EasyChair Preprint

№ 10715

Comparison of Hardware Complexity of Multipliers GF(Pm)

Ivan Zholubak, Valeriy Hlukhov and Oleksandr Muliarevych

August 15, 2023

The 12th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
7-9 September, 2023, Dortmund, Germany

Comparison of hardware complexity of multipliers $GF(p^m)$

Ivan Zholubak [1], Valeriy Hlukhov [2] ,Oleksandr Muliarevych[3]
[1] Lviv Polytechnic National University, Computer Engineering Department, Stepana Bandery 12, Lviv, Ukraine 79013, E-mail: Ivan.M.Zholubak@lpnu.ua
[2] Lviv Polytechnic National University, Computer Engineering Department, Stepana Bandery 12, Lviv, Ukraine 79013, E-mail: valerii.s.hlukhov@lpnu.ua
[3]Lviv Polytechnic National University, Computer Engineering Department, Stepana Bandery 12, Lviv, Ukraine 79013, E-mail: oleksandr.v.muliarevych@lpnu.ua

***Abstract*—The article discusses the use of Galois field (GF) multipliers for cryptographic data protection based on elliptical curves. It recommends using extended Galois fields with characteristics d > 2 for digital signatures. The article proposes a criterion for determining the best field to use for data protection. It also describes methods for creating cryptoprocessors cores, including a VHDL generator for extended Galois field multipliers. However, generating field multipliers with large characteristics can be time-consuming. To improve efficiency, the article suggests simplifying the design and minimizing logic gates in an FPGA. The article determines the best fields for data protection based on the selected criterion and a certain algorithm.**

Keywords—modified guild cell; Galois fields; Boolean functions; multiplier generator; field programmable gate array

I. INTRODUCTION

Microprocessors are designed based on fundamental components that execute basic logical operations, known as Boolean functions. The Quine McCluskey technique is applied to minimize these functions [1].

Presently, computer technology is prominently directed towards advancing and utilizing cyberphysical systems (CPS), while also preparing for the advent of quantum computers. CPS, reliant on wireless technology, introduces data security challenges. Adversaries can exploit technological and algorithmic progress to breach information security, necessitating the exploration of new dependable protective methods. Digital signatures, commonly employing elliptic curves (EC) and Galois fields $GF(2^m)$ and $GF(d)$, constitute prevailing tools for information security. However, the ascent of potent quantum computers renders these encryption techniques susceptible. A prospective remedy involves leveraging EC isogenies within $GF(2^m)$ or other extended Galois fields like $GF(d^n)$. The operation units for Galois fields are employed to process codes within these fields. The paper outlines a utility for generating VHDL descriptions of Galois field multipliers, pivotal in EC-based information security. The algorithms within information security systems possess a multi-tiered structure, necessitating the adjustment of operational devices to modify the Galois field, its characteristic, or the elliptic curve. Field-programmable gate arrays (FPGAs) play a vital role in designing specialized computer functional units and encompass the subsequent significant attributes [2]:

FPGAs expedite seamless transitions to ASICs, ensuring efficient mass production.

FPGAs enable hardware implementations of algorithms and the storage of interim results within the chip during algorithm execution.

Modern FPGAs safeguard intellectual property, complicating unauthorized replication and reverse engineering.

Contemporary information security tools employ operations with extended Galois fields $GF(2^n)$ with substantial degrees denoted by n. The field elements are expressed in either polynomial or normal basis [3]. Consequently, the development of tools for conducting operations on elements within these Galois fields presents an immensely promising avenue for scientific and engineering exploration. Investigating the hardware, structural, and temporal intricacies of element multipliers within these fields is a formidable undertaking due to the need to multiply codes whose bit sizes align with the degree of the Galois field, potentially reaching up to 1000 bits. The subject of inquiry encompasses fields characterized by prime numbers (2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, and so forth, up to 997). The order of such fields approximates around 2^{998}.

