International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V73I5P102

Volume 73 Issue 5, 9-15, May 2025
© 2025 Seventh Sense Research Group®

Original Article

Reconfigurable Architecture for Elliptic Curve Cryptography using Runtime Reconfiguration

Pravin Zode[1], Pradnya Zode[2], Pankaj Joshi[3], Sudhanshu Maurya[4], Nilesh Shelke[5]

[1,2]*Department of Electronics Engineering, Yeshwantrao Chavan College of Engineering, Nagpur, India.*
[3]*Department of Electronics Engineering, Ramdeobaba College of Engineering and Management, Nagpur, India.*
[4,5]*Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International Deemed University, Pune, India.*

[1]*Corresponding Author : ppzode@ycce.edu*

Received: 30 July 2024 Revised: 22 January 2025 Accepted: 30 April 2025 Published: 31 May 2025

Abstract *- The **e**lliptic curve is mainly used in cryptographic applications for shorter keys. This helps to increase the security level without decreasing the security level. ECC security completely depends on associated domain parameters. Day by day, the security requirements are changing, and hence, the domain parameters are also changing. Interoperability is the main issue in such changing standards. This paper presents new runtime reconfigurable hardware for such changing requirements. We have used the dynamic partial reconfiguration technique for the runtime reconfiguration of domain parameters. By changing the domain parameters, different security levels are achieved. This also helps avoid side-channel attacks and provides rigid security using programmable hardware. The proposed architecture is implemented on the Xilinx ML605 development board. The experimental results show that the area delay product of the implementation is the same and hence can be used for resource constraint environments where security is important considering other tradeoff parameters.*

Keywords *- Partial reconfiguration, Runtime reconfiguration, Montgomery multiplication, Elliptic curve cryptography, Public key cryptography.*

1. Introduction

The pervasive use of computing and communication networks created a need for a secure and reliable cryptosystem. Elliptic Curve Cryptography (ECC) was introduced independently by cryptographers Victor Miller [1] and Neal Koblitz [2] in 1985. It is public-key cryptography based on group operations in the finite field and the difficulty of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP). The main advantage of ECC is the short keys used compared to traditional public-key cryptosystems, which do not decrease the security level. This significantly saves area, time, and computational complexity [3]. ECC is particularly useful in applications where memory, bandwidth and computational power are limited [4]. The main advantage of a reconfigurable and scalable design lies in its capacity to adjust the key size dynamically, allowing for the negotiation of ECC domain parameters during runtime and the reconfiguration of functional blocks to accommodate these changes. Interoperability with security is the main issue in hardware implementations. The best example is an IPSec security protocol, where security protocol configured runtime [5]. Software implementation provides interoperability, but more computational time is required. In hardware, this can be accelerated as ECC standards move towards key sizes of 163, 233, 283,409, and 571 bits. Thus, it is necessary to support these large bits efficiently. Also, the cryptosystem should support a variety of devices and security needs. The drawback of hardware implementation is that once implementations' parameters are defined, they cannot be changed runtime as per security requirements. Upgrading hardware often faces certain limitations and challenges due to the inherent characteristics of physical components, compatibility issues, and practical constraints. Upgrading hardware may require more power, which the existing power supply may not be able to provide. More powerful hardware components can generate increased heat. If the system lacks adequate cooling solutions, an upgrade may cause overheating issues, leading to performance throttling or even hardware damage. Reconfigurability features in Field Programmable Gate Arrays (FPGA) combine the advantages of software flexibility and hardware, which makes it possible to accelerate hardware performance. The FPGA could be used to adapt new cryptographic algorithms or protocols' primitive requirements. In a side-channel attack, the physical implementation of the systems leaks information like power consumption, execution time, and electromagnetic field. By observing and processing this information, the adversary reveals the secret from implementation [6]. Also, it could be possible that while designing the hardware, some bugs in the system or changing the security requirement for a complete life cycle cannot be predicted. Different choices of


This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

tuples and recommendations from different standards lead to customizable processor architecture, which will support multiple tuples. Furthermore, the randomization of side-channel information is required to hide secret information. In this paper, the customizable implementation of ECC hardware architecture is proposed. The proposed architecture updates different ECC parameters (tuples), and functionality is changed at runtime using the capabilities of the FPGA to countermeasure any fault attack and avoid side-channel information leakage.

