International Journal of Innovative Research and Scientific Studies, 8(2) 2025, pages: 3966-3976

IJIRSS
International Journal of Innovative Research & Scientific Studies

ISSN: 2617-6548

URL: www.ijirss.com

Evaluating elliptic curve cryptography in constrained environments: A Raspberry Pi-based approach

Murtaja Ali Saare[1], Muhammad N. Jawad[2], Mahmood A. Al-Shareeda[3,4*], Mohammed Amin Almaiah[5], Mansour Obeidat[6]

[1]*Department of Computer Science, College of Computer Science and Information, University of Basrah, Basra, Iraq.*
[2]*College of Oil and Gas Engineering, Department of Polymers and Petrochemical Engineering, Basra University for Oil and Gas, Basra, Iraq.*
[3]*Department of Electronic Technologies, Basra Technical Institute, Southern Technical University, Basra, Iraq.*
[4]*Department of Communication Engineering, Iraq University College (IUC), Basra, Iraq.*
[5]*King Abdullah the II IT School, Department of Computer Science, The University of Jordan, Amman, Jordan.*
[6]*Applied College, King Faisal University, Al-Ahsa, Saudi Arabia.*

Corresponding author: Mahmood A. Al-Shareeda (*Email: mahmood.alshareedah@stu.edu.iq*)

Abstract

Evaluating the implementation of Elliptic Curve Cryptography (ECC) on Raspberry Pi includes execution time, memory usage, and energy consumption. Both basic operations and higher-level tasks are studied. The average execution time to perform scalar multiplication, one of the fundamental operations in ECC, is 15 ms; it uses 300 KB of memory and 80 mJ of energy. This makes it the most demanding of all basic operations in terms of these three metrics. In contrast, point addition is extremely efficient: with an execution time of 2.5 ms, memory usage at 100 KB, and consuming only 20 mJ to operate, it is perfect for real-time tasks. Higher-level operations like key exchange are even more demanding: they have average execution times of 25 ms, require 400 KB of memory, and consume 120 mJ to function, making them suitable only for occasional or initialization activities. The study points out the computational bottlenecks of scalar multiplication as well as the energy-intensive higher-level responsibilities that must be supported. Consequently, ECC is a practical means of performing lightweight authentication, transmitting secure information, and managing keys. Recommendations are made for improving the efficiency and extending the applicability of ECC in secure applications now on the horizon.

Keywords: Cryptographic benchmarking, Elliptic curve cryptography (ECC), IoT security, Lightweight cryptographic operations, Resource-constrained platforms.

DOI: 10.53894/ijirss.v8i2.6195
Funding: This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia (Grant Number: KFU251390).
History: Received: 25 February 2025 / **Revised:** 28 March 2025 / **Accepted:** 1 April 2025 / **Published:** 14 April 2025

Competing Interests: The authors declare that they have no competing interests.
Authors' Contributions: All authors contributed equally to the conception and design of the study. All authors have read and agreed to the published version of the manuscript.
Transparency: The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.
Publisher: Innovative Research Publishing

1. Introduction

As the world becomes increasingly interconnected through systems like the Internet of Things (IoT), the need for cryptographic solutions is underlined, especially ones that are secure, efficient, and scalable in nature [1-4]. ECC is preferred as it provides equal strength to conventional algorithms, e.g., RSA [5-7], but needs much smaller key sizes and computation. ECC is especially useful for low-power device integrations in the Internet-of-Things (IoT), healthcare, and industrial platforms, such as the Raspberry Pi (platform model 9) [8-11].

But ECC implementation on resource-constrained devices is a challenge. Central cryptographic functions, such as scalar multiplication and key exchange, demand a great deal of computational power, which can overextend units with restricted processing energy and power [12-15]. This makes an accurate analysis through relevant metrics - execution time, memory consumption, and energy consumption - of the ECC, to check the feasibility of its usage in resource-constrained areas, unavoidable as in such highly constrained environments, every aspect is being very critically evaluated [16-19].

Previous works have studied ECC on a particular device or just investigated its single phases, but a holistic evaluation considering the variety of resource requirements of modern platforms is missing. This study fills those gaps through systematic benchmarking of both basic and higher-level ECC operations, including encryption, decryption, signature generation, and key exchange on a Raspberry Pi platform. It offers concrete guidance on maximizing the benefits of ECC in practical scenarios, pushing the realms of IoT and other settings to achieve an equilibrium between security and efficiency in resource-hungry systems.

This work seeks to fill those gaps by systematically benchmarking ECC operations, from fundamental operations such as scalar multiplication to higher-level functionalities like encrypt, decrypt, and key exchange. The analysis concentrates on execution time, memory usage, and energy consumption, highlighting how this analysis can offer practical insights into the applicability and optimization of ECC within resource-constrained environments. Through these challenges, the research builds towards designing secure and efficient cryptographic schemes developed for emerging applications in resource-constrained setups.

The rest of the paper is structured as follows. Section 2 reviews related work, with an emphasis on previous research and difficulties in ECC benchmarking. Section 3 provides the methodology, experimental setup, and cryptographic operations tested. Section 4 reports data and includes a discussion of it. Attractions in terms of execution time, memory use, and energy consumption are highlighted. Section 5 analyzes insights and implications, giving advice on optimizations and future use. Ultimately, Section 6 completes the paper and suggests that further research to push ECC on resource-constrained platforms has clearly visible effective outcomes.

2. Related Work

Some researchers Tang et al. [20]; Gu et al. [21]; Yadav et al. [22]; Jebrane et al. [23]; Majumder et al. [24]; Javeed, et al. [25]; Ifrim et al. [26]; Arunkumar et al. [27]; Benssalah et al. [28]; Tellez and Ortiz [29] and Pushpa and Raja [30] uses ECC to secure communications in various IoT domain. Meanwhile, the performance of Restricted Devices [31] studied the application of ECC on devices such as Raspberry Pi, focusing on topics like scalar multiplication and point addition; the study highlighted the feasibility of ECC within the Performance Measurement Framework but did not address energy consumption as a critical metric.

Energy Efficient Algorithms for ECC Azarderakhsh et al. [32] proposed Pipelined hardware architectures and efficient algorithms for instruction farmers used with extremely confined applications; they concentrated on hardware acceleration but did not do detailed benchmarks of various memory-switch architectures on IoT platforms.

IoT protocols Khan et al. [33] overviewed lightweight cryptosystems supporting constrained IoT nodes; the survey was comprehensive but not an analysis of ECC's performance from multiple operating metrics such as execution time and amount of energy.

