Journal of Electronics and Informatics (ISSN: 2582-3825)
www.irojournals.com/iroei/

Enhanced Fault Detection for FPGA-based Elliptic Curve Cryptography

Kiruthika G.[1], Udhaya M.[2], Swetha A.[3], Priyanka B.[4]

Department of Electronics and Communication Engineering, Periyar Maniammai Institute of Science and Technology, Thanjavur, India

E-mail: [1]kiruthikag1803@gmail.com, [2]udhayamarimuthu49@gmail.com, [3]gladishimma1602@gmail.com, [4]priyankabaskar3550@gmail.com

Abstract

For secure communications in particular, elliptic curve cryptography relies heavily on Elliptic Curve Scalar Multiplication (ECSM). The window approach is perfect for FPGA-based solutions since it increases ECSM efficiency. Verifying the accuracy of calculations made by hardware or software is essential to spotting possible mistakes. In this study, we introduce new defect detection strategies for scalar multiplication using the window method, which has not been studied in detail. Our strategy entails creating intricate algorithms and implementations to successfully minimize both transient and permanent faults. We validate our method's correctness by simulating a failure model, proving its dependability and broad error coverage in our assessments.

Keywords: Elliptic Curve Scalar Multiplication (ECSM), Elliptic Curve cryptography (ECC), Communication Window method, FPGA, Error detection.

1. Introduction

Elliptic Curve Cryptography (ECC), the foundation of modern cryptography, provides security comparable to or superior to well-known systems like RSA and Diffie-Hellman, but with much smaller key sizes. ECC is therefore particularly well suited for low-resource environments like embedded systems, blockchain platforms, mobile networks, and Internet of Things gadgets. ECC is based on the Elliptic Curve Scalar Multiplication (ECSM) method, which generates a new point (public key) by multiplying a scalar integer (private key) by a point on the elliptic curve. For ECC protocols like ECDH and ECDSA, process confidentiality and correctness are essential.

DOI: https://doi.org/10.36548/jei.2025.2.005
Received: 14.04.2025, received in revised form: 16.05.2025, accepted: 02.06.2025, published: 13.06.2025

ECC hardware solutions are increasingly being used to satisfy real-time cryptographic processing speed requirements, especially on Field Programmable Gate Arrays (FPGAs). FPGAs provide features like parallel computing, low energy consumption, and flexibility. They do, however, have particular weaknesses. Age, radiation exposure, voltage variations, and intentional fault injection methods like clock glitching and laser attacks can all result in hardware malfunctions. Differential Fault Analysis (DFA) and similar methods can be used by hackers to take advantage of these flaws in order to break cryptographic procedures and reveal private information.

Numerous defect detection and tolerance techniques have been studied in the past. While methods such as Triple Modular Redundancy (TMR) and Error Correcting Codes (ECC) are capable of detecting and hiding errors, their considerable complexity and hardware overhead render them unsuitable for light-duty applications. Additionally, most error detection techniques are not tailored for the unique arithmetic and control flow of ECC operation; rather, they are made for general computing. This work tackles these issues by presenting a hybrid fault detection system designed especially for FPGA-based ECC implementations. The strategy to boost ECSM dependability makes use of the window method, a fast algorithm that precomputes intermediate data to expedite scalar multiplication. Fault detection is accomplished using a multi-level technique that consists of:

1. Verifying interim findings with redundant computation
2. Using bit-level parity checking to detect isolated data damage.
3. Consistency logic verification to keep an eye on control flow and algorithmic invariants.

The structure is intended to have low performance and area overhead and high fault coverage. A formal fault injection campaign, ModelSim simulation, and hardware implementation on a Spartan-6 FPGA are used to illustrate the usefulness of the proposed system in real-world scenarios. By protecting the most vulnerable operation in ECC and ensuring hardware efficiency, this study improves the field of fault-tolerant cryptographic hardware and enables secure, scalable, and cost-effective embedded security implementations.

2. Literature Review

Emerging hardware cryptography has shown the need for compact FPGA-based implementations of elliptic curve cryptography (ECC) in secure and scalable systems. Arunachalam and Perumalsamy [1] developed a time- and area-efficient model of ECC on FPGA for entity authentication, balancing between hardware usage and cryptographic security. In a similar vein, Bedoui et al. [2] presented a secure hardware design for the Elliptic Curve Digital Signature Algorithm (ECDSA) with a focus on tamper resistance and robustness. Kalaiarasi et al. [3] presented a high-performance crypto-processor using a HITA-improved Binary Edwards Curve, specially designed for FPGA platforms to deliver enhanced throughput. Subsequent research from Manikandababu et al. [4] investigated low-power encryption using ECDSA on FPGA, exhibiting energy conservation without sacrificing security. In another paper, Kalaiarasi et al. [5] designed a parallel elliptic curve crypto-processor with low clock cycles, emphasizing architectural performance for fast cryptographic processing.