The rest of the paper is organized as follows: Section -2 describes related work. Section 3 overviews the basics of elliptic curve operations. Partial reconfiguration and its benefits are discussed in Section 4. The proposed architecture is detailed in Section 5. Implementation details are discussed in Section 6. Section 7 contains results and discussions, conclusions and future work in Section 8.

2. Related Work

In the last few years, sophisticated techniques have been developed, and few measurements are required to attack cryptosystems. The main aim of a side-channel attack is to obtain a signature from known traces of noise generated during computation and correlate it with unknown information. ECC implementation imposes several challenges in performance, security, and flexibility [7, 8]. Customizable hardware designs are adapted or tailored to meet specific requirements and constraints. In literature, many customizable architectures based on the curve checker, time redundancy [9], and hardware redundancy [10] schemes were proposed for interoperability and adapting security requirements without considering the side-channel attacks. Most of the works [11-13] are custom implementing specific tuples. High-performance and reconfigurable architecture are presented in [14, 15].

In [16], scalable architecture for pseudorandom and binary curves supports different key sizes without reconfiguring the hardware with low latency. Whereas information about the side channel is not considered. Also, they incur very high area and performance penalties, and none of these countermeasures provide complete security. The work proposed in [17] is based on customizable architecture for elliptic curve cryptography. In this work, the author proposed changing the serial multiplier design to get different levels of security. However, some published papers provide some degree of freedom in selecting the ECC parameters.

Several tuples are used in ECC, and two parties must agree to the same tuples to interpolate. In this work, we aimed at run time reconfigurable architecture which will adapt to changing requirements of security needs by changing the tuples used in computation without changing the device. The cryptosystem can be upgraded and made to be attack-resistant and efficient. Customizable ECC architectures are particularly valuable in embedded systems, IoT devices, and other applications where resource constraints, power efficiency, and security are critical concerns.

3. Overview of ECC

The concept of ECC relies on a unique addition function and blends principles from number theory and algebraic geometry. These curves can be constructed over various number fields. The elliptic curve is characterized by the collection of solutions to the equation ($y^2 + xy = x^3 + ax^2 + b$), and its characteristics, such as shape and properties, are determined by the specific values of a and b. Even slight alterations in these parameters can lead to significant variations in the set of (x, y) solutions. ECCs are computationally complex due to scalar multiplication with higher-order bit operations [16]. Figure 1 shows the geometrical representation of the Elliptic curve. ECC is often defined by tuple $T = (GF(q), a, b, G, n, h)$. Where $GF(q)$ is the finite field over which the curve is defined. The coordinates 'x' and 'y' of the points on the curve are elements of this finite field.

The values of 'a' and 'b' determine the shape and properties of the curve on $GF(q)$. G is the base point, also known as the generator point. It is a specific point on the elliptic curve, and the discrete logarithm of this point is with respect to itself. The number of distinct points on the curve is defined as n, which can be generated by repeatedly adding G to itself such that $nG = 0$. h is a cofactor, representing the number of distinct points on the curve divided by the order 'n'. It is essential to have a large prime order 'n' for security. The ECC involves the selection of an elliptic curve, E, and a point on that curve, P, as system parameters for secure key exchange and message transfer between parties. To encrypt a message, Person (X), who wishes to encrypt the message, will randomly generate an integer and multiply the integer 'd' by the point P to create a public key (Q =d*P). Person X releases Q as the public key and securely stores the private key 'd' as the secret key. Person (Y) who wishes to secretly send a message (M) to an individual (X) is to randomly generate an integer (k) as their private key and multiply the integer (k) by the point (P) to create (A = k*P).

