Evaluation Study of Elliptic Curve Cryptography Scalar Multiplication on Raspberry Pi4

Fatimah Alkhudhayr[1], Tarek Moulahi[2]
Department of Information Technology
College of Computer, Qassim University
Buraydah, Saudi Arabia

Abdulatif Alabdulatif[3]
Department of Computer Science
College of Computer, Qassim University
Buraydah, Saudi Arabia

***Abstract*—The internet of things (IoT) is defined as a collection of autonomous devices that connect and network with each other via the Internet without the requirement for human interaction. It enhances daily our lives such as through personal devices, healthcare sensing, retail sensing, and industrial control, as well as the smart homes, smart cities, and smart supply chains. Although the IoT offers significant benefits, it has inherent issues, including security and privacy risks, memory size limitations, and processing capability challenges. This paper describes the application of elliptic curve cryptography (ECC) in a simulated IoT environment to ensure the confidentiality of data passed between the connected devices. Scalar multiplication represents the main operation of ECC, and it is primarily used for key generation, encryption, and decryption. The aim of this paper is to evaluate and show the efficiency of adapt lightweight ECC with an IoT devices. In the study outlined in this paper, scalar multiplication was implemented on Raspberry Pi4 and processing time and consumed energy were measured to compare the performance. The comparison was made on the scalar multiplication of both fast and basic ECC algorithms. The result of the performance test revealed that a fast scalar multiplication reduced the computation time in comparison with basic scalar multiplication while consuming a similar level of energy.**

***Keywords*—IoT; elliptic curve cryptography; fast scalar multiplication; raspberry Pi4**

I. Introduction

Since its conception by Kevin Ashton in 1999, the increasing popularity of the internet of things (IoT) has led to rapid changes in fields as varied as lifestyles, standards, and business models. The IoT refers to the connection of autonomous devices to the internet with the capacity to transmit data via a network without human intervention. IoT devices range from small accessories to large machines, including smartphones, tablets, laptops, personal computers, and similar portable embedded devices [1]. The IoT is not one single technology, but an agglomeration of technologies, by which embedded sensors, actuators, processors, and transceivers of connected devices comprise the IoT [2]. The communication system facilitating the communication between IoT devices can be based on sensors or wireless technologies, further enabling the transfer of data to a centralized system following processing [1].

As IoT devices imply a constant internet connection, privacy, and security issues are paramount. For example, it was demonstrated that 70 % of IoT devices are unable to resist attacks [1]. This underlines the need for security mechanisms that can ensure IoT security, e.g., in terms of access control, authentication, data integrity, confidentiality, and secrecy as well as protecting connected devices from attack.

Conventional security mechanisms and protocols designed to protect computers against cyberattacks are not appropriate for use with the IoT, primarily as the connected devices have insufficient memory size and processing capability. Hence, efficiently protecting such low-resource devices requires the consideration of other security protocols, with cryptography offering a suitable solution [3].

Cryptography refers to the encryption and decryption process, i.e., converting plain text (readable form) into ciphertext (encoded form) and vice versa, respectively using cryptographic algorithms. These algorithms can be symmetric or asymmetric: If the same key is used for both the encryption and decryption, then it is a symmetric encryption, while using a public key to encrypt and a private key to decrypt is asymmetric encryption. Cryptography strengthens computational security by making the cost of breaking the encryption exceed the value of the information that is encrypted or making the breaking time exceed the information's useful lifetime [4].

Elliptic Curve Cryptography (ECC), proposed by Miller and Koblitz in 1985, is among the most popular cryptography protocols [5]. It is like the Rivest–Shamir–Adleman (RSA) public-key cryptosystem in terms of the security level, although it has a smaller key size. The security strength of ECC relies on the Elliptic Curve Discrete Logarithm Problem (ECDLP) difficulty, which includes point doubling and adding operations, making it more computationally efficient than RSA exponentiation. Furthermore, as it consumes less memory, ECC leads to reduced performance costs and computational costs [6].

The paper explores the use of ECC to enhance data security in the IoT. It hereby aims to show that ECC is applicable for the IoT due to its efficiency and performance regarding time and energy. In achieving this aim, the following motivations are considered:

- Highlight previous studies that concern with lightweight ECC to shows the importance of an ECC and evaluate the efficient technique.