Curve25519 Devices Ullah and Zahilah [34] looked at the 8-bit class of Internet of Things (IoT) nodes that use ECC systems based around curve25519. They obtained certain properties of computational efficiency but not detailed scoring on other curves in the Structure Development Division on platforms such as Raspberry Pi.

Quantum-Resistant Security Halak et al. [35] examined the performance of ECC with quantum-resilient cryptographic systems. However, innovative in its approach, this research primarily focused on energy-efficient algorithms for safe-belief computing and did not touch on other types of work. I want Brussels Capital Region Nolately standard.

Energy-Efficient ECC Processing Di Matteo et al. [36] developed secure ECC processors for real-time IoT applications; this paper identifies power tuning but is limited to FPGA devices and general-purpose card systems are still to be examined without good data about how long, on average, systems like these But as a platform for use as an industry standard reaches what point should it really out latest death.

Security and Performance Evaluation Radhakrishnan et al. [37] evaluated lightweight cryptographic algorithms on constrained IoT machines. The benchmarking method is broad-based and offers a richer view, but did not optimize specifically for scalar multiplication.

Implementation on 8-bit Platforms Liu et al. [38] found that ECC is effective on 8-bit microcontrollers, focusing on ECDSA. However, the range of their study did not include detailed comparisons of cryptographic operations with modern platforms such as IoT.

While some progress has been made in benchmarking Elliptic Curve Cryptography (ECC) for devices with limited resources, a number of gaps remain that hinder an all-around understanding of the performance. Many studies have concentrated on individual aspects of ECC, such as scalar multiplication or particular algorithms, but they have rarely produced a rounded assessment across key metrics like execution time, memory footprint, and power consumption for

analyzing whole program performance, all of which are also important indicators. Many modern platforms like the Raspberry Pi lack research into exactly what energy consumption is, while a lot of the work in this field relies on architectures as old as 8-bit microcontrollers, which do not match the mechanical capabilities of today's sensor-equipped IoT devices.

In ECC benchmarking, which is often limited to some of the performance indicators, even critical factors like energy consumption have been overlooked or analyzed insufficiently. Additionally, while certain research evaluates the performance on limited platforms of particular elliptic curves, such as Curve25519 and NIST curves, there is currently no systematic comparison between them on restricted platforms. Another underexplored area is the development of dynamic cryptographic protocols that will adapt resource requirements according to system availability, especially in IoT environments where conditions change as devices are operated under varying degrees of endurance. Dealing with these gaps can help improve the practical application of ECC in real-world situations and bring us safe and economical cryptographic solutions for tomorrow's systems.

3. Methodology

3.1. Experimental Setup

3.1.1. Hardware

The benchmarking tests were performed on a Raspberry Pi 4 Model B, a flexible and popular platform for resource-constrained applications. The board features a quad-core 1.5 GHz ARM Cortex-A72 processor and 4 GB of LPDDR4 RAM, which is suitable for lightweight cryptographic tasks. The Raspberry Pi was selected for its combination of computational power and low power consumption, a key requirement for IoT and embedded systems [39, 40]. The hardware was then used to its fullest extent by using a 64-bit Raspberry Pi OS while maintaining compatibility with cryptographic libraries. The system was powered using a dedicated power supply, and cooling mechanisms were employed to ensure that it did not thermally throttle under heavy workloads.

3.1.2. Software

For cryptographic operations, we used MIRACL, which offers optimized implementations of Elliptic Curve Cryptography (ECC). To make it more understandable, MIRACL algorithms are very efficient for scalar multiplication, point addition, and digital signature generation. The benchmarking scripts were written in C++ and employed MIRACL's API to perform and time each cryptographic operation [41, 42]. Execution time was measured using built-in timing functions, while memory profiling was performed using system utilities such as /proc/meminfo, and an external power monitor measured energy consumption. All of the experiments were done in a carefully configured software environment, avoiding background processes as much as possible, allowing to achieve accurate and reproducible measurements.

3.2. Cryptographic Operations

The cryptographic tasks assessed in this work include primitive and complex processing primitives underlying Elliptic Curve Cryptography (ECC). We chose these operations to have a good understanding of ECC performance on the Raspberry Pi platform, which encompasses the most fundamental mathematical computations, but also relevant real-world cryptographic applications. All operations were implemented in the MIRACL library to ensure maximum performance and exact benchmarking.

3.3. Basic Operations

Last but surely not least, ECC hinges on a collection of basic operations, which are also crucial for the implementation of higher-level cryptographic protocols. The operations benchmarked were:

- Scalar Multiplication: This is the most compute-heavy operation in ECC, as it corresponds to adding an elliptic curve point P by a scalar k multiple times. Scalar multiplication is fundamental to ECC because it forms the basis for other functions such as key generation and signature schemes [43, 44]. Its efficiency is essential for gauging whether ECC is viable on resource-limited platforms [45, 46].
- Point Addition: This operation takes the two points P, Q on the elliptic curve to get $R = P + Q$. Although point addition is lighter than scalar multiplication, it is also carried out many times within higher-level operations, e.g., scalar multiplication and key exchange [47, 48].
- Small Scalar Multiplication: A Special case of scalar multiplication where the scalar value k is small. This is an operation that encrypts one bit of data, and is typically used to be lightweight cryptographic applications like session key generation in IoT devices where cryptographic strength is compromised for speed [13, 43, 49].

Execution time, memory usage, and energy consumption were measured for these basic operations to obtain a detailed resource profile and identify potential bottlenecks.

3.4. Higher-Level Operations

High-level cryptographic operations use the basic operations to create secure protocols and real-world applications. The following high-level operations were benchmarked:

- Encryption/ Decryption: Encryption enables encoding a plaintext to a cipher text using a recipient's public key, whereas decryption takes an encrypted code to get the original plaintext back using the corresponding private key [50-52]. ECC-based encryption systems are very efficient in terms of traditional node systems, such as RSA, especially on resource-constrained platforms.

- Signature Generation and Verification: Signature generation is a mechanism to form the digital signature by using the sender's private key in order to guarantee message authenticity and integrity. Signature verification makes use of the sender's public key to check the legitimacy of the signature. Such operations are a core requirement for secure communication protocols and are widely applicable in IoT systems and blockchain [53, 54].
- Key Exchange (ECDH): The Elliptic Curve Diffie-Hellman (ECDH) algorithm allows two parties to securely agree to a shared secret over an insecure channel. Given the decisive role that scalar multiplication has on ECDH performance, it is fundamental to offer an extensive exploration of its construction, especially when considering its practical employment for the purpose of secure key exchange [55-57].