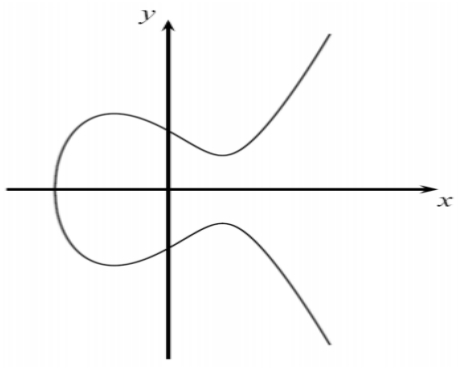

Fig. 1 Elliptic curve

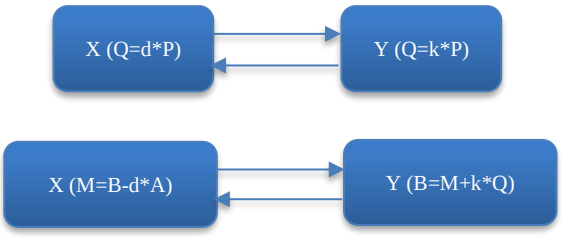

Fig. 2 Key exchange and message transfer

Then, person Y generates (B = M + k * Q) using the Q public key and the M message and sends the final encrypted message (A, B) to X. Person X, who receives the Coded Message (A, B), calculates d*A using secret key d and performs an arithmetic operation represented by (M = B - d *A) to return the message M. Figure 2 shows the key exchange and message transfer using elliptic curve cryptography. By combining the EC-DH key exchange and the elliptic curve-based encryption/decryption, ECC provides secure and efficient communication between parties in various cryptographic applications. It is widely used in modern systems where resource constraints and strong security are both essential.

Numerous sets of tuples are suggested by international standards such as ANSI, ISO, and IEEE, resulting in challenges with interoperability. Various field operations required at the bottom level are inversion, squaring, and adding, depending upon the field used. Optimization is essential at a lower level as it affects the performance of cryptosystems. The most popular implementation is projective, as point addition and field inversion are free as this is very expensive in hardware.

4. Partial Reconfiguration

Partial Reconfiguration (PR) refers to the capability to modify a specific part of the device without causing disruptions to its regular operation. This functionality enhances system flexibility by accommodating additional functions, adjusting size, and reducing costs through multiplexing in the hardware, especially in scenarios where applications must manage a range of functions.

Additionally, it contributes to power reduction by deactivating power-intensive tasks when they are not needed [19]. Reconfigurable logic includes Slice logic, memory blocks and DSP blocks, and static logic includes clock modifying blocks, buffers, I/O components and device feature blocks.

This feature is available in Xilinx Virtex- 4/5/6 family devices and supported by Xilinx ISE. In most cases, user design is structured into segments housing both noncritical and critical technology blocks. The non-critical segment is present within both the static and critical sections in the partial reconfiguration sections.

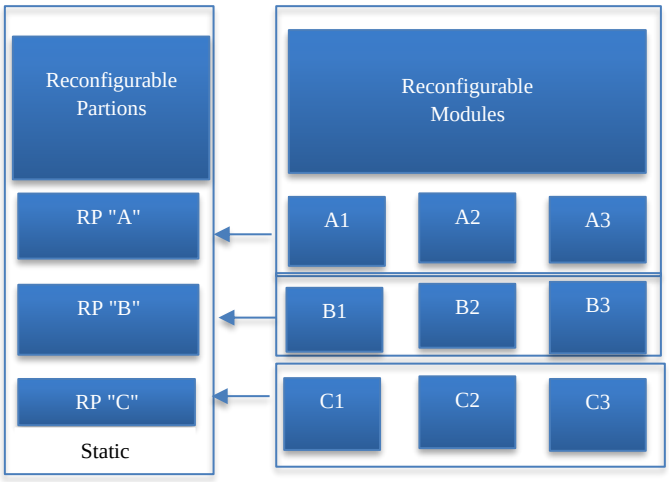

Fig. 3 Partial reconfiguration methodology

This feature reduces fault tolerance, enhances flexibility and reduces power consumption. Partial reconfiguration of an FPGA can be accomplished through two methods: static and dynamic. In dynamic partial reconfiguration, a specific area of the FPGA can be reconfigured while it is actively running.