- The importance of adapting a security mechanism to the IoT environment to allow it to operate in the fastest possible way.
- The application of fast scalar multiplication to the finite field of ECC, presenting a highly suitable technique for embedded devices in the context of the IoT.
- Deduce the gap present in the literature since, few studies concern with applied fast scalar multiplication alongside with IoT devices.

The rest of this paper is organized as follows. Section II discusses similar prior works in Internet of Things' security and Elliptic Curve Cryptography. Section III describes the methodology study, while the experimental study is explained in Section IV. Section V presents the results and discusses it. Section VI concludes the paper and section VII points out our research contribution to the future work.

II. Related Work

This section mainly highlights related works on the security of Internet of Things as well as application of an Elliptic Curve Cryptography. These two subsections are considered in this paper due to its applications in many aspects in real life, in addition to its security and privacy aspects.

A. IoT

The growth of the IoT provides opportunities to enhance many aspects of our lives, such as through personal devices, healthcare sensing, retail sensing, and industrial control, as well as the smart homes, smart cities, and smart supply chains [7]. The popularity of smart home technology is made possible by the development of sensors and actuators that can be utilized in a wireless sensor network. At the same time, people have become more comfortable with and trusting toward technology, allowing companies to overcome concerns by offering benefits to the security and quality of life. Smart homes require sensors to provide intelligent services to the user. Their incorporation into domestic environments can assist with many aspects of daily living, such as by automating tasks, saving energy, and enhancing security. However, smart home technology presents issues and challenges, among which security and privacy remain the most pressing and problematic because data are recorded regarding many activities around the home. Systems must be safeguarded from attack [2].

In [8], Ayoub et al. proposed a lightweight secure scheme for IoT objects and cloud computing. Their recommendations depend on ECC and message queuing telemetry transport (MQTT), for which the key attributes are publishing and subscription. The driving factor behind their techniques is the provision of secure interactions between the IoT and cloud computing, along with enhanced communication speeds. Their security procedures consist of initialization, sub- scription, and publication, and they validated the scheme's performance by comparing it with TLS/SSL. They automatically verified the protocol's safety using the Automated Validation of Internet Security Protocols and Applications (AVISP) tool.

In [9], Sha et al. analyzed numerous issues and security challenges presented by IoT systems and found that the IoT faces more issues than wireless sensor networks (WSNs). They identified the security architecture factors of end-to- end security, edge computing-based designs, and distributed security models, discussed their benefits and restrictions, and provided examples of implementations of each design. They found that to achieve comprehensive security for IoT systems, capable low-end devices must be supported from higher up in the command structure.

In [10], Hossain et al. proposed a technique to ensure quality end-to end security for IoT systems based on biometrics and cryptography. They depicted an infrastructure of biometric-based end to end security solution for IoT with four layers: device, communication, cloud, and application. They discussed the security challenges and possible solutions for each layer and determined that their proposal based on the biometrics of facial recognition was 99 % effective by comparing face recognition in a local server and in cloud-server. Consequently, they demonstrated that to use biometrics for authentication ensures IoT system security to a greater extent than password authentication.

B. ECC

In terms of lightweight Elliptic curve cryptography various studies and experiments conducted to evaluate the performance of lightweight ECC in terms of efficiency and security with different techniques, for both hardware and software implementations. This section highlights several studies that involve for enhance scalar multiplication operation of an elliptic curve in different environments.

Firstly, number of studies have applied a parallel implantation technique for speed up scalar multiplication. In [11], Yanbo Shou et al. applied ECC cryptography to network security. To optimize the performance of scalar multiplication, which is the most expensive ECC operation, they applied it in parallel via distributed tasks that were split into neighbor nodes and executed simultaneously. Due to the required energy consumption, they found that parallel computing is only suitable when execution time is the critical factor.

Another study applied parallel technique provided by Albahri et al. [12] proposed a new algorithm that enable parallel implementation of ECC on multi-core platforms by modified algorithms that overcome data dependencies in ECC computation. Their work aims to explore the efficiency of parallel implementation of ECC as well as enhance point multiplication operation on ECC. Their proposed modifications based on two novel algorithm modifications for performing ECC point multiplication. They perform a vertical parallelization for operations of point doubling and point additions, which is they first modification. It depends on perform multiple finite field operations with no data dependency by different parallel logic cores. Their second modification to remove data dependencies by modifying the left to right double and add binary point multiplication. They implement modified algorithms with pure software implementation for ECC scalar multiplication over GF(2^{163}) using Xmos multi-core microcontroller, the result of their proposed multi-core implementation to enhance operations of point multiplication and point addition up to 60% and around 50%, respectively. Finally, their experiments show the feasibility and efficiency of adapting parallelism in ECC implementation.