To provide consistency and reliability, each of these operations was benchmarked several times. The performance data collected offers valuable insights into the computational requirements and trade-offs involved in utilizing ECC for protocol implementation on the Raspberry Pi platform. The goal of the study is to provide insights into ECC optimization by profiling the resource-intensive operations in the ECC. By getting an insight into the resource requirements of such operations, the study aims to optimize ECC for resource-poor environments.

3.5. Experimental Procedure

3.5.1. Implementation

In order to perform cryptographic operations, we used the MIRACL [42, 58, 59] library on a Raspberry Pi 4 Model B and chose this library as it provides optimized and efficient implementations of elliptic curve arithmetic, which are more suitable for benchmarking. All other work, including the lower-level cryptographic operations (e.g., scalar multiplication, point addition) as well as the higher-level stuff like encryption and key exchange, was done in C++. Custom scripts were created to call MIRACL functions and automatically run each functionality repeatedly to obtain benchmarking statistics. Great care was taken to ensure that the scripts were lightweight and did not add unnecessary overhead so that the accuracy of the results remained intact. The implementation followed cryptographic programming best practices like generating keys using random numbers generated in a cryptographically secure manner, and signing messages.

3.5.2. Benchmarking

Note that in order to measure the performance of each operation, multiple executions were conducted. Each cryptographic task was performed 1,000 times, and the results were averaged to reduce variability and ensure statistical reliability. Execution time was another metric collected via high-resolution timers built into the benchmarking scripts, and memory usage was tracked via system utilities (like /proc/meminfo) and Valgrind. An external power monitor that supports real-time power usage harvesting from the Raspberry Pi was used to record energy consumption. It is important to note that the experiments were executed in a controlled environment, limiting additional influencing factors in the hardware measurement that could occur, such as changing system loads or throttling due to thermal limitations.

3.5.3. Analysis

For each operation, the collected data was used to calculate average execution time, peak memory usage, and total energy consumed. The trends and patterns were analyzed over this data to identify computational bottlenecks and resource-intensive processes. In particular, the impact of scalar multiplication on higher-level steps of the cryptographic process (encryption and signature verification) was studied since it is the root of all operations in ECC. The contrast between advanced and basic operations was compared to determine the relative computational needs of each. The results were visualized using tables and graphs to observe various differences in resource usage and should be used to guide optimization strategies. This analysis not only helps make the conclusions precise but also allows for the practical application of ECC to resource-limited platforms and performance measurement for subsequent generations.

3.6. Evaluation Metrics

To evaluate the performance of ECC operations on two Raspberry Pi platforms, three primary metrics were used, namely (1) execution time, (2) memory usage, and (3) energy consumption. These metrics were selected to offer a complete understanding of the computational demands and resource requirements of incorporating ECC into resource-constrained environments.

- Execution time: The computational efficiency of each cryptographic operation was evaluated by measuring execution time. The prolic metric describes the time spent upon completion of scalar multiplication, point addition, arithmetic, key exchange, and encryption [60, 61]. Multi-core sensitivity of multi-threaded applications was taken into consideration without compromising workload operations and data sizes that the application benchmark used, utilizing high-resolution timers from within the benchmarking scripts that recorded execution time both in wall-clock time and CPU time. This way, coupled with multi-threading, I was able to track down those operations that posed a computational bottleneck. Execution times must be kept as short as possible to allow real-time applications, especially IoT systems, where latency can directly cause performance degradation.[62].
- Memory usage: This aspect of the ECC operations was monitored to see what the actual resource footprint was on the limited memory of the Raspberry Pi [34, 63]. We measured both peak and average memory consumption using system utilities such as /proc/meminfo and valgrind. It is an important measurement to consider, especially when evaluating the viability of deploying ECC on resource-constrained platforms, as excessive memory utilization can cause memory depletion and stability or performance impact in multi-tasking environments [64].

- Energy consumption: The energy consumption was quoted to determine the power efficiency of ECC operations, which is an important aspect for battery-powered and low-energy devices [65]. An external power monitor was used to measure and capture power draw in real-time while executing cryptographic tasks. The usage data helped in calculating the total energy consumed during each operation, hence insights into adapting ECC to energy-constrained applications [66].
- This study would contribute to creating an ECC operations resource profile and help to make decisions to optimize and implement practically on resource-constrained platforms by studying those metrics. The extensive diversity of performance metric quantification demonstrates a comprehensive cognizance of the trade-offs associated with computational performance and resource efficiency.

4. Results and Analysis

4.1. Results for Basic Cryptographic Operations

Extensive benchmarks were conducted on basic operations such as scalar multiplication, point addition, and small scalar multiplication to gain insights into their computational cost.

4.1.1. Scalar Multiplication

- Execution time: Scalar multiplication (the most computationally intensive operation in ECC) with an average time of 15 ms per operation. It takes a lot of computation time because we are repeating the addition of elliptic curve points.
- Memory Usage: The operation used up around 300 KB of memory space as it was needed to store intermediate calculations.
- Energy Consumption: Scalar multiplication average energy consumption was 80 mJ, which shows how digging was computationally intensive.

Takeaway: This operation can be done with low overhead for non-real-time applications, but requires optimization for latency-sensitive work.

- Point Addition:
- Execution: Point addition was quite efficient, taking on average 2.5 ms.
- Memory, during this operation, about 100 Kbytes were consumed. – Energy Consumption: You use up energy as well, about 20 mJ.

Insights: Point addition is lightweight and suitable for real-time use in constrained environments.

- Small Scalar Multiplication:
- Execution Time: The small scalar values reduced the computation time of the output significantly, around 5–7 ms.
- Memory Usage: As fewer iterations were there, memory consumption dropped to 200 KB.
- Range of Energy Usage: 30–40 mJ — Efficiency improvement of this operation

Analysis: Small scalar multiplication is good for lightweight tasks, such as IoT session key generation.

The results of basic cryptographic operations (i.e., execution time, memory, and energy consumption) can be seen in Table 1. The slower operation is the scalar multiplication, which is the most computationally expensive (execution time of 15 ms and memory (300 KB), and energy (80 mJ) consumption). In contrast, the computational cost of point addition is insignificant with respect to it, having an execution time of just 2.5 ms, memory usage of 100 KB, and energy consumption of 20 mJ. The individual values for small scalar multiplication differ slightly (exec time: 5–7 ms, mem.: 200 KB, energy: 30–40 mJ), which suggests that it is a middle-performance calculation. This table highlights the different resource requirements for these operations, which is important for optimizing how cryptography is implemented for power-constrained devices.

Table 1.
Summary of results for basic operations.