In contrast, in the static method, the device remains inactive during the reconfiguration process. Figure 3 shows the partial reconfiguration methodology with a static portion with three reconfigurable partitions with reconfigurable modules.

5. Scalar Multiplication

The most crucial and computationally intensive task within the elliptic curve cryptosystem is scalar multiplication. It is used to derive a new point on an elliptic curve from an initial point by adding the initial point to itself multiple times. This process involves a sequence of point additions and point doublings. The efficiency and performance of the system depend on the algorithm, coordinates system and underlying field [20, 21]. Montgomery Powering Ladder behaves regularly, and this inherent property makes it naturally attack resistance [22-25]. Computations are carried out in x-coordinate only, and a lot of multiplications are saved. Less memory is required as the y-coordinate is stored and needs not to be handled during computation but computed at the end.

Algorithm 1 shows the Montgomery powering ladder, and Figure 4 shows the control flow in the algorithm. Point addition and doubling are performed in each iteration. The algorithm maintains that the difference between points P1 and P2 is always P. These point additions and doubling operations are performed using various finite field additions, inversion, and multiplication. Addition is performed using bitwise XORing. Inversion is performed using the Itoh-Tsujii Algorithm [26, 27] and multiplication using Montgomery multiplier [28] and Interleaved multiplier as two configurations.

Algorithm 1: Montgomery Powering Ladder Algorithm

Input : k= $(k_{t-1},.....k_1,k_0)_2$ with k_{t-1} =1 P(x,y) ϵ GF(2)

Output : Point on the curve Q = kP (x_3,y_3)

Begin

1. $P_1 \leftarrow P$; $P_2 \leftarrow 2P$
2. for i = m-2 to 0 do
3. If k_i =1 then
   a. $P_1 \leftarrow P_1+P_2$
   b. $P_2 \leftarrow 2P_2$
4. Else
   a. $P_2 \leftarrow P_1+P_2$
   b. $P_1 \leftarrow 2P_1$
5. end if
6. end for
7. end & return (Q = P_1)

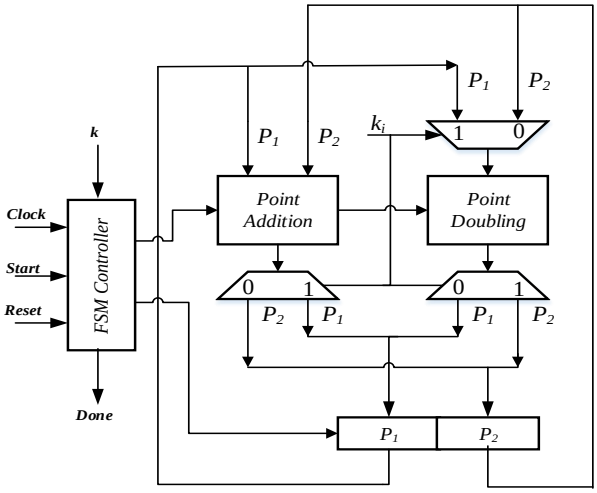

Fig. 4 Control flow of montgomery powering ladder

6. Proposed Architecture

Although ECC offers numerous benefits, it is susceptible to different types of side-channel attacks, such as power attacks, timing attacks, and fault injection attacks. These side-channel attacks exploit information obtained from the physical execution of a cryptographic system. While during the execution of scalar multiplication, the power traces for point addition and doubling are different.