In [13], Faye et al. proposed an approach to improve ECC performance for WSN capabilities, aiming to accelerate ECC scalar multiplication over primary fields, as well as avoid the storage of precomputation point by accelerating computation. They firstly proposed a new technique based on the negative of point and a point order to run fast computation of scalar multiplication. Secondly, they accelerate computation in parallel scalar multiplication on KP to avoid storage of precomputation by proposed an efficient algorithm depend on improvement of the double and add (DA) and quadrable-ana-quadrable algorithms. Finally, they showed that their proposed algorithm accelerates the computation of scalar multiplication on NIST-192 parameters for ECC. As well as they showed their technique for avoid storage computation very efficient especially for Jacobian coordinates.

As well several studies applied various technique to improve scalar multiplication operation. To confirm complete encryption for users of Online Social Networks (OSN), in [14], Rajam and Kumar proposed improved elliptic curve cryptography (IECC) to confirm complete encryption for users of online social networks (OSNs) and compared it with standard ECC. The algorithm requires the replacement of each repetitive text character with different ciphertext. Time, size, and security were used to evaluate their proposed algorithm and standard ECC based on time of key generation, time of encryption, time of decryption, and size of plaintext compared to ciphertext. They found that IECC performed better than the standard ECC.

In [15], Kalra and Dhillon conducted a comparative study to highlight viable solutions to security issues of real-time embedded systems. The characteristics of embedded systems, security issues, and threat models were presented. Public key cryptography (PKC) is the main challenge to security implementation in embedded systems; hence, efficient PKC solutions require an ECC that uses smaller key sizes. Suitable solutions suggested by this analysis of resource- constrained real-time embedded systems were all ECC-related.

In [16], Mingquan Hong et al. proposed an encryption scheme that uses ECC-based homomorphic encryption to solve the problem of secure multiparty computation (SMC). They compared their scheme's performance with RSA and Paillier homomorphic encryption, particularly in the context of computation time and communication. ECC was found better than either of these options, and they applied their scheme to GPS earthquake data to show that it is efficient and provides high security.

In [6], Arora and Chhabra proposed a security scheme to prevent eavesdropping attacks in the Cloud environment. They compared the ECC-based scheme's performance, such as the time taken to process encryption and decryption data, with that of traditional RSA schemes, and found it much faster.

In [17], Javed R. Shaikh et al. focused on implementing ECC in resource- constrained e-commerce applications. They analyzed a set of selected curves recommended by several sources to find an efficient option for constrained resources. The elliptic-curve Diffie–Hellman (ECDH) algorithm and elliptic curve digital signature algorithm (ECDSA) were applied to selected curves. They found that SECP256r1 and M221 can be used to implement ECDH and ECDSA algorithms, respectively, offering suitable curves for e-commerce applications.

Additionally, Liu et al. [18] optimized an efficient scalar multiplication algorithm for wireless sensor networks based on symmetric ternary. Their optimization depends on eliminate the modulo inversion operation in the fundamental operation by using a Jacobi coordinate. As well as a nonzero weight reduced which can reduce the operation of point addition by optimized the symmetric ternary representation of the positive integer. So that, the efficiency of the scalar multiplication algorithm is improved. The basic idea of the symmetric ternary scalar multiplication model is computing the calculation of the scalar multiplication by calling the operation of point tripling and point addition constantly. By applying symmetric ternary scalar multiplication based on Jacobi coordinate their study indicates that it is better than the scalar multiplication based on affine coordinate. They found that it has an efficiency improvement of 4.3%.

Recognizing the importance of security for mobile devices, in [19] Mullai and Mani focused on enhancing crucial aspects and optimizing the cryptography operations of RSA and ECC. To suit these two algorithms, they proposed generating addition chains (ACs) based on particle swarm optimization (PSO) and simplified swarm optimization (SSO) before measuring performance using mobile emulators of Android and Windows. The two algorithms were compared based on time of processing, power consumption for encryption and decryption, and level of security. They observed that, when considering the security aspects of SSO-optimized AC, ECC provides more security, although RSA consumes less power.

