International Journal of Gender, Science and Technology
..ISSN: 2040-0748
UGC Care Group I Journal
Vol-12 Issue-01 June 2023

RELIABLE CRC BASED ERROR CORRECTION AND DETECTION FOR FINITE FIELD MULTIPLIERS

K.Neha Nandini[1]., A.Lavanya[2]., K.Anjali[3]., K.Akshitha[4]., K.Sai Saraswathi[5]

1 Assistant Professor, Department Of ECE., Malla Reddy College Of Engineering For Women., Maisammaguda., Medchal., Ts, India (✉nehanandini.kella@gmail.com)

2,3,4,5 B.Tech ECE, (19RG1A04C4, 19RG1A04E7, 19RG1A04E8, 19RG1A04FO),

Malla Reddy College Of Engineering For Women., Maisammaguda., Medchal., Ts, India

Abstract— Applications in encryption and error-detecting codes have brought finite-field multiplication to the forefront of the academic literature. This arithmetic procedure is a difficult, expensive, and time-consuming activity that may need millions of gates for various cryptographic algorithms. In this paper, we provide a case study for the Luov cryptographic algorithm and suggest a hardware architecture based on cyclic redundancy check (CRC) for use in PQC as an error-detection system. The Luov project was entered into the NIST PQC standardization competition, where it progressed to the semifinals. The chosen CRC polynomials have sufficient error detection capabilities and are appropriate for the given field widths. To ensure the correctness of the derived formulations, we have created verification algorithms that may be used to run software implementations of the suggested schemes. The suggested error-detection techniques are verified to produce good error coverage with appropriate overhead by performing hardware implementations of the original multipliers on a Xilinx field-programmable gate array (FPGA)**.**

Index Terms— **Synonyms: finite-field multiplication, field-programmable gate array (FPGA), cyclic redundancy check (CRC), and fault detection.**

I. INTRODUCTION

1) Finite-field multiplication is a popular finite-field operation used in many current, sensitive applications and systems. Multipliers for finite fields operate by modulo, the irreducible polynomial that characterizes the finite field. Post-quantum cryptography (PQC) often requires finite-field multipliers with millions of logic gates due to the size of the inputs. As a result, research has concentrated on error elimination and obtaining better reliability with acceptable overhead [1]-[6] since it is a challenging challenge to construct such structures robust to natural and deliberate defects. In addition, there is a body of work dedicated to protecting against fault attacks and ensuring PQC's dependability. Error-detection techniques based on the number theoretic transform (NTT) were used by Sarker et al. [7] to identify both persistent and fleeting problems. Fault detection for PQC signatures based on stateless hashes was done by Mozaffari-Kermani et al. [8]. To further improve the robustness of these systems against both natural and intentional faults, error-detection hash trees for stateless hash-based signatures are suggested in [9]. For the Galois counter mode (GCM) architectures with various finite-field multipliers in GF(2128), algorithm-oblivious implementations are presented in [10] by recomputing with swapped ciphertext and extra authenticated blocks. In [11], we see a number of defenses for the NTRU encryption technique that include error-detection checksum codes and spatial/temporal redundancy.
2) While we focus on the Luov cryptographic algorithm [12], the error-detection techniques we suggest may be used with any PQC method that employs finite-field multipliers. The Luov algorithm made it to the second round of a standardization competition hosted by the National Institute of Standards and Technology (NIST) [13]. Our suggested hardware constructions use cyclic redundancy check (CRC) error-detection algorithms to guarantee efficient use of resources while providing comprehensive error checking. The following is a synopsis of our contributions to this summary.
3)
4) 1) New error-detection algorithms are presented for the finite-field multipliers GF(2m) for m > 1. These are employed in the Luov cipher. CRC-5 is the foundation of these error-detection frameworks. We also investigate and compare the complexity of primitive and standard generator polynomials for CRC-5.
5) 2) We develop novel forms of the Luov's algorithm's error-detection systems and test them through software implementations. We point out that the spectrum of possible uses

and degrees of security for this kind of derivation is rather large. However, the proposed techniques are not limited to these examples.