By analyzing the power traces, the attacker tries to correlate variations in power consumption with specific intermediate values or operations of the ECC algorithm. The number of point additions or doubling operations might be distinguishable during point multiplication based on their power consumption patterns. Methods like Simple Power Analysis (SPA), Differential Power Analysis (DPA), Refined Power Analysis (RPA), and Zero-value Point Attack (ZPA) are commonly used techniques for extracting sensitive information through analysis. To defend against power analysis attacks, various countermeasures can be employed. The introduction of randomness in the computation facilitates the introduction of noise in implementation; incorporating constant-time algorithms and masking the secret value can provide protection against power analysis attacks. Implementing these countermeasures helps make it more difficult for attackers to extract sensitive information from the power consumption of the cryptographic device during ECC operations.

We proposed a reconfigurable hardware architecture that adapts its operation dynamically to operate with a different set of parameters (tuples) and reconfigurable multiplier modules. Figure 5 shows the framework for implementation. Self-reconfigurable FPGA is used for dynamic reconfiguration. The control unit is designed and modelled using a Finite State Machine (FSM). Xilinx Microblaze 32-bit soft processor is used to control and manage the reconfiguration process. Internal Block RAM (BRAM) is attached through a Local Memory Bus (LMB) and memory controllers.

The ICAP (Internal Configuration Access Port) port allows MicroBlaze to access and control the internal configuration memory of the FPGA. It can decide when and which configuration bitstreams to load based on system requirements.

This is essential for implementing dynamic reconfiguration, where portions of the FPGA can be reprogrammed on the fly without interrupting the operation of the rest of the FPGA. Microblaze acts as a supervisor to control the reconfiguration process. Appropriate partial bits stored in the compact flash card incorporate randomization into the system, making power analysis attacks more difficult.