Lastly, Javeed et al. [20] proposed hardware architecture with high performance for elliptic curve scalar multiplication over prime field. It depends on parallel technique, which can either execute modular addition or modular subtraction in parallel to four modular multiplication operations. Totally, it executed five mathematic instructions in parallel. Then, presented the scalar multiplication using Jacobian coordinate. Finally, it implemented on Xilinx Virtex-6, Virtex-5 and Virtex-4 FPGA (field-programmable gate array) platforms. Their tested shows that, for one operation of elliptic curve scalar multiplication with 256-bit it takes 2.01 ms, 2.62 ms and 3.91 ms for three platforms respectively. Significantly, it is 1.96 times faster and it is applicable for any value of prime p less than 256 bits.

From above presented studies, we can observe that an improved elliptic curve cryptography has gained attention of the researchers for various platforms or environments such as WSN either for implemented in hardware or software. As well, technique applied for accelerate scalar multiplication divers some depends on parallel or sequential. Also, to the bast of our knowledge, few studies have applied a lightweight cryptography especially in term of fast scalar multiplication in an IoT devices. That encourage the needed for further study and research in this field. So, in this paper a fast scalar multiplication is applied in an IoT device with various key sizes to evaluate scalar multiplication in terms of energy and running time.

III. Contribution Theoretical Study

The following subsections explain and discuss the main ideas behind IoT scenarios in these contexts, together with ECC methods and fast scalar multiplication.

A. Architecture of IoT

The main components of IoT are presented by the three-layered architecture shown in Fig. 1. These factors can be considered as the most basic architecture [2], consisting of the three layers of perception, network, and application.

- The perception, or physical layer, is responsible for sending and receiving environmental information via sensors.
- The network layer connects with smart applications, other network de- vices, and servers. It is also used to transmit and process sensor data.
- The application layer interacts with and delivers application services to the user.

B. IoT Scenario

Smart home systems, or automated homes, allow the remote control and operation of electrical devices via electronic devices, such as smartphones or laptops, that have applications with user-friendly interfaces. A related form, the intelligent home, acts based on predefined information [21].

In our scenario, the smart home could be an IoT virtual environment, as presented in Fig. 2. Smartphone applications can interact with various applications, making them easier to control and more efficient. User registration is required for security reasons, but notifications to control and interact can be conveniently sent by email or text messages. The technology consists of wireless sensor nodes, such as for lighting, temperature, motions, and cameras that provide connections to gather and send data between users and applications via a base station that functions as a gateway. In our experiments, we used Raspberry Pi 4 for this gateway, which allowed communication with the private cloud or a specific database. For example, to check whether a light is on, the user accesses the appropriate application and chooses the light option. The gateway will allow the light sensor to send information to the user, who can then read the data, via email or text message. This is a typical example, but what if the lighting sensor reads and sends the wrong data? Such an error can cause the system to work in abnormal or malicious ways.

C. ECC Cryptosystem

Like RSA, ECC is a public key cryptosystem. However, ECC security depends on interpreting logarithm problems, e.g., how to determine K given KP and P. Table I compares key differences between ECC and RSA in terms of computational effort for cryptanalysis [4]. Algorithms of key generation, encryption, and decryption are presented in Algorithm 1, Algorithm 2 and Algorithm 3, respectively [22].

Key generation step is the most important step in which an algorithm is used to generate both public and private keys. The parameters in the key generation algorithm defined as:

- E : elliptic curve defined over finite field Fp.
- P: point on the curve that has prime order n.
- Equation of elliptic curve, prime, point on the curve with its order n are all public domain parameters denoted by E, p, P respectively.
- d: private key selected randomly from interval [1, n-1].
- Q = dP : it corresponding public key.

And parameters in the encryption and decryption algorithms identified as:

- M: point that represent a message (plain text m).
- k: randomly selected integer between range [1,n-1].
- Q: public key's recipients.
- d: private key's recipients.
- B1, B2: two points that represent cipher text.

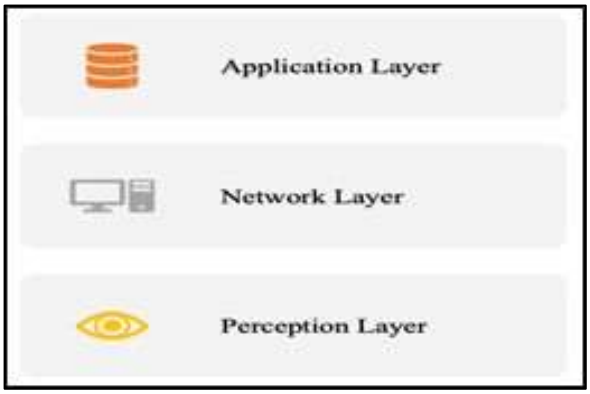

Fig. 1. IoT Architecture.