II. RELATED WORK

In recent times, the utilization of elliptic curves has found its stride within the realm of cryptography [4]. This stems from the fact that elliptic curves established over finite fields give rise to finite groups, endowing them with a structurally rich framework that simplifies the determination of arithmetic operations [5]. Historically, cryptography has revolved around multiplicative groups situated on specific finite fields. Elliptic curves share similarities with these groups, yet they offer a distinctive advantage: the freedom to select an elliptic curve is more

expansive compared to the options available for finite field selection. Moreover, elliptic cryptosystems yield heightened levels of data protection. To facilitate operations involving points on elliptic curves, Galois field arithmetic is harnessed, wherein the codified elements of these fields are represented in either polynomial or normal bases [6], [7]. The multipliers essential for Galois fields exhibit intricate characteristics encompassing formidable hardware requirements [8], structural intricacies [9], and temporal intricacies [10].

III. THE GOAL OF THE WORK

The primary aim of this endeavor is to establish a theoretical foundation supporting the viability of employing extended Galois fields $GF(d^m)$ where the characteristic d exceeds 2 within the domain of data security tools. This effort seeks to formulate a yardstick for assessing and comparing data security tools grounded in such Galois fields and subsequently identifying the optimal fields based on the specified benchmark. Additionally, the article's objective encompasses the development of a tool capable of generating VHDL descriptions for multipliers operating on elements within these fields. These generated descriptions are intended for subsequent integration into the implementation of data protection mechanisms on Field-Programmable Gate Arrays (FPGAs).

The generator's functionality is designed to encompass three prevalent configurations:

The first variant entails the utilization of a Modified Guild Cell (MGC) as a consolidated entity, akin to a black box.

The second variant involves the deployment of an MGC as both a multiplier and an adder.

The third variant employs an MGC structured as a circuit comprising basic logical gates, effectively executing Boolean functions and delivering their outcomes.

IV. FEAUTURES OF THE GALOIS FIELDS ELEMENTS MULTIPLIERS CORE GENERATORS DESIGN

A significant challenge arises in the creation of multipliers for Galois field elements due to their substantial size, compounded by the presence of numerous resemblant components.

Subsequently, a possibility worth exploring involves the juxtaposition of fields with roughly equivalent orders but varying characteristics for comparative analysis. The task of manually devising such multipliers is exceedingly intricate, if not verging on impractical. Consequently, the decision was made to conceive a Galois field multiplier generator tailored to Galois fields boasting orders of approximately up to 2^{998} and showcasing diverse characteristics.

This generator was realized through the utilization of the C++ programming language. The generator's structural framework is illustrated in Figure 1.

The generator executes multiplier generation using three distinct approaches:

- Founded upon the Modified Guild Cell (MGC) concept, presented as a unified entity akin to a black box;
- Constructed around MGC as a combined multiplier and adder;
- Structured on MGC functioning as a circuit comprising basic logical gates, effectuating the minimization of Boolean function results.

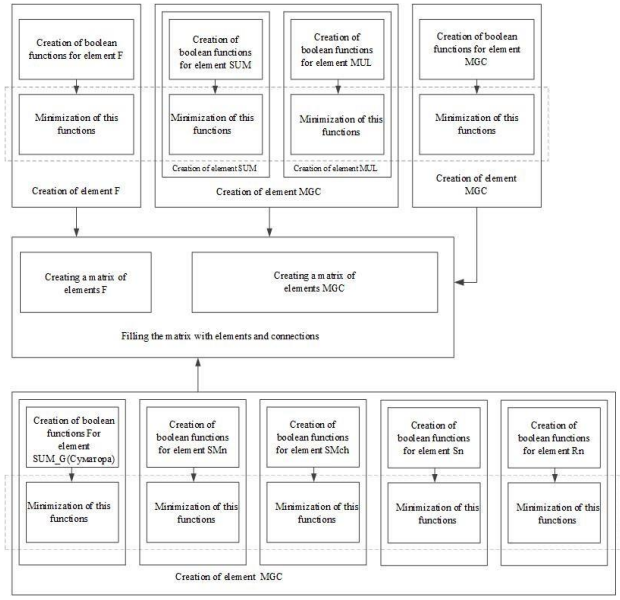

Figure 1. Structure scheme of the Galois field multiplier generator

The configuration of the multiplier, where MGC is presented as a unified entity, as well as a multiplier and adder, has been expounded in [9], while MGC's composition of basic logic gates is elaborated in [8].