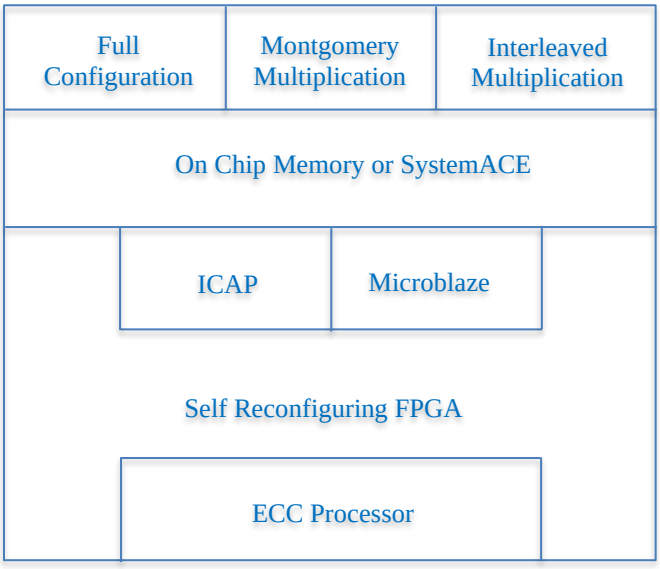

Fig. 5 Framework for ECC implementation

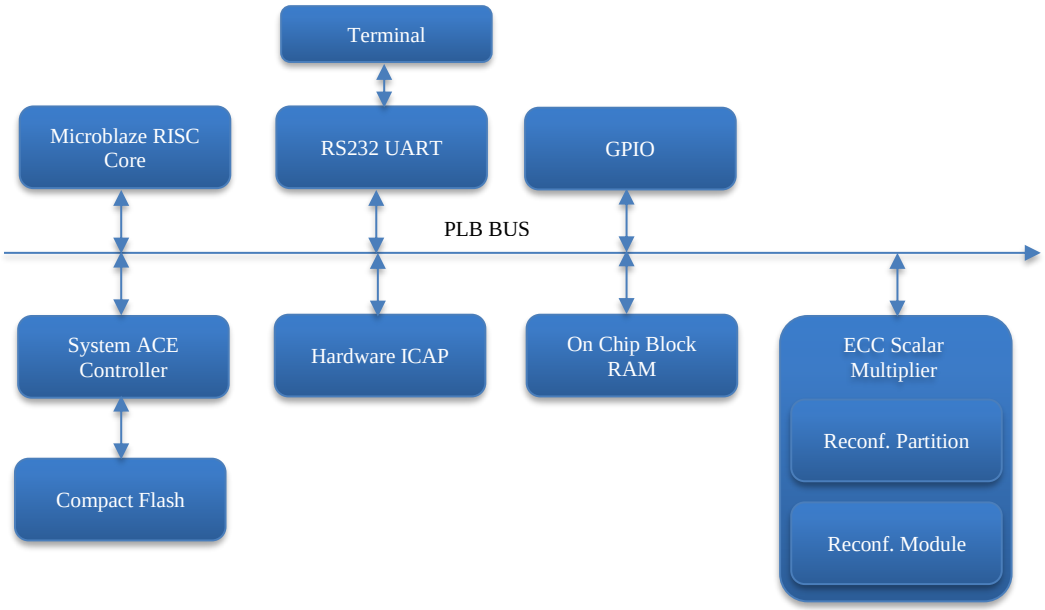

Fig. 6 Reconfigurable architecture with reconfigurable module

Figure 6 shows the architectural details of the proposed reconfigurable architecture. The reconfigurable processor is defined as the Reconfigurable Partition (RP) and Allocates Different Tuples and Multiplier Modules as (RM). Hardware elements, Addition, Multiplication and subtraction, and inversion are used for implementation. Multiplier modules are used for dynamic reconfiguration as these modules are compute-intensive and have almost the same performance. Partial bitstreams and System Advanced Configuration Environment (ACE) files are stored in compact flash for implementation on FPGA.

7. Results and Discussions

Security and efficiency should be the primary goals in the ECC implementation of FPGA, especially when dealing with cryptographic operations. For FPGA implementation, secure Koblitz curve (y2 + xy = x3 + ax2 + b) over GF (2) and extension field GF (163) with irreducible polynomial (f(z) = z163 + z7 + z6 + z3 + 1) is considered [18]. Xilinx Platform Studio is used to create a processor system. Top-level design defines two reconfigurable partitions with two reconfigurable modules, Montgomery and Interleaved multiplier.

The proposed runtime reconfigurable architecture is modeled using Verilog HDL and simulated AMD Vivado Design suite and synthesized with the default setting. Static implementation is reused for the rest of the configuration. Plan ahead Tool is used for reconfigurable portioning, floorplan the design, adding reconfigurable modules, and running implementation to generate the full and partial bitstream. Full bitstream is used for the initial configuration. Partial bitstream is loaded by using ICAP on demand. A control unit for Elliptic Curve Cryptography (ECC) is an essential component in cryptographic hardware or software that manages and orchestrates the operations required for ECC. It controls the execution of ECC algorithms, coordinates key generation, performs point multiplication, and ensures secure and efficient cryptographic processes. The finite number of point additions and doubling states transitions between states and logic to control the operation of scalar multiplication is considered for design. The proposed scalable architecture is implemented on the Virtex-6 ML605 Embedded Development Board. It is evaluated in terms of functionalities, resource utilization, power consumption, and latency for this partial reconfiguration. Initial configuration was done using the Compact Flash Memory card, and then for dynamic reconfiguration, AXI HWICAP peripheral with software control was used, and functionality was verified on Hyper Terminal. The Architecture is tested using testbench using standard test vectors, post route simulation models are generated for calculation of delay, area and power analysis. The comparative results of the proposed architecture implementation are reported in Table 1. The proposed architecture provides runtime reconfiguration with two multipliers with different delays and allows quick curve modifications by changing the system's tuples. The effectiveness of countermeasures used in the cryptosystem is time-limited; hence, the configuration will be changed under attack-prone conditions.