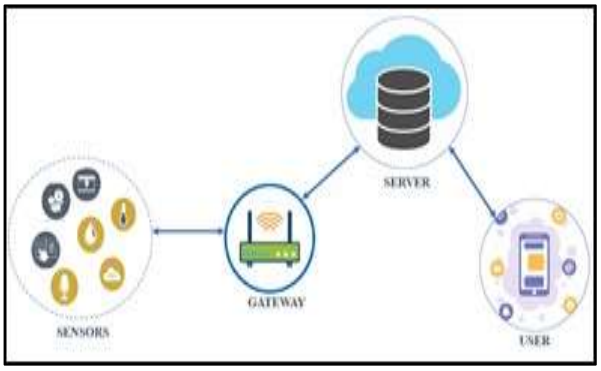

Fig. 2. Smart Home Environment.

TABLE I. Key Size Comparison RSA Vs. ECC

RSA	ECC
1024	160
2048	224
3072	256
7680	384
15360	521

Algorithm 1: Key Generation algorithm

1: begin

2: Input variables E, p, P

3: Set private Key $1 < d < n - 1$

4: Compute Public Key($Q = dp$)

5: Get (Q, d)

6: end

Algorithm 2: Encryption Algorithm

1: begin

2: Input variables E, P , p , n, m

3: Set $m \leftarrow M$

4: Set $1 < k < n - 1$

5: Compute Cipher text1(kP)

6: Compute Cipher text2(M + kQ)

7: Return (Cipher text1,Cipher text2)

8: end

Algorithm 3: Decryption Algorithm

1: begin

2: Input variables p, E, P, n, d, Cipher text1, Cipher text2

3: Compute m1(d ∗ Cipher text1)

4: Compute M (Cipher text2–m1)

5: Return M

6: end

D. Fast Scalar Multiplication

We applied the algorithm proposed by Faye et al. [13] to accelerate scalar multiplication KP over a finite field (Fp). This improvement concerns the negative of a point and a particular reduction of the scalar in a selected interval. A well-known cryptanalysis trick is to use negation, which is also used in cryptography, for scalar multiplication with addition-subtraction chains.

The algorithm of fast multiplication depends on the replacement of the KP point of the scalar multiplication operation with an equivalent representation point SP, where s and k are scalars and $k > s$. This technique is used in the interval $[\ln/2J + 1, n - 1]$, where ln/2J denotes the integer-part of n/2. The negative of a point can be used for fast computation because it is freely obtained. The point KP is replaced with an equivalent point representation SP by utilizing the negative point because, for each point P on an elliptic curve, the point - P is also on the curve. Given that point P = (xp, yp) in affine coordinates, to calculate the inverse of a point KP = (xkp, ykp), we can compute KP = (xkp, ykp) then we can change the sign on the y-coordinate (ykp).

Thus, we have the following equations for a secret key (integer number) by point KP to obtain an equivalent point representation of SP:

$$\text{If} \quad K > n, \quad Kp = Sp \quad \text{where} \quad S = (k - \lfloor k/2J.n) \tag{1}$$

$$\text{If} \quad K \in]\lfloor n/2J, n - 1], \quad kP = SP \quad \text{where } S = (k - n) \tag{2}$$

$$\text{If } K \in]0, \lfloor n/2J], \quad kP = SP \quad \text{where} \quad S = K \tag{3}$$

$$\text{If } K = n \quad \text{or } 0 \quad \text{or} - n\,, \; kP = \infty \tag{4}$$

$$\text{If} \quad k \in [-(n - 1), -\lfloor n/2J[, \; kP = SP \; \text{ where } \; S = (n + k) \tag{5}$$

$$\text{If } k \in [-\lfloor n/2J, 0[, \quad KP = SP \qquad \text{where} \quad S = K \tag{6}$$

$$\text{If } k < -n, \; KP = SP \qquad \text{where } \; S = k + n \; .\lfloor |k|/2J \tag{7}$$

IV. Experiment

An experiment was conducted to evaluate and compare the operational performance of point multiplication and fast multiplication in ECC. We explain the experimental setup and procedure.