The process of generating multipliers is subdivided into the following phases:

1) Development of Boolean functions for individual units.

2) Minimization of these functions.

3) Creation of distinct units:

SUM (an adder)

MUL (a multiplier)

F = (-Gm) mod d, where d signifies the field's characteristic, and Gm represents the output of MGC, SUM, or MUL.

SMn (a module facilitating modular multiplication and addition, featuring result and carry outputs)

SMch (a module executing addition or subtraction in the two's complement during division without restoring remainders)

Sn (a module determining the operation type, addition, or subtraction, during division without restoring remainders)

Rn (a module determining the necessity for an additional addition operation to ascertain the outcome, applicable solely for variants 2 and 3)

4) Creation of the MGC.

5) Establishment of interconnections between units to assemble the multiplier.

During the formulation of Boolean functions, the generator automatically generates them in accordance with truth tables. The subsequent step involves minimizing these Boolean functions using the Quine–McCluskey method.

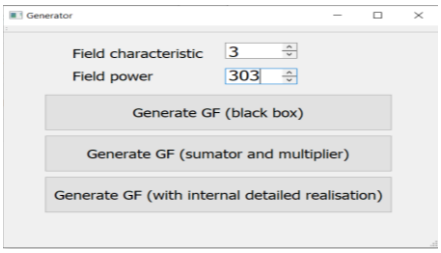

Figure 2. Galois field multiplier generator interface

Figure 2 portrays the interface of the multiplier generator, offering options to configure the field characteristic as a prime number ranging from 2 to 2^{998}, and the polynomial degree within the range of 2 to 998, while adhering to an order near 2^{998} or below. This interface features three buttons, each corresponding to a distinct generation variant. The process of generating multipliers for Galois fields can be notably time-intensive, spanning from a few minutes to several hours.

Illustrated in Figure 3 is the schematic representation of the synthesized multiplier for $GF(7^3)$ using Xilinx Vivado. This multiplier is formed from the Modified Guild Cell (MGC), encompassing both a multiplier and an adder. In Figure 4, the multiplier's diagram is presented for visual reference.

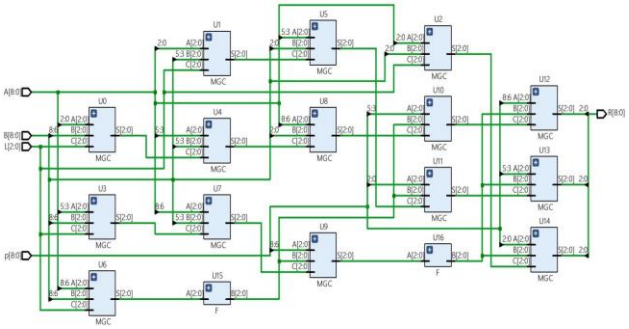

Figure 3. The scheme of the multiplier $GF(7^3)$ in the implementation of the MGC as a unified entity

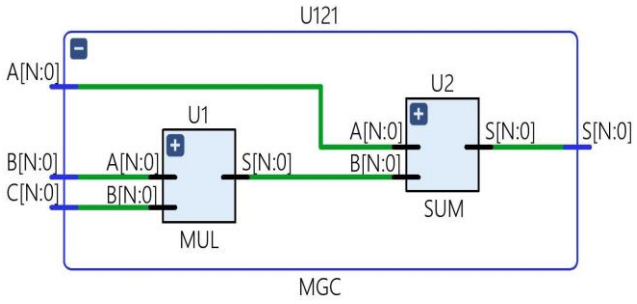

Figure 4. Scheme of the MGC in the implementation

Several multipliers were produced using the generator, and subsequent analysis was conducted on the synthesized circuits within the Xilinx Vivado environment. The forthcoming analysis will encompass the evaluation of the generated multipliers within three distinct scenarios: the MGC implemented as a single element, the MGC composed of both a multiplier and an adder, and the MGC constructed as a circuit employing elementary logical gates. The substantial quantity of MGCs involved renders manual creation of these multipliers an exceedingly daunting task. In practical terms, millions of MGCs are employed within multipliers.