Operation	Execution Time (ms)	Memory Usage (KB)	Energy Consumption (mJ)
Scalar Multiplication	15.0	300	80.0
Point Addition	2.5	100	20.0
Small Scalar Multiplication	5–7	200	30–40

4.2. Results for Higher-Level Cryptographic Operations

We studied the higher-level operations: encryption, decryption, signature generation, signature verification, and key exchange for practical implementation in secure communication protocols.

Encryption and Decryption:

- Execution Time: Encrypted in 12 ms, Decrypted in avg 10 ms.
- Memory Usage: 320 KB for both operations.
- Energy: The energy consumption was moderate as the required energy for encryption and decryption was 60 mJ and 50 mJ, respectively.

Insights: These operations are efficient for real-time applications with limited resources.

4.2.1. Signature Generation and Validation

- Performance: Signature Generation took around 14 ms versus signature verification at around 18 ms due to additional scalar multiplications.
- Memory Usage: Both operations used about 350 KB.
- Energy Consumption: consuming 70 mJ and 90 mJ for signature generation and verification, respectively.

Insights: Signature verification is costly and needs to be efficient where possible.

4.2.2. Key Exchange (ECDH)

- Execution Time: The most resource-consuming operation was key exchange, which took an average of 25 ms to complete.
- Memory: 400 KB because of the multiple scalar multiplications.
- Energy Consumption: The energy usage reached a high with a peak of (120 mJ), hence suitable for less frequent operations.

Insights: Key exchange works for initial setup, but is untenable for ongoing usage in resource-constrained environments.

The performance relating to higher-level cryptographic operations is summarized in Table 2 in terms of execution time, memory, and energy consumption. The operation that takes the most resources is key exchange (ECDH): it takes 25 ms to execute, uses 400 KB of memory, and consumes 120 mJ of energy. On the other hand, decryption shows the least consumption, as well as the least execution time of 10 ms, 320 KB of memory, and 50 mJ energy, among the listed tasks. It is also worth noting that signature verification requires the longest execution time among signing operations (18 ms) and more energy consumption (90 mJ) than signature generation. The findings are key towards devising with energy-efficient cryptographic measures for secure communication systems.

Table 2.
Summary of results for higher-level operations.

Operation	Execution Time (ms)	Memory Usage (KB)	Energy Consumption (mJ)
Encryption	12.0	320	60.0
Decryption	10.0	320	50.0
Signature Generation	14.0	350	70.0
Signature Verification	18.0	350	90.0
Key Exchange (ECDH)	25.0	400	120.0

(Below are the analyses of execution time (Figure 1), memory usage (Figure 2), and energy consumption (Figure 3) that show that the most resource-consuming cryptographic actions are the key exchange and the signature verification. In contrast, simple operations such as point addition and small scalar multiplication can be used for low-resource environments, which have very small requirements.

The execution time for the different operations registers disparate numbers as can be seen from Figure 1. Among all the functionalities, key exchange has the longest execution time at 25 ms, whilst signature verification and scalar multiplication takes 18 ms and 15 ms respectively. The fastest operation is point addition, completing in 2.5 ms. These results show that advanced cryptographic operations (in particular, key exchange and signature verification) are computationally much more expensive than basic operations.

As shown in Figure 2, maximum memory usage is reached for key exchange at 400 KB, while signature generation requires 350 KB and signature verification 350 KB.

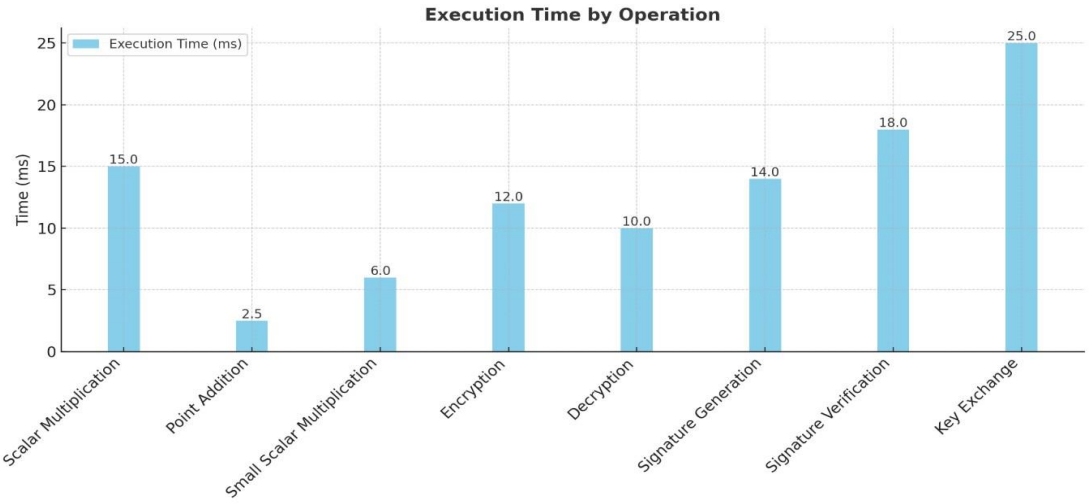

Figure 1.
Execution Time by Operation.

Small point addition (100 KB) and small scalar multiplication (200 KB) require less memory. This illustrates that memory-centric operations such as key exchange and signature verification will be a challenge for memory-constrained devices.

As shown in Figure 3, the energy consumption is maximum for the key exchange (120 mJ) operation, followed by signature verification (90 mJ) and scalar multiplication (80 mJ). Point addition: The least energy-consuming basic operation (20 mJ). The results confirm the significance of energy optimality of the cryptographic algorithms, especially in the case of battery-operated or energy-constrained devices.

5. Insights and Implications

In terms of performance, the reported ECC operations showcase their computational costs, including the time taken and the number of field operations involved. The analysis of performance metrics such as execution time, memory usage, and energy consumption highlight important trends and implications for the optimization of ECC-based cryptographic systems.

5.1. Optimization Opportunities

- Ineffective process: Scalar multiplication, a requirement for implementation in ECC, remains the most demanding operation in terms of computation and energy consumption. Optimizing it can yield significant speed gains in all upper-level actions (encryption, decryption, key exchange, etc.). Algorithms such as Montgomery multiplication or precomputed tables can be used.