6) 3. The suggested error-detection architectures are integrated into the original finite-field multipliers. To ensure the schemes are overhead-aware and provide extensive error coverage, we implement them on a Xilinx Kintex Ultrascale+ FPGA, specifically the xcku5p-ffvd900-1-i device.

II. PRELIMINARIES

There are five popular PQC algorithm classes: code-based, hash-based, isogeny-based, lattice-based, and multivariate-quadratic- equation-based cryptosystems [15]. Code-based cryptography differs from others in that its security relies on the hardness of decoding in a linear error-correcting code. Hash-based cryptography creates signature algorithms based on the security of a selected cryptographic hash function. The security of isogeny-based cryptography is based on the hard problem to find an isogeny between two given supersingu- lar elliptic curves. Lattice-based cryptography is capable of creating a public-key cryptosystem based on lattices. Lastly, the security of multivariate-quadratic-equation-based cryptography depends on the difficulty of solving a system of multivariate polynomials over a finite field. Such cryptographic schemes use large field sizes to provide theneeded security levels.

Luov is a multivariate public key cryptosystem and an adaptation of the unbalanced oil and vinegar (UOV) signature scheme, but there is a restriction on the coefficients of the public key. Instead, the scheme uses two finite fields: one is the binary field of two elements, whereas the other is its extension of degree m. F_2 is the binary field and F_{2^m} is its extension of degree m. The central map $F: F_{2^m}^n \rightarrow F_{2^m}^o$ is a quadratic map, where o and v satisfy $n = o + v$, $\alpha_{i,j,k}$, $\beta_{i,k}$ and γ_k

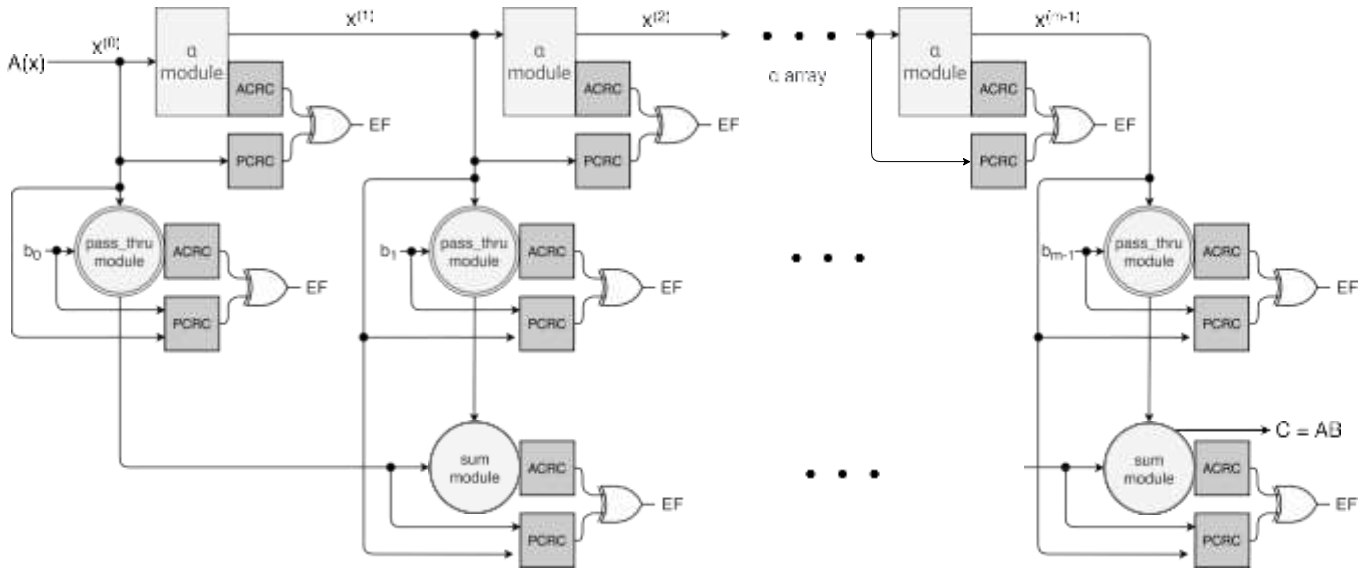

Fig. 1. Finite-field multiplier with the proposed error-detection schemes based on CRC.