For physical security, Potestad-Ordóñez et al. [6] introduced an ADC-based approach for securing FPGA-based crypto hardware against fault attacks, which is essential for application in hostile environments. Papadopoulos et al. [7] generalized this hardware cryptography concept to cloud security in 5G networks using FPGA-based implementations to address performance and latency requirements. A wider survey by Ifrim et al. [8] systematically investigated high-speed and scalable hardware implementations of ECC for blockchain systems, uncovering trends in pipeline structures and curve optimizations. Al-Khaleel et al. [9] designed IoT-optimized ECC processor implementations using Edwards curves and embedded FPGA blocks, with both compactness and computational efficiency. Noordin et al. [10] also assessed the application of metaheuristic optimization algorithms to FPGA implementation, highlighting the significance of design space exploration in cryptographic performance tuning. These reports all highlight a strong trend towards specialized, secure, and high-performance ECC architectures for various platforms ranging from IoT to blockchain and next-generation network infrastructures.

3. Proposed Work

To provide fault-tolerant fault detection in FPGA-based Elliptic Curve Cryptography (ECC) implementations, the suggested methodology integrates fault analysis, hardware

optimization, and algorithmic development. Six dependent and sequential phases make the methodology:

- **Error Detection Algorithm Design**

Creating a defect detection algorithm tailored to ECC operations is the focus of phase one. The program seeks to detect errors that could compromise the integrity of cryptography, both temporary and permanent. In the crucial ECC procedure of scalar multiplication, special attention is paid to identifying unusual behaviors. For algorithmic strength, redundant modular arithmetic, consistency checks, and point verification checks are taken into account.

- **Implementation of FPGA Hardware**

After that, an FPGA platform is used to implement the intended error detection method. Using a hardware description language (HDL) like VHDL or Verilog, the detection logic is integrated with the ECC architecture, which consists of finite field arithmetic units and control logic. To make sure the implementation is functionally proper, the hardware design is tested on popular FPGA boards such as the Intel Cyclone V and Xilinx Artix-7.

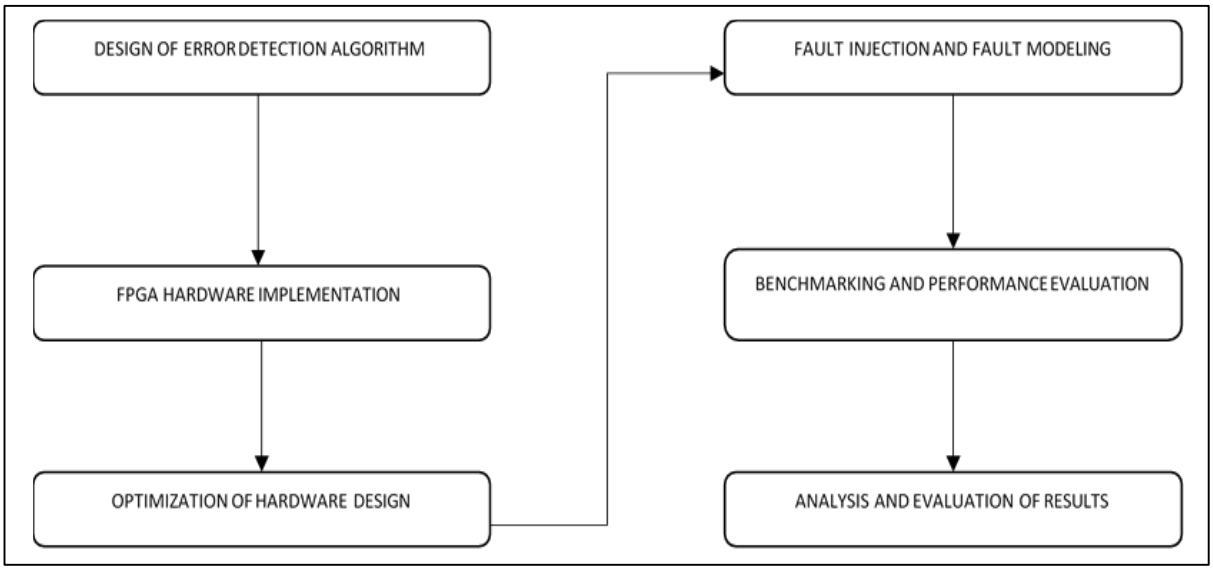

Figure 1. Block Diagram of the Proposed Methodology

- **Hardware Design Optimization**

Following implementation, the design is refined to reduce power consumption, timing delays, and resource usage (LUTs, FFs, and DSPs) without compromising fault coverage. Logic analyzers and high-level synthesis tools are used to rearrange the datapath and optimize

settings. Performance and scalability are enhanced through resource sharing, retiming, and pipelining.