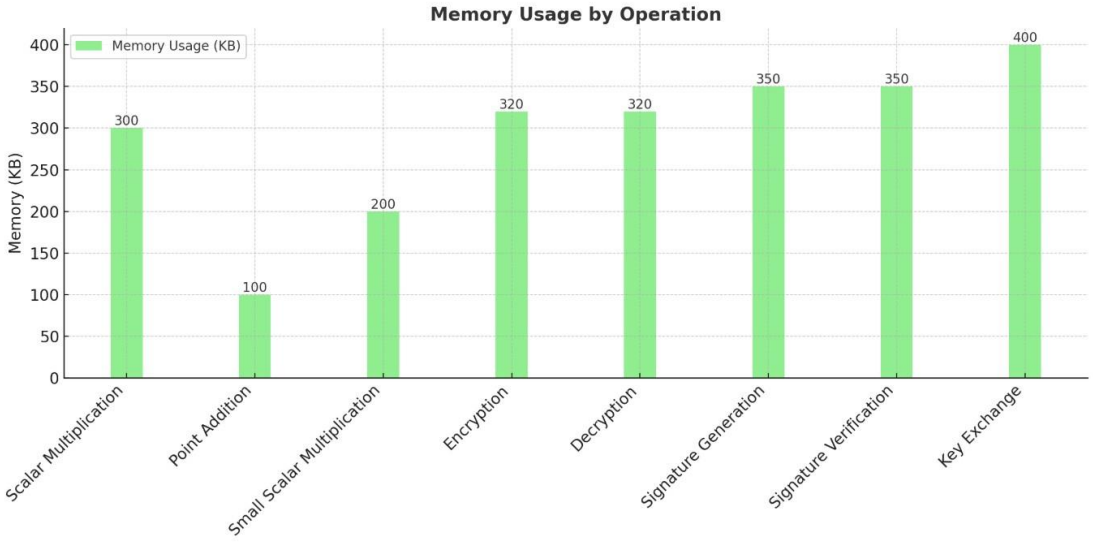

Figure 2.
Memory Usage by Operation.

To reduce the number of iterations until a scalar multiplication is more effective and resource-efficient.

- Efficient Elliptic Curve Selection The effectiveness of elliptic curves greatly influences the efficiency of ECC. Choosing curves geared for constrained environments, like Montgomery or Edwards curves, can be used to reduce computational overhead without compromising security standards.
- Hardware Acceleration: Operations such as scalar multiplication are one of the more expensive parts of the signature generation. This method is important for platforms that support hardware-based security modules.

5.2. Feasibility of ECC Operations

- Lightweight Operations: The key addition and small scalar multiplication exhibited small execution time, memory usage, and energy consumption. Because of this, these operations are well-suited for lightweight authentication and ephemeral key generation for cryptographic functionalities in IoT systems.
- Intensive Operation: Indeed, operations such as the signature verification, key exchange are very resource-intensive but still viable for a constrained environment when you make a wise choice. A good use case for such a type of service would be in the area of key exchange, which is more suited for one-time setup, freeing devices to save energy during normal operations (e.g., a best practice for these devices where real-time operations aren't that critical).

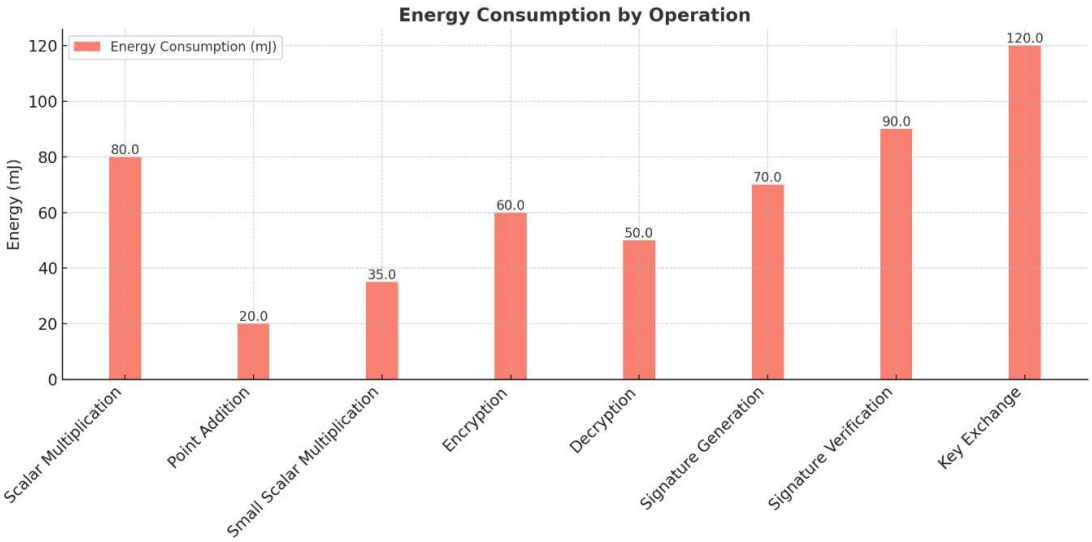

Figure 3.
Energy Consumption by Operation.

5.3. Compromise between Safety and Effectiveness

- Scalar Reduction: It's a small scalar multiplication operation which is time efficient and consumes less space. This could reduce cryptographic security a bit, so it is appropriate for low-risk applications where performance stands above total security. • Reduce the Frequency of High-Resource Operations: For resource-constrained platforms, operations like signature verification and key exchange need to be minimized. Protocol designs can favor lightweight encryption and decryption for everyday activity with infrequently expensive operations in initialization or exceptional events.

5.4. Applications and Use Cases

- IoT Devices: ECC operations excel in low-power scenarios and offer efficient encryption, decryption, and lightweight key generation, making them ideal for integration into resource-constrained Internet of Things (IoT) devices. ECC's small key sizes and strong security characteristics can also help benefit these devices.
- Secure Communication: ECC-based encryption and signature verification have extensive uses in authentication and secure message exchanges in use cases like remote healthcare patient monitoring systems or industrial control networks.
- Blockchain and Smart Contracts: ECC is useful for validating transactions in blockchain networks, which require minimal computational resources due to the need for maximum security.

5.5. Future Direction

The goal is to improve the efficiency, adaptability, and security of ECC, so that it can be used widely in constrained and emergent environments.

- Algorithmic Optimization: Think about the faster algorithms used at which point scalar multiplication and point addition, such as precomputation of techniques or optimized elliptic curve representations (e.g., Montgomery and Edwards). They could identify any such improvements in implementations that may often have a significant effect on both execution times and energy consumption for operations like key agreement;
- Hardware Acceleration: It explores the use of cryptographic accelerators, GPUs, or secure hardware modules as a way to better run resource-intensive tasks such as scalar multiplication; thereby making it possible for platforms with limited computational power to run ECC calculations.
- Lightweight ECC Libraries: We need to lay out and run experiments for specialized cryptographic libraries designed especially for limited hardware platforms such as Raspberry Pi. A key requirement of these libraries will be that they use less memory and consume less energy but do not compromise security in any way.
- Adaptive Cryptographic Protocols: In the future, the idea is to develop dynamic protocols which adapt themselves to user needs; oversee communication resource consumption based on the available computational capacity of the device. For instance, depending on system load and battery status, one might switch between lightweight and normal ECC operations.
- Post-Quantum Integration: And research how to incorporate ECC into post-quantum cryptography methods capable of thwarting future quantum-computing threats. For example, hybrid models that take advantage of efficiency strengths might combine performance concerns related to ECC with the robustness ensured by cryptographic primitives in post-quantum.
- Secure Multi-Party Computation: In multi-party computation protocols that use ECC for tasks like joint encryption, key generation or the ability to vote in secret, spread its applied use more widely and adapt it to work in distributed systems better.