are chosen from the base field F_2, and whose components $f_1, \ldots, f_o$

Luov algorithm. Thus, we derive and apply CRC signatures [17] to

are in the form $f_k(x) = \sum_{i=1}^{v} \sum_{j=i}^{n} \alpha_{i,j,k} x_i x_j + \sum_{i=1}^{n} \beta_{i,k} x_i + \gamma_k$.

the finite-field multipliers used in Luov algorithm. This would be a

These finite-field multiplications are very complex and require large-area footprint. Therefore, it is a complex task to implement such architectures resilient to natural and malicious faults. The aim of this work is to provide countermeasures against natural faults and fault injections for the finite-field multipliers used in cryptosystems such as the Luov algorithm as a case study, noting that the proposed error-detection schemes can be adapted to other applications and cryptographic algorithms whose building blocks need finite-field multiplications. Readers who are interested

in knowing more details about the Luov's cryptographic algorithm are encouraged to refer to [12].

III. PROPOSED FAULT-DETECTION ARCHITECTURES

The multiplication of any two elements A and B of $GF(2^m)$, following the approach in [16], can be presented as $A \cdot B \bmod f(x) = \sum_{i=0}^{m-1} b_i \cdot ((A\alpha^i) \bmod f(x)) = \sum_{i=0}^{m-1} b_i \cdot X^{(i)}$, where a_i the set of coefficients, $f(x)$ is the field polynomial, $X^{(i)} = \alpha X^{(i-1)} \bmod f(x)$, and $X^{(0)} = A$. To perform finite-field multiplication, three different modules are needed: *sum*, α, and *pass-thru* modules. The *sum* module adds two elements in $GF(2^m)$ using m two-input XOR gates, the α module multiplies an element of $GF(2^m)$ by α and then reduces the result modulo $f(x)$, and lastly, the *pass-thru* module multiplies a $GF(2^m)$ element by a $GF(2)$ element. One finite-field multiplication uses a total of m 1 *sum* modules, m 1 α modules, and m pass-thru modules to get the output. Fault injection can occur in any of these modules, and formulations for parity signatures in $GF(2^m)$ are derived in [16]. Parity signatures provide an error flag (EF) on each module. The major drawback of parity signatures is that their error coverage is approximately 50%, that is, if the number of faults is even, the approach would not be able to detect the faults. This highly predictable countermeasure can be circumvented by intelligent fault injection.

step forward toward detecting natural and malicious intelligent faults, especially and as discussed in this brief, considering both primitive and standardized CRCs with different fault multiplicity coverage. CRC was first proposed in 1961 and it is based on the theory of cyclic error-correcting codes. To implement CRC, a generator polynomial $g(x)$ is required. The message becomes as the dividend, the quotient is discarded, and the remainder produces the result. In CRC, a fixed number of check bits are appended to the data and these check bits are inspected when the output is received to detect any errors.

The entire finite-field multiplier with our error-detection schemes is shown in Fig. 1, where actual CRC (ACRC) and predicted CRC (PCRC) stand for ACRC signatures and PCRC signatures, respectively. In Fig. 1, only one EF is shown for clarity; however, for CRC-5, which is the case study proposed in this brief, 5 EFs are computed on each module. In Fig. 2, the α module is shown more in-depth to clarify how the proposed CRC signatures work in each finite-field multiplier.

