position F1 in order to maintain the equivalence relation. Throughout the cancelation process with the forced bit set, if a single-error position (denoted hereafter P1) is obtained from the single-error correction algorithm, we determine a double-error pattern with errors at positions F1 and P1 .

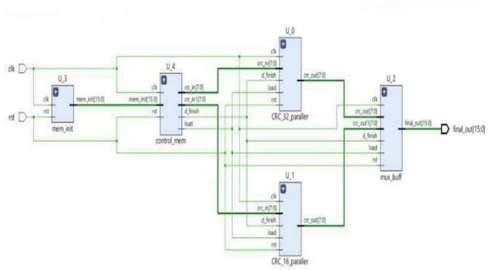


Fig.1. CRC 16 RTL model.

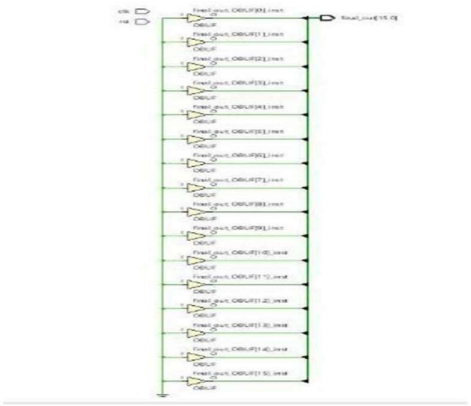


Fig.2. Schematic diagram

As we want to get the whole list and we do not know the actual position of the first error, we test each possible forced position in order to output all the double-error patterns associated with the computed syndrome. In the proposed algorithm, we suggest forcing positions starting from LSB to MSB. Moreover, starting from LSB at each tested forced position would lead to a cancelation of the same first positions several over and degrade the computational efficiency. To avoid verifying the same possibilities repeatedly, we store the value of e when a bit is forced and recall this state to start from it and save computations for the next forced position to test.

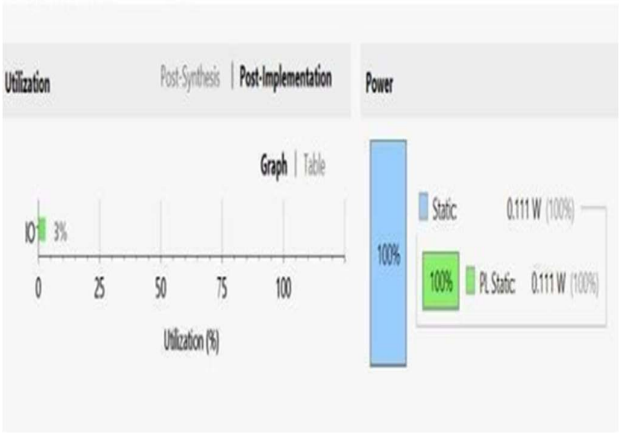


Fig.3. Simulation results

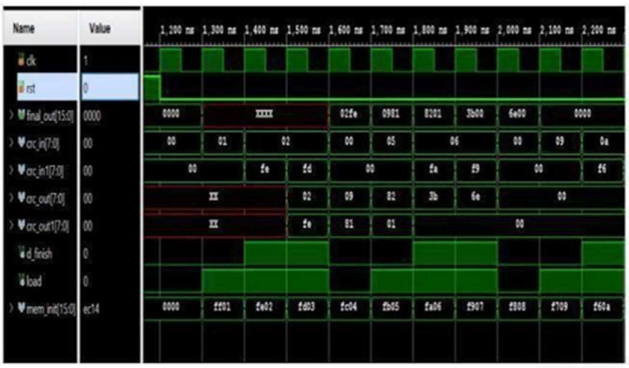


Fig.4. Simulation results 2

Fig.5. Power output

Fig.6. Power Rating.

IV CONCLUSION

A novel CRC-aided error pattern estimation technique, along with proposed even for cases with hundreds of Non-Unique Redundancy (NUR).

This technique can be applied to a wide range of receivers that generate soft decision values, including soft output Viterbi decoders and turbo codes. By incorporating CRC checks into the memory, we have enhanced the effectiveness of error rate corrections. In Table 1, we present the results of our proposed algorithms, showcasing their error reduction capabilities and their impact on memory performance when compared to existing methods.

REFERENCES

[1] J. L. Danger et al., "On the performance and security of multiplication in GF(2N)," Cryptography, vol. 2, no. 3, pp. 25–46, 2018.

[2] M. Mozaffari-Kermani and A. Reyhani- Masoleh, "Reliable hardware architectures for the third-round SHA-3 finalist Grostl benchmarked on FPGA platform," in Proc. DFT, Oct. 2011, pp.325– 331.

[3] M. Mozaffari-Kermani and A. Reyhani- Masoleh, "A low-cost S-box for the advanced encryption standard using normal basis," in Proc. IEEE Int. Conf. Electro/Inf. Technol., Jun. 2009, pp. 52–55.

algorithms, was developed to handle a single soft[1] M. Yasin, B. Mazumdar, S. S. Ali, and O. decision sequence. Unlike previous methods with larger search spaces, our technique limits the search space to significantly reduce worst-case complexity. We introduced a new definition of optimality for error pattern sets and demonstrated the efficient incremental generation of optimal error patterns, Sinanoglu, "Security analysis of logic encryption against the most effective side-channel attack: DPA," in Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst.(DFTS), Oct. 2015, pp. 97–102.

[2] M Mozaffari-Kermani, R. Azarderakhsh, A. Sarker, and A. Jalali, "Efficient and reliable error detection architectures of hash-counter-hash . tweakable enciphering schemes," ACM Trans.

Embedded Comput. Syst., vol. 17, no. 2, pp. 54:1– 54:19, May 2018.

[3] M. Mozaffari-Kermani, R. Azarderakhsh, and

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

[4] A. Sarker, M. Mozaffari-Kermani, and R. Azarderakhsh, "Hardware constructions for error detection of numbertheoretic transform utilized in secure cryptographic architectures," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 27, no. 3, pp. 738–741, Mar. 2019.

[5] M. Mozaffari-Kermani, R. Azarderakhsh, and

A. Aghaie, "Fault detection architectures for post- quantum cryptographic stateless hash-based secure signatures benchmarked on ASIC," ACM Trans. Embedded Comput. Syst., vol. 16, no. 2, pp. 59:1– 59:19, Dec. 2016.

[6] M. Mozaffari-Kermani and R. Azarderakhsh, "Reliable hash trees for post-quantum stateless cryptographic hashbased signatures," in Proc.

IEEE Int. Symp. Defect Fault Tolerance VLSI

[7] A. Sarker, M. Mozaffari-Kermani, and R. Azarderakhsh, "Hardware constructions for error detection of numbertheoretic transform utilized in secure cryptographic architectures," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 27, no. 3, pp. 738–741, Mar. 2019.