V. Using the Template

Tables I, II and III provide a comprehensive depiction of both actual and theoretical outcomes in the generation of VHDL-descriptions for multipliers, intended for the FPGA Virtex UltraScale+ XCVU9P, which incorporates a substantial 2,069,000 Look-Up Tables (LUTs). Notably, in the case of implementing multipliers with the MGC functioning as an integral unit, the hardware costs experience rapid escalation as the field order increases. Among various Galois fields, the costs are at their lowest for binary and ternary fields.

Similarly, when employing the MGC composed of a multiplier and an adder, the hardware complexity exhibit a swift upsurge with the expansion of the field order. For this architecture, the most economical hardware costs materialize for fields with characteristics d = 2, 3, 5, and 7.

In scenarios where MGC multipliers are designed with the architecture featuring MGC comprised of elementary logical gates, hardware costs exhibit an upward trajectory in tandem with the augmentation of the field order. As a result, the three specified architectures offer the flexibility to construct multipliers catering to varying requirements.

The tables also presents the duration required for multiplier generation. It's observed that the time taken increases in direct proportion to the field characteristic. All measurements were conducted on a computer system with the following parameters:

CPU: Intel(R) Core i7-4770 CPU
Frequency: 3.40 GHz
Memory: 32 GB
Operating System: Windows 11, 64 bit.

TABLE I. Hardware complexity of multipliers in terms of LUT count NR_D, NT_D and generation duration for Galois field multipliers on FPGA Virtex UltraScale+ (MGC implemented as a unified entity)

The field for which FPGA multiplier is built	d	Galois field order (approximately, Od)	MGC implemented as a unified entity		
			LUT amount, NRd	Time, sec.	LUT amount, NTd
$GF(2^{50})$	2	1,1E+15	2504	1,0	4901
$GF(3^{32})$	3	1,9E+15	4034	1,4	3970
$GF(5^{22})$	5	2,4E+15	19936	1,6	41625

GF(7^{18})	7	1,6E+15	16851	3,0	27585
GF(13^{14})	13	3,9E+15	32134	8,0	185420

TABLE II. HARDWARE COMPLEXITY OF MULTIPLIERS IN TERMS OF LUT COUNT NR_D, NT_D AND GENERATION DURATION FOR GALOIS FIELD MULTIPLIERS ON FPGA VIRTEX ULTRASCALE+ (MGC COMPOSED OF A MULTIPLIER AND AN ADDER)

The field for which FPGA multiplier is built	d	Galois field order (approximately, Od)	MGC composed of a multiplier and an adder		
			LUT amount, NRd	Time, sec.	LUT amount, NTd
GF(2^{50})	2	1,1E+15	2190	0,5	2450
GF(3^{32})	3	1,9E+15	4032	0,7	1984
GF(5^{22})	5	2,4E+15	5615	0,8	2768
GF(7^{18})	7	1,6E+15	3522	1,2	1837
GF(13^{14})	13	3,9E+15	10211	2,0	10216

TABLE III. HARDWARE COMPLEXITY OF MULTIPLIERS IN TERMS OF LUT COUNT NR_D, NT_D AND GENERATION DURATION FOR GALOIS FIELD MULTIPLIERS ON FPGA VIRTEX ULTRASCALE+ (MGC COMPOSED OF SIMPLE LOGIC GATES)