In this work, our aim is the derivation of error-detection schemes that provide a broader and higher error coverage than parity signatures and explore the application of such schemes to the parity signatures described in [16]. For the *sum* module in CRC-1, $\hat{p}_X$ is equal to the sum of the parity bits of the input elements A and B in $GF(2^m)$, $\hat{p}_X = p_A + p_B$. Furthermore, for the *pass-thru* module in CRC-1, $p_X = b \cdot p_A$, where b is an element in $GF(2)$. For any other CRC-n scheme, instead of summing all the bits, it checks n bits at a time in the *sum* and *pass-thru* modules. For the α module, we have

$$A(x) \cdot x = a_{m-1} \cdot x^m + a_{m-2} \cdot x^{m-1} + \cdots + a_0 \cdot x \quad (1)$$

for which a set of derivations is needed to implement CRC-n into it. In Table I, the generator polynomials used to derive the CRC-5 signatures are shown. The generator polynomial $g0(x)$ is one of the standards used for radio frequency identification [18]. The other three generator polynomials $g1(x)$, $g2(x)$, and $g3(x)$ are primitive polynomials. The benefit of using a primitive polynomial as the generator that the resulting code has full total block length, which means that all 1-bit errors within that block length have separate

TABLE I

STANDARDIZED (STAND.) AND PRIMITIVE (PRIM.) GENERATOR POLYNOMIALS AND THEIR CORRESPONDING CRC SIGNATURES

$g(x)$ Utilized	Type	Predicted CRC-5 signatures	Actual CRC-5 signatures
$g_0(x) = x^5 + x^3 + 1$	Stand.	$(a_{15} + a_{12} + a_{11} + a_9 + a_8 + a_7 + a_5 + a_3)x^4 + (a_{12} + a_{11} + a_9 + a_8 + a_7 + a_6 + a_4 + a_2)x^3 + (a_{15} + a_{14} + a_{12} + a_{11} + a_{10} + a_8 + a_6 + a_1)x^2 + (a_{14} + a_{13} + a_{11} + a_{10} + a_9 + a_7 + a_5 + a_0)x + (a_{15} + a_{13} + a_{12} + a_{10} + a_9 + a_8 + a_6 + a_4)$	$(\gamma_{13} + \gamma_{12} + \gamma_{10} + \gamma_9 + \gamma_8 + \gamma_6 + \gamma_4)x^4 + (\gamma_{13} + \gamma_{12} + \gamma_{11} + \gamma_{10} + \gamma_9 + \gamma_5 + \gamma_3)x^3 + (\gamma_{15} + \gamma_{13} + \gamma_{12} + \gamma_{11} + \gamma_9 + \gamma_7 + \gamma_2)x^2 + (\gamma_{15} + \gamma_{14} + \gamma_{12} + \gamma_{11} + \gamma_{10} + \gamma_8 + \gamma_6 + \gamma_1)x + (\gamma_{14} + \gamma_{13} + \gamma_{11} + \gamma_{10} + \gamma_9 + \gamma_7 + \gamma_5 + \gamma_0)$ $(\gamma_{15} + \gamma_{14} + \gamma_{13} + \gamma_{10} + \gamma_9 + \gamma_7 + \gamma_4)x^4 + (\gamma_{15} + \gamma_{14} + \gamma_{13} + \gamma_{12} + \gamma_9 + \gamma_8 + \gamma_6 + \gamma_3)x^3$