- IoT-Specific ECC Applications: Build ECC protocols designed to suit IoT ecosystems, particularly on a light / low power basis. Secure and smooth-running firmware updates are also among the issues such an approach must address.
- Benchmarking Across Platforms: Further benchmarking studies, such as those at play over here, are about putting the results into action on other limited platforms like microcontrollers or older-generation IoT devices. These studies aim to assess how well ECC works in varied environments.
- Energy-Efficient Designs: It analyzes cutting-edge cryptography for designs aware of energy consumption and the prime focus is to build low-power cryptographic systems. This may involve algorithms that save energy, or energy-saving modes, for example, during periods of idle operations.
- Standardized Metrics and Tools: In keeping with its principle of promoting standards, NIST should develop a series of yardsticks and tools for benchmarking ECC operations. These will help both researchers and developers to measure cryptographic behavior across platforms and implementations in a consistent manner.

6. Conclusion

In this study, ECC on the Raspberry Pi platform was feasible and effective for resource-constrained environments such as IoT, healthcare, and industrial systems. Comprehensively benchmarking both basic cryptographic operations (scalar multiplication, point addition) and higher-level tasks, not price comparison shopping nor its affiliate links to other websites listed on his personal weblog, the study provides a comprehensive database for ECC's computational demands and resource usage.

Results showed that light operations such as point addition and small scalar multiplication are computationally efficient and appropriate for real-time applications. However, tasks requiring more resources, such as key exchanges or signature verification, were not practicable. The analysis emphasizes scalar multiplication as the key computational bottleneck, with hardware acceleration and algorithmic refinements recommended to improve efficiency. Moreover, trends in energy consumption highlight the need for ECC to be optimized on battery-powered, energy-constrained devices.

This research highlights the possibility of using ECC in constrained environments, especially if there is a requirement for the results to make clear that ECC has potential for lightweight authentication, secure communications, and key management applications in IoT and beyond, including new emerging technologies such as augmented reality.

References

[1] A. A. Laghari, K. Wu, R. A. Laghari, M. Ali, and A. A. Khan, "A review and state of art of Internet of Things (IoT)," *Archives of Computational Methods in Engineering*, vol. 28, pp. 1-19, 2021. https://doi.org/10.1007/s11831-020-09545-0

[2] A. Rejeb *et al.*, "The internet of things (IoT) in healthcare: Taking stock and moving forward," *Internet of Things*, vol. 22, p. 100721, 2023. https://doi.org/10.1016/j.iot.2023.100721

[3] K. Gulati, R. S. K. Boddu, D. Kapila, S. L. Bangare, N. Chandnani, and G. Saravanan, "A review paper on wireless sensor network techniques in Internet of Things (IoT)," *Materials Today: Proceedings*, vol. 51, pp. 161-165, 2022. https://doi.org/10.1016/j.matpr.2021.05.360

[4] I. H. Sarker, A. I. Khan, Y. B. Abushark, and F. Alsolami, "Internet of things (iot) security intelligence: A comprehensive overview, machine learning solutions and research directions," *Mobile Networks and Applications*, vol. 28, no. 1, pp. 296-312, 2023. https://doi.org/10.1007/s11036-022-01955-w

[5] O. F. A. Wahab, A. A. Khalaf, A. I. Hussein, and H. F. Hamed, "Hiding data using efficient combination of RSA cryptography, and compression steganography techniques," *IEEE Access*, vol. 9, pp. 31805-31815, 2021. https://doi.org/10.1109/ACCESS.2021.3060793

[6] E. Alotaibi, R. B. Sulaiman, and M. Almaiah, "Assessment of cybersecurity threats and defense mechanisms in wireless sensor networks," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 1, pp. 47-59, 2025.

[7] M. Merkepci, M. Abobala, and A. Allouf, "The applications of fusion neutrosophic number theory in public key cryptography and the improvement of RSA algorithm," *Fusion: Practice and Applications*, vol. 10, no. 2, pp. 69–74, 2023. https://doi.org/10.54216/FPA.100206

[8] Swati and R. P. Gupta, *Implementation and performance analysis of ECC-based text encryption on Raspberry Pi 3. In Artificial Intelligence and Sustainable Computing: Proceedings of ICSISCET 2020*. Cham: Springer, 2021.

[9] A. Alshuaibi, M. Almaayah, and A. Ali, "Machine learning for cybersecurity issues: A systematic review," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 1, pp. 36-46, 2025.

[10] F. Alkhudhayr, T. Moulahi, and A. Alabdulatif, "Evaluation study of elliptic curve cryptography scalar multiplication on raspberry Pi4," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 9, pp. 105–114, 2021.

[11] T. Islam, R. A. Youki, B. R. Chowdhury, and A. T. Hasan, "An ecc based secure communication protocol for resource constraints iot devices in smart home," in *Proceedings of the International Conference on Big Data, IoT, and Machine Learning: BIM , Springer*, 2021, pp. 431–444.

[12] K. Javeed, "Fpga implementation of area-time aware ecc scalar multiplication core," presented at the 2023 30th IEEE International Conference on Electronics, Circuits and Systems (ICECS), IEEE, 2023.

[13] C. Guo and B. Gong, "Efficient scalar multiplication of ECC using SMBR and fast septuple formula for IoT," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, pp. 1-14., 2021.

[14] R. S. Mousa and R. Shehab, "Applying risk analysis for determining threats and countermeasures in workstation domain," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 1, pp. 12-21, 2025.

[15] F. Herbaut, N. M´eloni, and P. V´eron, "Compact variable-base ecc scalar multiplication using euclidean addition chains," presented at the 18th International Conference on Security and Cryptography (SECRYPT), 2021.

[16] Y. Yan, "The overview of elliptic curve cryptography (ecc)," presented at the Journal of Physics: Conference Series, IOP Publishing, 2022.

[17] C. Jang, J. Han, A. M. V. V. Sai, Y. Li, and O. Yi, "A study on scalar multiplication parallel processing for X25519 decryption of 5G core network SIDF function for MMTC IoT environment," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, p. 4087816, 2022. https://doi.org/10.1155/2022/4087816

[18] S. Otoom, "Risk auditing for Digital Twins in cyber physical systems: A systematic review," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 1, pp. 22-35, 2025.