The field for which FPGA multiplier is built	d	Galois field order (approximately, O_d)	MGC composed of simple logic gates		
			LUT amount, NR_d	Time, sec.	LUT amount, NT_d
GF(2^{50})	2	1,1E+15	18784	0,5	29208
GF(3^{32})	3	1,9E+15	17950	0,5	36510
GF(5^{22})	5	2,4E+15	17867	0,5	35049
GF(7^{18})	7	1,6E+15	16689	0,5	23366
GF(13^{14})	13	3,9E+15	15369	0,5	23658

Tables IV, V, and VI present a comprehensive comparison between the theoretical KTmul and the actual KRmul hardware costs of multipliers, juxtaposed against their relationship with the KT2 and KR2 costs of multipliers for binary fields. This analysis encompasses three distinct variants of MGC implementation. Specifically, KTmul is defined as NTd/NT2, and KRmul is expressed as NRd/NR2.

The data highlighted in Tables IV, V, and VI, and graphs Fig. 5, 6, 7 underscore that, when the MGC is implemented as an integral entity, ternary Galois fields outperform binary ones by approximately 3%. However, when generating a multiplier utilizing MGC consisting of both a multiplier and an adder, a discernible pattern emerges. In comparison to binary fields, a field with characteristic 3 exhibits an 11% larger cost index, while a field with characteristic 5 demonstrates a 20% larger cost index, and a field with characteristic 7 reflects an 18% larger cost index.

TABLE IV. COMPARISON OF THEORETICAL AND ACTUAL HARDWARE COSTS OF MULTIPLIERS ON FOR FPGA VIRTEX ULTRASCALE+ (MGC IMPLEMENTED AS A UNIFIED ENTITY)

Galois field	d	GF order (O_d)	$C_o = \frac{O_d}{O_2}$	MGC implemented as a unified entity			
				KT_{mul}	$\frac{KT_{mul}}{C_o}$	KR_{mul}	$\frac{KR_{mul}}{C_o}$
GF(2^{50})	2	1,1E+15	1	1	1	1	1
GF(3^{32})	3	1,9E+15	1,65	0,81	0,43	1,61	0,97
GF(5^{22})	5	2,4E+15	2,12	8,49	4	7,96	3,75
GF(7^{18})	7	1,6E+15	1,35	5,63	4,17	6,72	4,97
GF(13^{14})	13	3,9E+15	3,5	37,8	10,8	12,83	3,66

TABLE V. COMPARISON OF THEORETICAL AND ACTUAL HARDWARE COSTS OF MULTIPLIERS ON FOR FPGA VIRTEX ULTRASCALE+ (MGC COMPOSED OF A MULTIPLIER AND AN ADDER)

Galois field	d	GF order (O_d)	$C_o = \frac{O_d}{O_2}$	MGC composed of a multiplier and an adder			
				KT_{mul}	$\frac{KT_{mul}}{C_o}$	KR_{mul}	$\frac{KR_{mul}}{C_o}$
GF(2^{50})	2	1,1E+15	1	1	1	1	1
GF(3^{32})	3	1,8E+15	1,65	0,81	0,49	1,84	1,11
GF(5^{22})	5	2,4E+15	2,12	1,13	0,53	2,56	1,2
GF(7^{18})	7	1,6E+15	1,35	0,75	0,55	1,6	1,18
GF(13^{14})	13	3,9E+15	3,5	4,17	1,19	4,65	1,32

TABLE VI. COMPARISON OF THEORETICAL AND ACTUAL HARDWARE COSTS OF MULTIPLIERS ON FOR FPGA VIRTEX ULTRASCALE+ (MGC COMPOSED OF SIMPLE LOGIC GATES)

Galois field	d	GF order (O_d)	$C_o = \frac{O_d}{O_2}$	MGC composed of simple logic gates			
				KT_{mul}	$\frac{KT_{mul}}{C_o}$	KR_{mul}	$\frac{KR_{mul}}{C_o}$
GF(2^{50})	2	1,1E+15	1	1	1	1	1
GF(3^{32})	3	1,9E+15	1,65	1,25	0.75	0,95	0,57
GF(5^{22})	5	2,4E+15	2,12	1,2	0,56	0.95	0,44
GF(7^{18})	7	1,6E+15	1,35	0,8	0,59	0,88	0,65
GF(13^{14})	13	3,9E+15	3,5	0,81	0,23	0.81	0,23

The Fig 5, 6, 7 represents the change in hardware complexity of multipliers for Galois fields with different field characteristics and different order of the polynom.