$(x) = x^5 + x^2 + 1$	Prim.	$(a_{14} + a_{13} + a_{12} + a_9 + a_8 + a_6 + a_3)x^4 + (a_{14} + a_{13} + a_{12} + a_{11} + a_8 + a_7 + a_5 + a_2)x^3 + (a_{15} + a_{14} + a_{13} + a_{12} + a_{11} + a_{10} + a_7 + a_6 + a_4 + a_1)x^2 + (a_{14} + a_{11} + a_{10} + a_8 + a_5 + a_0)x + (a_{15} + a_{14} + a_{13} + a_{10} + a_9 + a_7 + a_4)$	$+(\gamma_{15} + \gamma_{14} + \gamma_{13} + \gamma_{12} + \gamma_{11} + \gamma_8 + \gamma_7 + \gamma_5 + \gamma_2)x^2 + (\gamma_{15} + \gamma_{12} + \gamma_{11} + \gamma_9 + \gamma_6 + \gamma_1)x + (\gamma_{15} + \gamma_{14} + \gamma_{11} + \gamma_{10} + \gamma_8 + \gamma_5 + \gamma_0)$
$g_2(x) = x^5 + x^4 + x^2 + x + 1$			$(\gamma_{15} + \gamma_{14} + \gamma_{13} + \gamma_{12} + \gamma_{10} + \gamma_9 + \gamma_6 + \gamma_5 + \gamma_4)x^4 + (\gamma_{11} + \gamma_{10} + \gamma_8 + \gamma_6 + \gamma_3)x^3 + (\gamma_{15} + \gamma_{10} + \gamma_9 + \gamma_7 + \gamma_5 + \gamma_2)x^2 + (\gamma_{13} + \gamma_{12} + \gamma_{10} + \gamma_8 + \gamma_5 + \gamma_1)x + (\gamma_{15} + \gamma_{14} + \gamma_{13} + \gamma_{11} + \gamma_{10} + \gamma_7 + \gamma_6 + \gamma_5 + \gamma_0)$
$g_3(x) = x^5 + x^4 + x^3 + x^2 + 1$			$(\gamma_{15} + \gamma_{14} + \gamma_{13} + \gamma_{12} + \gamma_{11} + \gamma_8 + \gamma_5 + \gamma_4)x^4 + (\gamma_{15} + \gamma_{10} + \gamma_8 + \gamma_7 + \gamma_5 + \gamma_3)x^3 + (\gamma_{13} + \gamma_{12} + \gamma_{11} + \gamma_9 + \gamma_8 + \gamma_7 + \gamma_6 + \gamma_5 + \gamma_2)x^2 + (\gamma_{15} + \gamma_{14} + \gamma_{13} + \gamma_{10} + \gamma_7 + \gamma_6 + \gamma_1)x + (\gamma_{15} + \gamma_{14} + \gamma_{13} + \gamma_{12} + \gamma_9 + \gamma_6 + \gamma_5 + \gamma_0)$

remainders. Moreover, since the remainder is a linear function of theblock, all 2-bit errors within that block length can be identified.

TABLE II

OVERHEADS OF THE PROPOSED ERROR-DETECTION SCHEMES FOR THE FINITE-FIELD MULTIPLIERS USED IN THE LUOV ALGORITHM DURING THE POLYNOMIAL GENERATION ON XILINX FPGA FAMILY KINTEX ULTRASCALE+ FOR DEVICE XCKU5P-FFVD900-1-I

Architecture	Area (CLBs)	Delay (ns)	Power (mW) @50 MHz	Throughput (Gbps)	Efficiency ($Gbps/CLBs$)
Luov Multiplier with primitive CRC-5 using $g_1(x)$	134 (11.67%)	4.194 (3.71%)			
Luov Multiplier with primitive CRC-5 using $g_2(x)$	131 (9.17%)	4.242 (4.90%)			
Luov Multiplier with primitive CRC-5 using $g_3(x)$	120	4.044	0.465	3.96	0.033
Luov Multiplier with standardized CRC-5 using $g_0(x)$	139 (15.83%)		0.466 (~0%)	3.81 (-3.79%)	0.027 (-18.18%)
			0.466 (~0%)	3.77 (-4.80%)	0.028 (-15.15%)
			0.466 (~0%)	3.76 (-5.05%)	0.028 (-15.15%)
	142 (18.33%)	4.499 (11.25%)	0.466 (~0%)	3.56 (-10.10%)	0.025 (-24.24%)