Table 1. Comparative results of implementation

Parameters	Using Interleaved Multiplier	Using Montgomery Multiplication
LUT (150720)	2379	2662
Delay	11.2ms	10.231ms
Area-Delay-Product	26644	27232
Overhead	2.15%	

The architecture supports 163 and 233 bits, satisfies the demand for higher security and is compatible with various key sizes. Table 1 shows the comparative implementation results. Analysis of results shows that small changes in performance parameters can be used to mislead the adversary from gaining the secret key from the implementation. The dynamic reconfiguration method improves the reconfiguration time and area efficiency. Comparing hardware implementations with other reported results directly is not possible due to the utilization of various FPGA technologies, different synthesis settings, and architectural differences. Fault occurrences are assumed to be random and may be introduced intentionally for the purpose of recovering data. The proposed architecture has been tested using the NIST-recommended standard test vectors.

8. Conclusion

In this paper, reconfigurable architecture for elliptic curve cryptography is presented. We used two approaches to implementation. The first approach to modifying the tuples is to increase the security requirements, and the second is to secure the hardware against fault attacks. Our results indicate that Area Delay-Product is almost the same in both multiplier implementations. The proposed architecture is suitable for resource constraint and attack-prone implementations. The architecture also supports different tuples for Koblitz curves, as recommended by NIST. On-going and future work includes functional extensions and optimizations such as improving speed, minimizing resource utilization, and enabling runtime reconfiguration for customized designs.

References

[1] Neal Koblitz, "Elliptic Curve Cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203-209, 1987. [CrossRef] [Google Scholar] [Publisher Link]

[2] Victor S. Miller, "Use of Elliptic Curves in Cryptography," *Advances in Cryptology- CRYPTO '85 Proceedings, Conference on the Theory and Application of Cryptographic Techniques*, Santa Barbara, CA, USA, pp. 417-426, 2000. [CrossRef] [Google Scholar] [Publisher Link]

[3] Khalid Javeed, Ali El-Moursy, and David Gregg, "EC-Crypto: Highly Efficient Area-Delay Optimized Elliptic Curve Cryptography Processor," *IEEE Access*, vol. 11, pp. 56649-56662, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] K. Lauter, "The Advantages of Elliptic Curve Cryptography for Wireless Security," *IEEE Wireless Communications*, vol. 11, no. 1, pp. 62-67, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[5] Cristina Vintilă, "Interoperability Issues Among Various IPsec Implementations," *2010 8th International Conference on Communications*, Bucharest, Romania, pp. 411-414, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[6] Junfeng Fan, and Ingrid Verbauwhede, *An Updated Survey on Secure ECC Implementations: Attacks, Countermeasures and Cost*, Cryptography and Security: From Theory to Applications, Springer, Berlin, Heidelberg, pp. 265-282, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[7] Sultan Alqahtani et al., "A Highly Customizable and Efficient Hardware Implementation for Parallel Matrix Inversion," *2022 International Conference on Field-Programmable Technology*, Hong Kong, pp. 1-2, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Jason Cong, "From Parallelization to Customization-Challenges and Opportunities," *2021 IEEE International Parallel and Distributed Processing Symposium*, Portland, OR, USA, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Santosh Ghosh, Debdeep Mukhopadhyay, and Dipanwita Roychowdhury, "Petrel: Power and Timing Attack Resistant Elliptic Curve Scalar Multiplier Based on Programmable GF(P) Arithmetic Unit," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 8, pp. 1798-1812, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[10] Agustin Dominguez-Oviedo, and M. Anwar Hasan, "Error Detection and Fault Tolerance in ECSM Using Input Randomization," *IEEE Transactions on Dependable and Secure Computing*, vol. 6, no. 3, pp. 175-187, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[11] M. Morales-Sandoval et al., "A Reconfigurable GF(2m) Elliptic Curve Cryptographic Coprocessor," *2011 VII Southern Conference on Programmable Logic*, Cordoba, Argentina, pp. 209-214, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[12] Marcus Bednara et al., "Reconfigurable Implementation of Elliptic Curve Crypto Algorithms," *Proceedings 16th International Parallel and Distributed Processing Symposium*, Ft. Lauderdale, FL, USA, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[13] A. Hodjat, D.D. Hwang, and I. Verbauwhede, "A Scalable and High Performance Elliptic Curve Processor with Resistance to Timing Attacks," *International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II*, vol. 1, pp. 538-543, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[14] Gerardo Orlando, and Christof Paar, "A High-Performance Reconfigurable Elliptic Curve Processor For GF(2^m)," *Cryptographic Hardware and Embedded Systems - CHES 2000: Second International Workshop Worcester*, MA, USA, pp. 41-56, 2000. [CrossRef] [Google Scholar] [Publisher Link]