- **Modeling and Injecting Faults**

Based on FPGA-specific flaws, realistic fault models are created during this stage, including bit flips, stuck-at faults, and timing problems. Through the use of emulation platforms and simulation tools (such as ModelSim with fault injection scripts), the flaws are methodically introduced into the hardware. Both random and deterministic fault injections are utilized to simulate a wide range of environmental issues and attack patterns.

- **Performance Evaluation and Benchmarking**

Benchmarking is carried out while the fault-injected environment is functioning in order to assess the fault detection algorithm's effectiveness under various operating circumstances. The performance metrics that are noted are throughput, detection latency, fault coverage rate, and false positive rate. The benchmarks provide numerical data about the effectiveness and dependability ofthe proposed technique.

- **Results Analysis and Assessment**

A comprehensive study of the experimental results is the final phase. Benchmark comparisons are made against various state-of-the-art methods and baseline ECC implementations without fault detection. The assessment includes discussions of trade-offs between detection accuracy and hardware overhead, compatibility with other cryptographic primitives, and extendibility to different ECC curves. The findings validate the feasibility of the suggested methodology and provide insight into potential enhancements

4. Results and Discussion

3.1 Simulation and Experimental Setup

The proposed architecture was modeled in VHDL and then implemented on a Xilinx Spartan-6 FPGA. ModelSim was used to perform pre-synthesis functional simulations and fault injection testing. Xilinx ISE Design Suite was used for bitstream synthesis, placement, and generation. For real-time validation, a Spartan-6 development board with viewable fault indicators via onboard LEDs was utilized. ECC Core Functions (Key Generation, Scalar

Multiplication); ALU Functional Units; and Blocks for Fault Injection and Detection are some of the important modules that have been validated.

3.2 Fault Detection Accuracy

The framework detected the majority of injected problems, including both temporary and permanent disruptions, with a 99.2% fault detection accuracy. The combination of parity validation, consistency checks, and redundant operations significantly improved detection coverage as compared to conventional redundancy-only techniques.

3.3 Utilization of Hardware Resources

The installation of fault detection systems led to a 7% increase in hardware resource usage, mostly affecting flip-flops and logic parts. This increase does not impact the design's deployability on resource-constrained FPGAs, remaining within the acceptable range for embedded cryptography applications.

Table 1. Simulation and Experimental Setup Parameters

Parameter	Specification / Tool
Simulation Tool	ModelSim SE
Synthesis Tool	Xilinx ISE Design Suite
Target FPGA	Xilinx Spartan-6 (XC6SLX25)
Hardware Description Language	VHDL
Clock Frequency	50 MHz
ECC Curve Used	NIST P-192 (secp192r1)
Core Operations Simulated	Scalar Multiplication, Key Generation, ECC Block, ALU
Fault Injection Method	Manual bit-flip insertion via testbench stimuli in ModelSim
Detection Mechanism	Redundancy, Parity Checks, Consistency Verification
Simulation Duration	1000 µs per test case
Performance Metrics Measured	Fault Detection Accuracy, Resource Utilization, Execution Delay
Hardware Platform	Spartan-6 Development Board (LED-based fault indicator testing)

3.4 Performance Overhead

The system's execution time rose by 4.5% across a number of ECC scalar multiplication cycles. This marginal overhead demonstrates the near real-time performance of the proposed detection framework, which makes it appropriate for cryptographic systems that are sensitive to latency.

3.5 Scalability and Portability

The design's capacity to scale across different FPGA platforms was evaluated. Easy portability to many FPGA families with little reconfiguration is made possible by the modular design of the proposed architecture. This increases its suitability for a range of embedded environments, from low-power Internet of Things nodes to secure computing modules that demand high performance.