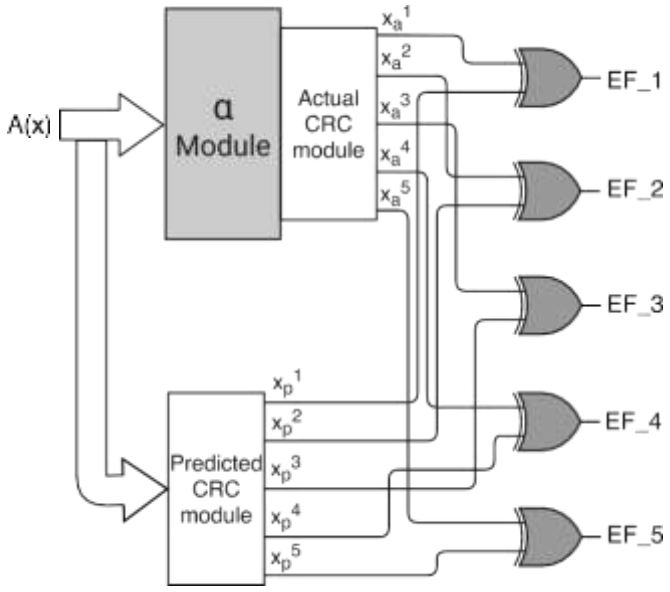

Fig. 2. Proposed error-detection constructions for α module.

with its respective one to produce five EFs, which are represented as $EF1$–$EF5$. As an example, to obtain $EF1$, x_p^1 (or $\alpha 15 + \alpha 13 + \alpha 12 + \alpha 10 + \alpha 9 + \alpha 8 + \alpha 6 + \alpha 4$ for $g0(x)$) is XORed with x^1 (or $\gamma 14$ $\gamma 13$ $\gamma 11$ $\gamma 10$ $\gamma 9$ $\gamma 7$ $\gamma 5$ $\gamma 0$ for $g0(x)$), which are calculated in (4) and (6), respectively. In our example, we use CRC-5, so the outputs are split into five categories. However, if CRC-n is used instead, the actual and anticipated outputs would be split into n categories, and there will be n EFs. The CRC signatures of the several basic polynomials are shown in Table I. We point out that the dependability needs and the overhead allowed for leeway may inform the selection of the used CRC. In other words, the size of CRC may be increased for applications where speed is paramount (and power consumption is not, as they are plugged in), such as gaming consoles. However, lower CRC is recommended for firmly embedded systems, such as implantable and wearable medical devices.

IV. ERROR COVERAGE AND FPGA IMPLEMENTATIONS

Finite-field multiplication is a costly operation and requires large footprint. We implement Luov polynomial generation to show that the proposed error-detection schemes provide high error coverage with acceptable overhead. Such implementation produces a polynomial $p(x) = a_{m-1}x^{m-1} + \cdots + a_1 x + a_0$, which requires m finite-field multiplications and m XOR operations. As pointed out before, each finite-field multiplication uses three different modules called α, *sum*, and *pass − thru* modules. A total of $m - 1$ α modules, $m - 1$ *sum* modules, and m *pass − thru* modules are needed to perform each finite-field multiplication. Moreover, a total of $m - 1$ *sum* modules are needed to perform an XOR operation. For each architecture, the error coverage is calculated as $100\ (1 - (1/2)^{sign})\%$, where *sign* denotes the number of signatures.