[15] R.C.C. Cheung et al., "Customizable Elliptic Curve Cryptosystems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 9, pp. 1048-1059, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[16] Tim Kerins et al., "Fully Parameterizable Elliptic Curve Cryptography Processor Over GF(2m)," *Field-Programmable Logic and Applications: Reconfigurable Computing Is Going Mainstream*, pp. 750-759, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[17] M. Ernst et al., "A Reconfigurable System on Chip Implementation for Elliptic Curve Cryptography Over GF(2^n)," *Cryptographic Hardware and Embedded Systems - CHES 2002: 4th International Workshop*, Redwood Shores, CA, USA, pp. 381-399, 2003. [CrossRef] [Google Scholar] [Publisher Link]

[18] Partial Reconfiguration User Guide, UG702 (v14.5), Technical Information Portal, 2013. [Online]. Available: https://docs.amd.com/v/u/en-US/ug702

[19] Marc Joye, and Sung-Ming Yen, "The Montgomery Powering Ladder," *Cryptographic Hardware and Embedded Systems - CHES 2002: 4th International Workshop*, Redwood Shores, USA, pp. 291-302, 2003. [CrossRef] [Google Scholar] [Publisher Link]

[20] Medien Zeghid et al., "Speed/Area-Efficient ECC Processor Implementation Over GF(2^m) on FPGA via Novel Algorithm-Architecture Co-Design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 31, no. 8, pp. 1192-1203, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[21] Binh Kieu Do-Nguyen et al., "Multi-Functional Resource-Constrained Elliptic Curve Cryptographic Processor," *IEEE Access*, vol. 11, pp. 4879-4894, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[22] Darrel Hankerson, Scott Vanstone, and Alfred J. Menezes, *Guide to Elliptic Curve Cryptography*, Springer New York, NY, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[23] Carlos Andres Lara-Nino, Arturo Diaz-Perez, and Miguel Morales-Sandoval, "Elliptic Curve Lightweight Cryptography: A Survey," *IEEE Access*, vol. 6, pp. 72514-72550, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[24] Venkata Reddy Kolagatla et al., "A Randomized Montgomery Powering Ladder Exponentiation for Side-Channel Attack Resilient RSA and Leakage Assessment," *2021 25th International Symposium on VLSI Design and Test (VDAT)*, Surat, India, pp. 1-5, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[25] Mouna Bedoui et al., "An Efficient Fault Detection Method for Elliptic Curve Scalar Multiplication Montgomery Algorithm," *2019 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS)*, Gammarth, Tunisia, pp. 1-5, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[26] Ravi Kishore Kodali et al., "FPGA Implementation of Itoh-Tsujii Inversion Algorithm," *International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014)*, Jaipur, India, pp. 1-5, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[27] Chester Rebeiro et al., "Revisiting the Itoh-Tsujii Inversion Algorithm for FPGA Platforms," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 8, pp. 1508-1512, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[28] A. Arunachalamani et al., "High Radix Design for Montgomery Multiplier in FPGA Platform," *2023 International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI)*, Chennai, India, pp. 1-5, 2023. [CrossRef] [Google Scholar] [Publisher Link]