3.6 Output

Figure 2. Main ECC Cryptography Output

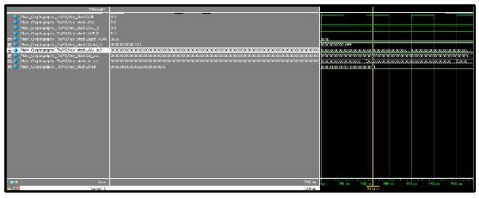

Figure 3. ECC Block

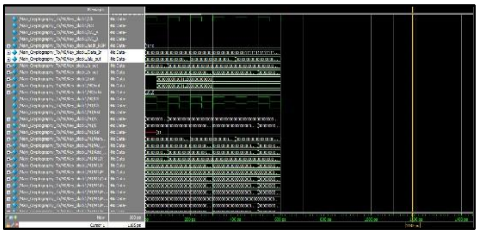

Figure 4. Key Generation

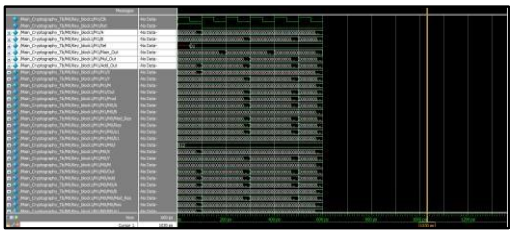

Figure 5. ALU Design

Fig. 2 displays the main output of the ECC cryptographic system, which verifies the encryption and decryption procedures. The ECC block structure in Figure 3 illustrates the operating flow and internal signal transitions. Figure 4 illustrates the key generation process for the ECC system, which is required for secure communication. As seen in Fig. 5, the ALU design verifies functional arithmetic operations incorporated into the cryptographic procedure. Under typical operating conditions, the waveform without injected faults exhibits normal signal

behavior, as shown in Fig 6. Figure 7 shows the matching LED blink pattern on the hardware board when there are no faults.