Luov uses the finite-field $GF(2^{16})$, or $m = 16$. Implementing its polynomials in the form of $p(x) = a_{15}x^{15} + a_{14}x^{14} + \cdots + a_1 x + a_0$ requires 14 finite-field multiplications and 15 XOR operations. Since each finite-field multiplication uses $m - 1$ α modules, $m - 1$ *sum* modules, and m *pass−thru* modules, 14×15 α modules, 14×15 *sum* modules, and 14×16 *pass − thru* modules are needed. Moreover, a total of 14 multiplications (15 α 15 sum 16 pass-thru) 15 XOR or 659 signatures are implemented. The error coverage percentage for the generation of Luov's polynomial using the finite-field $GF(2^{16})$ is $100\ (1 - (1/2)^{659})\%$. In Table II, we present the overhead of our error-detection architectures in terms of area-configurable logic blocks (CLBs), delay, power consumption (at the frequency of 50 MHz), throughput, and efficiency for the generation of poly- nomial $p(x)$, where $p(x) = a_{m-1}x^{m-1} + a_1 x + a_0$.

We utilize Xilinx FPGA family Kintex Ultrascale for device xcku5p-ffvd900-1-i, using Verilog as the hardware design entry and Vivado

as instrument used in actualizations. Adding CRC signatures to the original design results in more overhead in terms of space, latency, and power, but reduced overhead in terms of throughput and efficiency, as shown in Table II. To get the area, we read the CLBs from Vivado's place usage report. CLBs are the primary resources for creating general-purpose combinational and sequential circuits. Using Vivado's Timing Constraints Wizard, we establish a main clock period constraint of 20 ns, which is equivalent to a frequency of 50 MHz, and use this to calculate the delay. We also detail the total on-chip power, which is the power used by the FPGA itself and is calculated by summing the static power of the devices and the design power. You can calculate throughput by dividing the total amount of output bits by the delay, and you can calculate efficiency by dividing throughput by the total available space. From this data, we may conclude that efficiency losses of no more than 19% yield overhead costs that are within acceptable ranges. The error-detection architecture with the smallest area overhead, using the primitive generator polynomial g2(x), is 9.17%; however, the error-detection implementation with the smallest delay overhead, using the standardized generator polynomial for CRC-5, g0(x), is 3.71%.

To the best of our knowledge, no earlier work has been done on error-detection strategies for Luov's finite-field multipliers. Let's look at some examples to use as a basis for qualitative comparison, ensuring that the costs spent are reasonable. In their presentation of a signature-based fault diagnostic for the cryptographic block ciphers LED and HIGHT, Subramanian et al. [19] found that the two ciphers had a total overhead of 21.9% in area and 31.9% in latency. In addition, Mozaffari-Kermani et al. [6] have demonstrated Pomaranch cipher defect diagnostics, with a total overhead of 35.5% in both area and throughput. The worst-case overhead for the methods described in this summary, in terms of both area and latency, is less than 32%. The worst case scenario is in [7].

Applying NTT designs' error-detection systems results in a 24% overhead. Applying fault-detection architectures to stateless hash-based signatures results in a worst-case area overhead of more than 33% and a performance reduction of more than 14%, as shown in [8] and [9]. These and other related research in classical cryptography demonstrate that the suggested error-detection designs achieve an overhead that is within acceptable ranges when compared to previous efforts on fault detection. When compared to the original designs' inability to identify and prevent errors caused by either natural or intentional causes, these reductions in performance are tolerable.

.

V. CONCLUSION

Our work derives error-detection schemes for finite-field multipliers used in postquantum cryptographic algorithms like Luov, and the proposed error-detection schemes are generalizable to other applications and cryptographic algorithms whose building blocks require finite-field multiplications. For the purpose of verification, we have conducted software implementations of the error-detection structures described in this work, which are based on CRC-5 signatures. The complexity of both primitive and standard generator polynomials for CRC-5 has been investigated and compared. In order to attain great error coverage with appropriate cost, we have integrated the suggested error-detection techniques into the original finite-field multipliers of Luov's algorithm..

REFERENCES

Based on the work of J. L. Danger et al., "On the performance and security of multiplication in GF(2N)," Cryptography, volume 2, issue 3, pages 25-46, 2018.