A. Experimental Setup

The experiment was conducted in a laboratory environment, as shown in Fig. 3. The C programming language was used to develop a program that ran on a Raspberry Pi, and was used to simulate an IoT environment. The program performed a scalar multiplication KP for both a basic and a lightweight ECC. Subsequently, it measured the computational time of the scalar multiplication via a built-in library in C with a time() function. The consumed energy was computed using a digital voltage power meter. The time and energy data were recorded to evaluate and compare the performance of both the basic and fast scalar multiplication algorithms.

B. Raspberry pi

An IoT environment was created with a Raspberry Pi 4 Model B, which is the newest and fastest Raspberry product. It comes with various amounts of RAM (2, 4, or 8 GB), has a USB-C port for power, requires a MicroSD card to store all files and the operating system, has two micro HDMI ports, and offers the user the choice to connect to the internet via Ethernet cable or wirelessly. Table II shows the characteristics of the Raspberry Pi 4 Model B used in this experiment.

Fig. 3. Experimental Setup.

TABLE II. Raspberry Pi 4 Model B Specifications

Operating System	Raspbian OS
Card size	8 GB
Internet Connectivity	Wi-Fi

C. Experimental Procedure

A controlled laboratory environment was used as the setting of the experiment to avoid a disruption of the internet connection. Three main key sizes (K) were tested to evaluate the performance of the scalar multiplication KP. The key sizes were 32, 64, and 128 bits, representing the first, second, and third cases, respectively. For all three cases, the time was measured in seconds, and the energy consumption was computed in Watts.

V. Result and Discussion

The experimental analysis compared the performance of the standard scalar multiplication KP with that of the fast scalar multiplication SP based on the evaluation criteria of time and energy (in seconds and Watts, respectively). The evaluation was verified for three key sizes (32, 64, 128 bits). To compare basic and fast scalar multiplication in terms of time and energy consumption, the experiment was conducted multiple times for each key size using random numeric data, and the ratio for each key size was calculated.

The following subsections will discuss the result according to two aspects time and energy.

A. Time Consumption

Following Fig. 4, 5, and 6 present the time consumption for each key size. For the 32-bit key size, the times for both the basic and fast algorithms were similar at the beginning; however, at the end, the basic algorithm consumed more time, increasing by approximately 90 % compared with fast scalar algorithm, as shown in Fig.4. Second, for both the 64- and 128-bit key sizes, the fast algorithm consumed less than one second, while the basic algorithm consumed more time, as shown in Fig. 5 and Fig. 6, respectively.

Finally, as a result, at almost a half-second, fast scalar multiplication used less time than basic scalar multiplication for all three key sizes. The basic scalar multiplication needed more time for the three key sizes of 32, 64, and 128 bits, at 9.1, 98.7, and 216.1 seconds, respectively. As shown in Fig. 7, fast scalar multiplication running faster than basic scalar multiplication approximately 99%. In addition, the running time demonstrated a direct relationship with the received data, i.e., more time for more data.

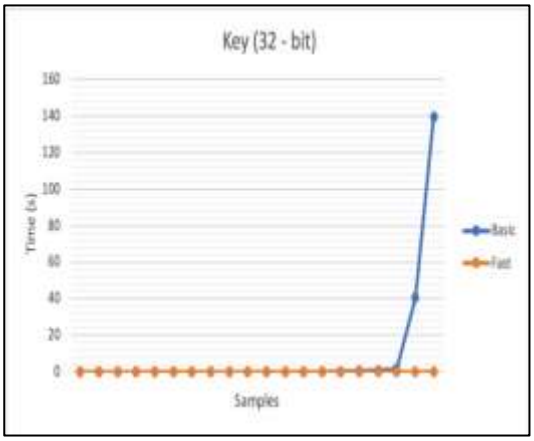

Fig. 4. Basic Running Time vs. Fast (Case1).

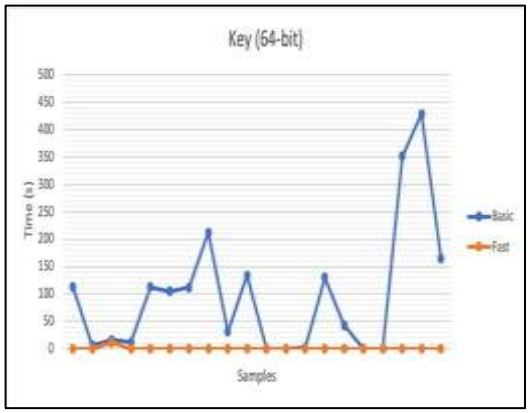

Fig. 5. Basic Running Time vs. Fast (Case2).