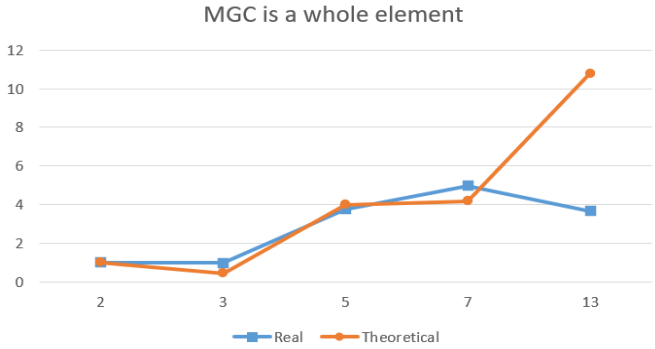

Figure 5. Comparison of theoretical and real hardware complexity coefficient when MGC is a whole element

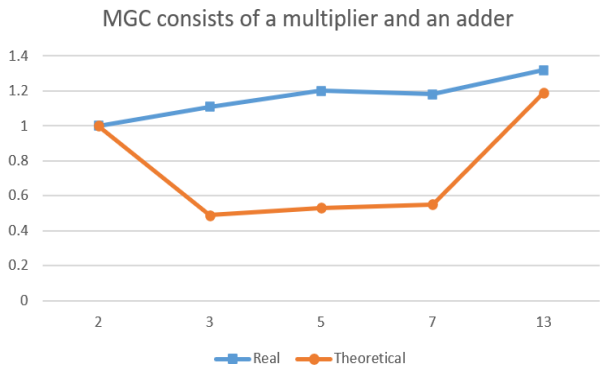

Figure 6. Comparison of theoretical and real hardware complexity coefficient when MGC consists of a multiplier and an adder

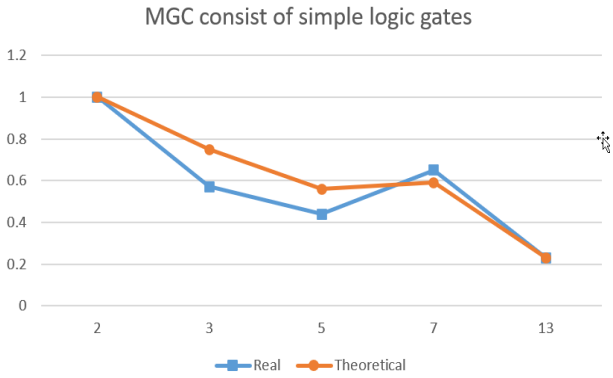

Figure 7. Comparison of theoretical and real hardware complexity coefficient when MGC consists of simple logic gates

VI. Conclusions

The paper provides a comprehensive consolidation of the theoretical groundwork underpinning the development of Galois field multipliers on FPGAs. It delineates three distinct variants for the construction of Galois field multipliers, and subsequently conducts an insightful comparison among these alternatives. A bespoke tool has been engineered to facilitate the generation of VHDL-descriptions for multipliers handling elements within such fields. These generated descriptions are intended for subsequent integration into the implementation of data protection mechanisms on FPGAs.

The criterion employed for the comparative analysis of Galois field multipliers is hardware complexity. The study reveals that employing extended Galois fields $GF(d^m)$ with characteristics $d > 2$ is a judicious choice for data protection tools. Notably, the specific complexity of fields characterized by $d = 3$ outperforms fields characterized by $d = 2$ by 3% when the MGC is implemented as a unified entity.

When the MGC is structured as both a multiplier and an adder, fields characterized by $d = 3$ exhibit an 11% higher hardware complexity index. Fields with $d = 5$ and $d = 7$, on the other hand, exhibit 20% and 18% higher indices of hardware complexity respectively, as compared to binary fields.