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

[3] "A low-cost S-box for the advanced encryption standard using normal basis," published in Proc. IEEE Int. Conf. Electro/Inf. Technol., June 2009, pp. 52-55.

"Security analysis of logic encryption against the most effective side-channel attack: DPA," by M. Yasin, B. Mazumdar, S. S. Ali, and O. Sinanoglu, published in Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst. (DFTS), October 2015, pages 97-102.

"Efficient and reliable error detection architectures of hash-counter-hash tweakable enciphering schemes," M. Mozaffari-Kermani, R. Azarderakhsh, A. Sarker, and A. Jalali, ACM Trans. Embedded Comput. Syst., vol. 17, no. 2, pages 54:1-54:19, May 2018.

[6] "Reliable and error detection architectures of Pomaranch for false-alarm-sensitive cryptographic

applications," published in IEEE Trans. Very Large Scale Integr. (VLSI) Syst., volume 23, issue 12, pages 2804-2812, December 2015.

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

"Fault detec- tion architectures for post-quantum cryptographic stateless hash-based secure signatures benchmarked on ASIC," by M. Mozaffari-Kermani, R. Azarderakhsh, and A. Aghaie. Embedded Computing Systems, ACM Transactions on, volume 16, issue 2, pages 59:1-59:19, December 2016.

According to [9] "Reliable hash trees for post-quantum stateless cryptographic hash-based signatures," by M. Mozaffari-Kermani and R. Azarderakhsh, published in Proc. IEEE Int.

For example, in [10] M. M. Kermani and R. Azarderakhsh, "Reliable architecture-oblivious error detection schemes for secure cryptographic GCM structures," December 2019 issue of IEEE Transactions on Relations, pages 1347-1355.

"Strengthening hardware implemen- tations of NTRUEncrypt against fault analysis attacks," A. A. Kamal and A. M. Youssef, J. Cryptograph. Eng., volume 3, issue 4, "Unbalanced oil and vinegar signature schemes," by A. Kipnis, J. Patarin, and L. Goubin, published in Proc. Int. Conf. Theory Appl. Cryptograph. Techn. Berlin, Germany: Springer, 1999, pages 206-222.

According to [13] D. Moody, "Post-quantum Cryptography: NIST's Future Vision," Tech. Rep., February 2016. [Online]. Post-quantum cryptography presentation papers may be found at https://csrc.nist.gov/csrc/media/projects/pqcrypto/docs/pqcrypto-2016-presentation.pdf.

"Post-quantum cryptography: Round 2 submissions," Tech. Rep., March 2019 [14], by D. Moody. [Online]. Please refer to https://csrc.nist.gov/CSRC/media/Presentations/Round-2-of-the-NIST-PQC-Competition-What-was-NIST/images-media/pqcrypto-may2019-moody.pdf for more information.

Boston, MA, USA: Springer, 2011, pp. 949–950, doi: 10.1007/978-1-4419-5906-5_386. [15] D. J. Bernstein, "Post-quantum cryptography," in Encyclopedia of Cryptography and Security, H. C. A. van Tilborg and S. Jajodia, Eds.

[16] "Error detection in polynomial basis multipliers over binary extension fields," by A. Reyhani-Masoleh and M. A. Hasan, published in Proc. CHES, 2002, pages 515-528.

Class-1 Generation-2 Ultra High Frequency (UHF) Radio Frequency Identification (RFID) Protocol for Communications at 860 MHz 960 MHz, EPC Global, Brussels, Belgium, Version 1.0.23, 2008.

[18] The article "A tutorial on CRC computations," written by T. V. Ramabadran and S. S. Gaitonde and published in IEEE Micro, volume 8, issue 4, pages 62-75, August 1988.

"Reliable hardware architectures for cryptographic block ciphers LED and HIGHT," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 36, no. 10, pp. 1750-1758, Oct. 2017, by S. Subramanian, M. Mozaffari-Kermani, R. Azarderakhsh, and M. Nojoumian.