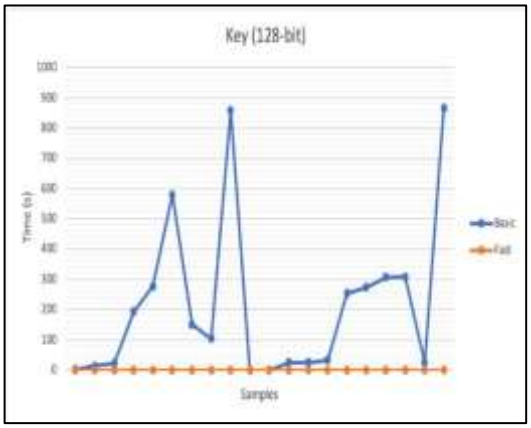

Fig. 6. Basic Running Time vs. Fast (Case3).

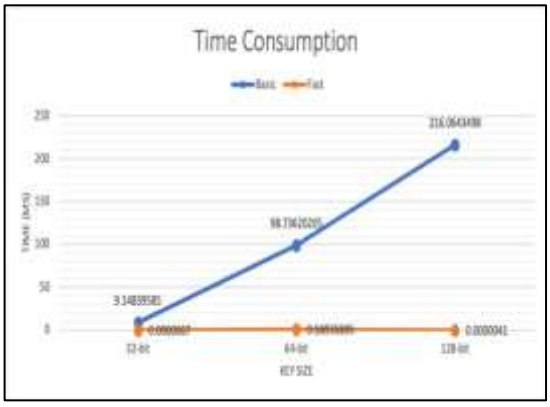

Fig. 7. Running Time Average for all Three Cases.

B. Energy Consumption

The energy consumption for each key size is shown in the following Fig. 8, 9, and 10. The basic algorithm consumed more energy than the fast algorithm when the key size was 32 bits, as shown in Fig. 8. For key sizes of 64 and 128 bits, as shown in Fig. 9 and Fig. 10, respectively, similar amounts of energy were consumed by fast and basic scalar multiplication.

Finally, as shown in Fig. 11, the fast algorithm uses less energy than the basic algorithm, by about 10 % in the case of the smaller key size. For the larger key sizes, the basic algorithm used about 0.5 % less energy than the fast algorithm.

To conclude, the fast scalar multiplication algorithm saves more time than the basic scalar multiplication algorithm. However, basic scalar multiplication and fast scalar multiplication consumed similar amounts of energy.

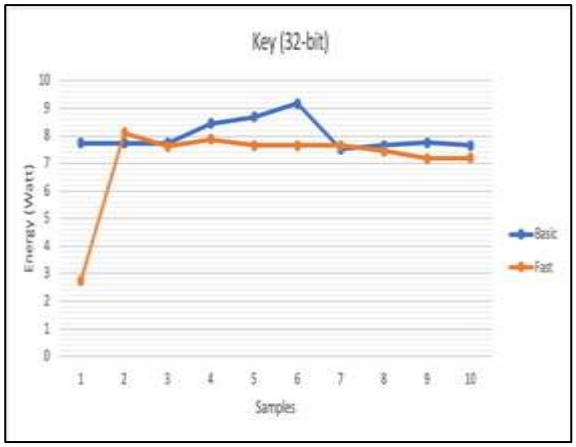

Fig. 8. Basic Energy Consumption vs. Fast (Case1).

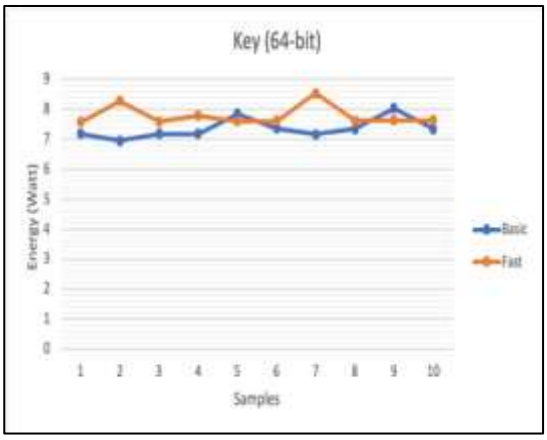

Fig. 9. Basic Energy Consumption vs. Fast (Case2).