The article outlines the architecture and outcomes of the Galois field multiplier core generator, accommodating various orders up to 9.49e+300 elements. This groundbreaking generator enables the creation of VHDL-descriptions for multipliers that would be prohibitively intricate to devise manually. The generator formulates multiplier descriptions centered around the MGC composition, alongside proposing three diverse MGC construction approaches. Employing MGC as a complete unit, a multiplier and adder, holds advantages for fields with characteristics $d = 2$, 3, 5, and 7. However, MGC as a matrix multiplier and adder shines when dealing with multipliers featuring substantial field characteristics.

The generation process for VHDL-descriptions can extend up to 889 seconds. Moreover, plans are underway for the creation of pipeline cores.

References

[1] Quine, W. Van Orman, *"The Problem of Simplifying Truth Functions"*, The American Mathematical Monthly, Vol.59, No.8, pp. *521–531*, 1952. *DOI:*10.2307/2308219. JSTOR 2308219

[2] S. Ellison, M. L. Boppana, “Bit-parallel systolic multiplier over GF(2^m) for irreducible trinomials with ASIC and FPGA implementations”, Department of Electronics and Communicating Engineering, National Institute of Technology, Varangal, Telangana 506004, India, IET Journal IET Circuits Devices Syst., Vol. 12, Iss. 4, pp. 315-325, 2018. DOI:10.1049/iet-cds.2017.0426

[3] H. El-Razouk, "Input-Latency Free Versatile Bit-Serial GF(2^m) Polynomial Basis Multiplication," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 30, no. 5, pp. 589-602, May 2022, doi: 10.1109/TVLSI.2022.3155611.

[4] Q. Yang, "Key Technologies in Computer Algorithm Dynamics Based on Front-end and Front-end Separation System," 2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballar, India, 2023, pp. 1-6, doi: 10.1109/ICDCECE57866.2023.10150518.

[5] D. Shah, T. Shah, M. M. Hazzazi, M. I. Haider, A. Aljaedi and I. Hussain, "An Efficient Audio Encryption Scheme Based on Finite Fields," in IEEE Access, vol. 9, pp. 144385-144394, 2021, doi: 10.1109/ACCESS.2021.3119515.

[6] K. Javeed, A. El-Moursy and D. Gregg, "EC-Crypto: Highly Efficient Area-Delay Optimized Elliptic Curve Cryptography Processor," in IEEE Access, vol. 11, pp. 56649-56662, 2023, doi: 10.1109/ACCESS.2023.3282781.

[7] J. Dong, P. Zhang, K. Sun, F. Xiao, F. Zheng and J. Lin, "EG-Four$\mathbb {Q}$: An Embedded GPU-Based Efficient ECC Cryptography Accelerator for Edge Computing," in IEEE Transactions on Industrial Informatics, vol. 19, no. 6, pp. 7291-7300, June 2023, doi: 10.1109/TII.2022.3205355.

[8] L. Wu, Y. Hu, K. Zhang, W. Li, X. Xu and W. Chang, "FLAM-PUF: A Response–Feedback-Based Lightweight Anti-Machine-Learning-Attack PUF," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 41, no. 11, pp. 4433-4444, Nov. 2022, doi: 10.1109/TCAD.2022.3197696.

[9] J. -S. Pan, C. -Y. Lee, A. Sghaier, M. Zeghid and J. Xie, "Novel Systolization of Subquadratic Space Complexity Multipliers Based on Toeplitz Matrix–Vector Product Approach," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 27, no. 7, pp. 1614-1622, July 2019, doi: 10.1109/TVLSI.2019.2903289.

[10] M. Rahma, I. Zholubak, V. Hlukhov, “Devices for multiplicative inverse calculation in the binary Galois fields”, Proceedings of 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies, DESSERT 2018, pp. 261–264, 2018. DOI: 10.1109/DESERT.2018.8409141