[19] O. Al-Khaleel, S. Baktir, M. Al-Khaleel, and A. Küpçü, "Efficient ecc processor designs for iot using edwards curves and exploiting fpga embedded components," *IEEE Access*, pp. 6-10, 2024.

[20] Z. Tang, C. Li, and J. Zhong, "Introducing artificial flaws in Engineered cementitious composites (ECC) through nanocomposite coating of aggregates: A novel approach and mechanism," *Cement and Concrete Composites*, vol. 157, p. 105893, 2025.

[21] D. Gu, J. Lin, B. Jiang, H. Chen, L. Xu, and J. Pan, "Shear performance of perfobond leiste (PBL) connectors in engineered cementitious composites (ECC)," *Engineering Structures*, vol. 322, p. 119093, 2025. https://doi.org/10.1016/j.engstruct.2024.119093

[22] A. Yadav, P. Sharma, and Y. Gigras, "A comparative study of elliptic curve and hyperelliptic curve cryptography methods and an overview of their applications," presented at the 2024 International Conference on Intelligent Systems for Cybersecurity, IEEE, 2024.

[23] J. Jebrane, A. Chhaybi, S. Lazaar, and A. Nitaj, "Elliptic curve cryptography with machine learning," *Cryptography*, vol. 9, no. 1, p. 3, 2024. https://doi.org/10.3390/cryptography9010003

[24] S. Majumder, S. Ray, D. Sadhukhan, M. K. Khan, and M. Dasgupta, "ECC-CoAP: Elliptic curve cryptography based constraint application protocol for internet of things," *Wireless Personal Communications*, vol. 116, no. 3, pp. 1867-1896, 2021. https://doi.org/10.1007/s11277-020-07769-2

[25] K. Javeed, A. El-Moursy, and D. Gregg, "EC-crypto: Highly efficient area-delay optimized elliptic curve cryptography processor," *IEEE Access*, vol. 11, pp. 56649-56662, 2023. https://doi.org/10.1109/ACCESS.2023.3282781

[26] R. Ifrim, D. Loghin, and D. Popescu, "A Systematic Review of Fast, Scalable, and Efficient Hardware Implementations of Elliptic Curve Cryptography for Blockchain," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 17, no. 4, pp. 1-33, 2024. https://doi.org/10.1145/3696422

[27] J. Arunkumar, S. Velmurugan, B. Chinnaiah, G. Charulatha, M. R. Prabhu, and A. P. Chakkaravarthy, "Logistic Regression with Elliptical Curve Cryptography to Establish Secure IoT," *Computer Systems Science & Engineering*, vol. 46, no. 1, pp. 1-11, 2023. https://doi.org/10.32604/csse.2023.031605

[28] M. Benssalah, I. Sarah, and K. Drouiche, "An efficient RFID authentication scheme based on elliptic curve cryptography for Internet of Things," *Wireless Personal Communications*, vol. 117, no. 3, pp. 2513-2539, 2021. https://doi.org/10.1007/s11277-020-07992-x

[29] F. Tellez and J. Ortiz, "Comparing ai algorithms for optimizing elliptic curve cryptography parameters in e-commerce integrations: A pre-quantum analysis," *arXiv preprint arXiv:2310.06752*, 2023. https://doi.org/10.14569/IJACSA.2024.0150629

[30] S. X. Pushpa and S. K. S. Raja, "Elliptic curve cryptography based authentication protocol enabled with optimized neural network based DoS mitigation," *Wireless Personal Communications*, vol. 124, no. 1, pp. 1-25, 2022. https://doi.org/10.1007/s11277-021-08902-5

[31] T. Markmann, "Modern elliptic curve cryptography for constrained devices," Project Report No. PR2, 2015.

[32] R. Azarderakhsh, K. U. Järvinen, and M. Mozaffari-Kermani, "Efficient algorithm and architecture for elliptic curve cryptography for extremely constrained secure applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 4, pp. 1144-1155, 2014. https://doi.org/10.1109/TCSI.2013.2283691

[33] M. N. Khan, A. Rao, and S. Camtepe, "Lightweight cryptographic protocols for IoT-constrained devices: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4132-4156, 2020. https://doi.org/10.1109/JIOT.2020.3026493

[34] S. Ullah and R. Zahilah, "Curve25519 based lightweight end-to-end encryption in resource constrained autonomous 8-bit IoT devices," *Cybersecurity*, vol. 4, pp. 1-13, 2021. https://doi.org/10.1186/s42400-021-00078-6

[35] B. Halak, T. Gibson, M. Henley, C.-B. Botea, B. Heath, and S. Khan, "Evaluation of performance, energy, and computation costs of quantum-attack resilient encryption algorithms for embedded devices," *IEEE Access*, vol. 12, pp. 8791-8805, 2024. https://doi.org/10.1109/ACCESS.2024.3350775

[36] S. Di Matteo, L. Baldanzi, L. Crocetti, P. Nannipieri, L. Fanucci, and S. Saponara, "Secure elliptic curve crypto-processor for real-time IoT applications," *Energies*, vol. 14, no. 15, p. 4676, 2021. https://doi.org/10.3390/en14154676

[37] I. Radhakrishnan, S. Jadon, and P. B. Honnavalli, "Efficiency and security evaluation of lightweight cryptographic algorithms for resource-constrained iot devices," *Sensors*, vol. 24, no. 12, p. 4008, 2024. https://doi.org/10.3390/s24124008

[38] Z. Liu, H. Seo, J. Großschädl, and H. Kim, "Efficient implementation of NIST-compliant elliptic curve cryptography for 8-bit AVR-based sensor nodes," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 7, pp. 1385-1397, 2015. https://doi.org/10.1109/TIFS.2015.2491261

[39] E. Gamess and S. Hernandez, "Performance evaluation of different Raspberry Pi models for a broad spectrum of interests," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 2, pp. 156-162, 2022. https://doi.org/10.14569/IJACSA.2022.0130295

[40] S. J. Johnston and S. J. Cox, *The raspberry Pi: A technology disrupter, and the enabler of dreams*. Basel, Switzerland: MDPI, 2017.

[41] D. F. Pigatto, N. B. F. da Silva, and K. R. L. J. C. Branco, "Performance evaluation and comparison of algorithms for elliptic curve cryptography with El-Gamal based on MIRACL and RELIC libraries," *Journal of Applied Computing Research*, vol. 1, no. 2, pp. 95-103, 2011.