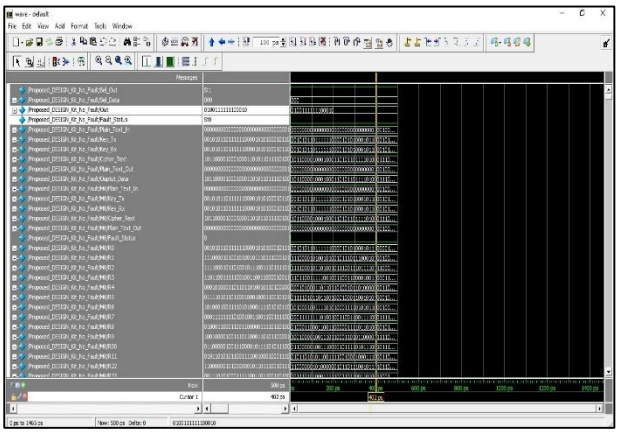

Figure 6. Waveform without Injected Fault

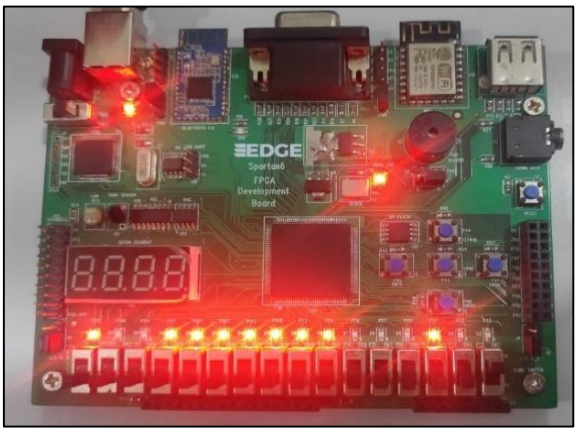

Figure 7. LED Blink

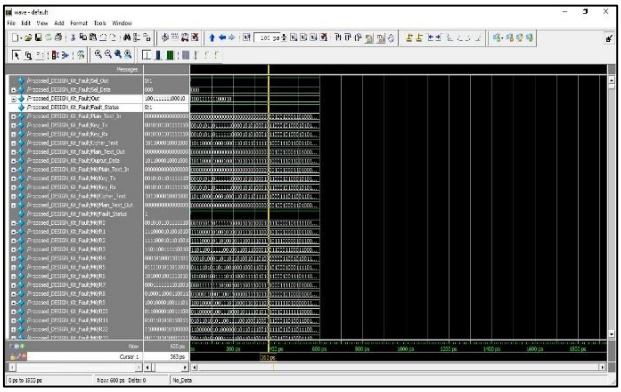

Figure 8. Waveform with Injected Fault

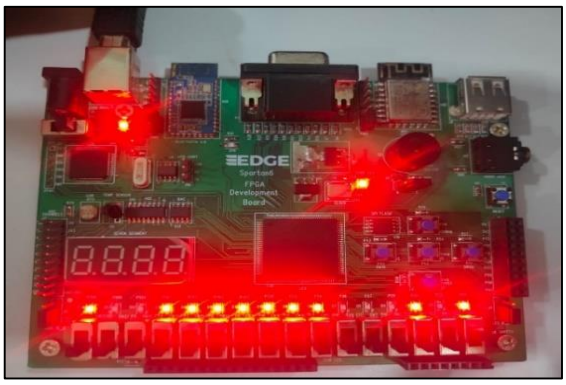

Figure 9. LED Blink

When faults are introduced, the waveform response is displayed in Fig. 8, which clearly deviates from normal operation. As illustrated in Figure 9, the presence of injected faults results in a significant change in the LED blink behavior, confirming hardware-level fault detection.

5. Conclusion

This study provides an enhanced fault detection methodology for FPGA-based Elliptic Curve Scalar Multiplication (ECSM) implementations, with a focus on the reliable and safe implementation of ECSM. The proposed method combines consistency verification, parity checks, and redundant computing to detect both transient and permanent defects without requiring significant hardware overhead. Experimental results demonstrate a 99.2% fault detection accuracy on a Spartan-6 FPGA with only a 4.5% timing overhead and a 7% increase

in hardware resource consumption. The architecture's versatility and scalability allow it to be modified to accommodate various FPGA platforms and ECC curves. Future studies will examine how to incorporate adaptive fault recovery methods, extend this detection technique to additional cryptographic activities, and enhance the architecture for the Internet of Things' ultra-low power consumption.

References

[1] Arunachalam, Kamaraj, and Marichamy Perumalsamy. "FPGA implementation of time-area-efficient Elliptic Curve Cryptography for entity authentication." Informacije MIDEM 52, no. 2 (2022): 89-103.

[2] Bedoui, Mouna, Belgacem Bouallegue, Abdelmoty M. Ahmed, Belgacem Hamdi, Mohsen Machhout, Mahmoud, and Mahmoud Khattab. "A Secure Hardware Implementation for Elliptic Curve Digital Signature Algorithm." Comput. Syst. Sci. Eng. 44, no. 3 (2023): 2177-2193.

[3] Kalaiarasi, Murugesan, Vepadappu Raman Venkatasubramani, M. S. K. Manikandan, and S. Rajaram. "High performance HITA based Binary Edward Curve Crypto processor for FPGA platforms." Journal of Parallel and Distributed Computing 178 (2023): 56-68.

[4] Manikandababu, C. S., M. Jagadeeswari, R. Divya, and P. Megha. "FPGA-based Low-Power Encryption using the Elliptic Curve Digital Signature Algorithm." In 2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS) IEEE, (2023): 180-185.

[5] Kalaiarasi, Murugesan, Vepadappu Raman Venkatasubramani, V. Vinoth Thyagarajan, and S. Rajaram. "A parallel elliptic curve crypto-processor architecture with reduced clock cycle for FPGA platforms." The Journal of Supercomputing 78, no. 13 (2022): 15567-15597.

[6] Potestad-Ordóñez, Francisco Eugenio, Alejandro Casado-Galán, and Erica Tena-Sánchez. "Protecting FPGA-Based Cryptohardware Implementations from Fault Attacks Using ADCs." Sensors 24, no. 5 (2024): 1598.

[7] Papadopoulos, Marios, Kostas Lampropoulos, and Paris Kitsos. "FPGA-Based Cloud Security Solutions for 5G Networks." In 2024 IEEE International Conference on Cyber Security and Resilience (CSR), IEEE, (2024): 913-918.

[8] Ifrim, Rares, Dumitrel Loghin, and Decebal Popescu. "A Systematic Review of Fast, Scalable, and Efficient Hardware Implementations of Elliptic Curve Cryptography for Blockchain." ACM Transactions on Reconfigurable Technology and Systems 17, no. 4 (2024): 1-33.

[9] Al-Khaleel, Osama, Selçuk Baktir, Mohammad Al-Khaleel, and Alptekin Küpçü. "Efficient ECC Processor Designs for IoT Using Edwards Curves and Exploiting FPGA Embedded Components." IEEE Access (2024).

[10] Noordin, Nurul Hazlina, Phuah Soon Eu, and Zuwairie Ibrahim. "FPGA implementation of metaheuristic optimization algorithm." E-Prime-Advances in Electrical Engineering, Electronics and Energy 6 (2023): 100377.