[42] B. Bhattarai, L. Marinovici, P. S. Sarker, and A. Orrell, "MIRACL Co-Simulation platform for control and operation of distributed wind in microgrid," *IET smart grid*, vol. 5, no. 2, pp. 90-100, 2022.

[43] J. J. Menandas and M. S. Christo, "Effective implementations of scalar multiplications in elliptic curve cryptography," presented at the 2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), IEEE, 2024.

[44] Y. Hao *et al.*, "Lightweight architecture for elliptic curve scalar multiplication over prime field," *Electronics*, vol. 11, no. 14, p. 2234, 2022. https://doi.org/10.3390/electronics11142234

[45] C. D. Walter, "Fast scalar multiplication for ecc over gf (p) using division chains," presented at the Information Security Applications: 11th International Workshop, WISA 2010, Jeju Island, Korea, August 24-26, 2010, Revised Selected Papers, Springer, 2011.

[46] S. Ma, Y. Hao, Z. Pan, and H. Chen, "Fast implementation for modular inversion and scalar multiplication in the elliptic curve cryptography," presented at the 2008 Second International Symposium on Intelligent Information Technology Application, IEEE, 2008.

[47] A. M. Awaludin, H. T. Larasati, and H. Kim, "High-speed and unified ECC processor for generic Weierstrass curves over GF (p) on FPGA," *Sensors*, vol. 21, no. 4, p. 1451, 2021. https://doi.org/10.3390/s21041451

[48] M. Habek, Y. Genc, N. Aytas, A. Akkoc, E. Afacan, and E. Yazgan, "Digital image encryption using elliptic curve cryptography: A review," in *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2022: IEEE, pp. 1-8.

[49] J. Zhang, Z. Chen, M. Ma, R. Jiang, H. Li, and W. Wang, "High-performance ECC scalar multiplication architecture based on comb method and low-latency window recoding algorithm," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 32, no. 2, pp. 382-395, 2023.

[50] B. S. Alhayani *et al.*, "Optimized video internet of things using elliptic curve cryptography based encryption and decryption," *Computers and Electrical Engineering*, vol. 101, p. 108022, 2022.

[51] Q. Zhang, "An overview and analysis of hybrid encryption: The combination of symmetric encryption and asymmetric encryption," in *2021 2nd International Conference on Computing and Data Science (CDS)*, 2021: IEEE, pp. 616-622.

[52] S. Ullah, J. Zheng, N. Din, M. T. Hussain, F. Ullah, and M. Yousaf, "Elliptic Curve Cryptography; Applications, challenges, recent advances, and future trends: A comprehensive survey," *Computer Science Review*, vol. 47, p. 100530, 2023.

[53] S. Yu *et al.*, "Efficient ECC-based conditional privacy-preserving aggregation signature scheme in V2V," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 11, pp. 15028-15039, 2023. https://doi.org/10.1109/TVT.2023.3287989

[54] S. Xiao, H. Wang, and J. Zhang, "New digital signature algorithm based on ECC and its application in bitcoin and IoT," *International Journal of High Performance Systems Architecture*, vol. 10, no. 1, pp. 20-31, 2021.

[55] M. A. Dar, A. Askar, D. Alyahya, and S. A. Bhat, "Lightweight and Secure Elliptical Curve Cryptography (ECC) Key Exchange for Mobile Phones," *International Journal of Interactive Mobile Technologies*, vol. 15, no. 23, p. 26337, 2021. https://doi.org/10.3991/ijim.v15i23.26337

[56] H. Alrudainy, A. K. Marzook, M. Hussein, and R. Shafik, "Understanding power gating mechanism based on workload classification of modern heterogeneous many-core mobile platform in the dark silicon era," *Iraqi Journal for Electrical & Electronic Engineering*, vol. 20, no. 2, p. 279, 2024.

[57] S. Baghbanijam, H. Sanaei, and M. Farajzadeh, "An improved authentication & key exchange protocol based on ecdh for wsns," in *2022 30th International Conference on Electrical Engineering (ICEE)*, 2022: IEEE, pp. 563-569.

[58] İ. K. Çekiş, A. Toros, N. Apaydın, and İ. Ozcelık, "Performance comparison of ECC libraries for IoT devices," *Eskişehir Technical University Journal of Science and Technology A-Applied Sciences and Engineering*, vol. 25, no. 2, pp. 278-288, 2024. https://doi.org/10.18038/estubtda.1064044

[59] R. Alimoradi, H. R. Arkian, S.-M.-J. Razavian, and A. Ramzi, "Scalar multiplication in elliptic curve libraries," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 24, no. 3, pp. 657-666, 2021. https://doi.org/10.1080/09720529.2021.1883619

[60] Y. A. Alamari, A. Fanfakh, and E. Hadi, "Parallel Message Authentication Algorithm Implemented Over Multicore CPU," *International Journal of Intelligent Engineering & Systems*, vol. 16, no. 4, pp. 176-184, 2023. https://doi.org/10.22266/ijies2023.0831.21

[61] Z. Xia, X. Yang, A. Li, Y. Liu, and S. He, "Research on information security transmission of port multi-thread equipment based on advanced encryption standard and preprocessing optimization," *Applied Sciences*, vol. 14, no. 24, p. 11887, 2024. https://doi.org/10.3390/app142411887

[62] A. Fanfakh, N. Abduljalil, and A. K. M. Al-Qurabat, "Parallel Multi-core Implementation of the Optimized Speck Cipher," *International Journal of Safety & Security Engineering*, vol. 14, no. 3, pp. 843–852, 2024. https://doi.org/10.18280/ijsse.140316

[63] C. Rahul, N. Kousarr, T. A. Yadav, P. Keerthi, S. Hariharan, and V. Kukreja, "Analysis of resource utilization in lightweight cryptographic algorithms," in *2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC-ROBINS)*, 2024: IEEE, pp. 884-889.

[64] A. E. Adeniyi, R. G. Jimoh, and J. AWOTUNDE, "A review on elliptic curve cryptography algorithm for Internet of Things: Categorization, application areas, and security," *Application Areas, and Security*, 2024. http://dx.doi.org/10.2139/ssrn.4683742

[65] T. Fadia and L. Toufik, "Elliptic curves cryptography for lightweight devices in IoT system," *Brazilian Journal of Technology*, vol. 7, no. 4, pp. e73725-e73725, 2024. https://doi.org/10.5935/braziliantech.73725

[66] M. Rana, Q. Mamun, and R. Islam, "Balancing security and efficiency: A power consumption analysis of a lightweight block cipher," *Electronics*, vol. 13, no. 21, p. 4325, 2024. https://doi.org/10.3390/electronics13214325