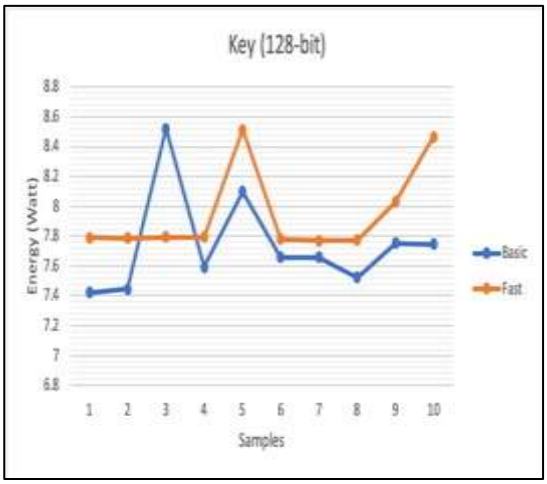

Fig. 10. Basic Energy Consumption vs. Fast (Case3).

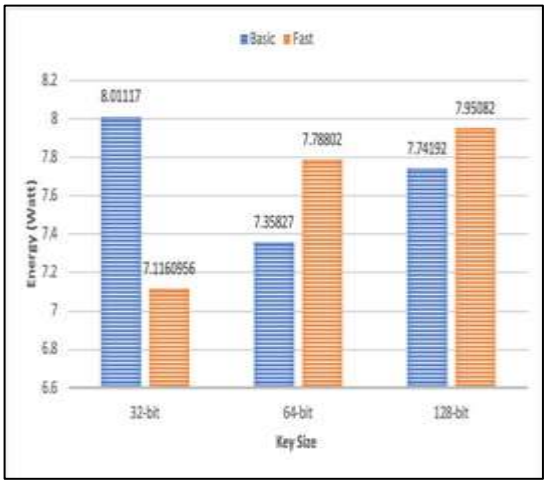

Fig. 11. Energy Consumption Average for all Three Cases.

VI. Conclusion

The increasing integration of IoT technology in various domains of daily life presents challenges regarding security, privacy, and cost of performance. Against this backdrop, this study focused on ECC in general, and on scalar multiplication methods in particulars. We presented various studies examining security solutions in the context of the IoT as well as enhancing the performance of ECC. As a time- and cost-effective security mechanism is crucial, our experiment examined and compared the performance of basic and fast scalar multiplication to show the efficiency and applicable of fast scalar multiplication in the embedded devices. The results revealed that fast scalar multiplication saves time for three key sizes with similar energy usage. The main contribution is that we demonstrated the fast scalar multiplication is faster than basic scalar multiplication around 99%. That indicates fast scalar multiplication is a suitable solution for embedded devices to reduce the cost of performance.

VII. Future Work

In future work, we plan to reduce the energy consumption by optimizing fast scalar multiplication. Furthermore, additional lightweight ECC can be examined and evaluated in terms of their performance. Moreover, other evaluation metrics can be adopted. Additionally, homomorphic encryption can be applied and compared with obtained result. Finally, studying and applying this technique to other devices offers an interesting research avenue.

References

[1] M. Abdur Razzaq, M. Ali Qureshi, S. Habib Gill, and S. Ullah, "Security Issues in the Internet of Things (IoT): A Comprehensive Study," Int. J. Adv. Comput. Sci. Appl., vol. 8, no. 6, pp. 383–388, 2017, doi: 10.14569/IJACSA.2017.080650.

[2] P. Sethi and S. Sarangi, "Internet of Things: Architectures, Protocols, and Applications," J. Electr. Comput. Eng., 2017, doi: 10.1155/2017/9324035.

[3] "Internet of Things for Telecom Engineers." [Online]. Available: http://forms1.ieee.org/rs/682-UPB-550/images/IEEE-IOT-White-Paper.pdf.

[4] W. Stallings, Cryptography and Network Security Prin- Ciples and Practice. Pearson Education, 2011.

[5] T. Daisy Premila Bai, K. Michael Raj, and S. Albert Rabara, "Elliptic Curve Cryptography Based Security Framework for Internet of Things (IoT) Enabled Smart Card," Proc. - 2nd World Congr. Comput. Commun. Technol. WCCCT 2017, pp. 43–46, 2017, doi: 10.1109/WCCCT.2016.20.

[6] A. Chhabra and S. Arora, "An Elliptic Curve Cryptography Based Encryption Scheme for Securing the Cloud against Eavesdropping Attacks," Proc. - 2017 IEEE 3rd Int. Conf. Collab. Internet Comput. CIC 2017, vol. 2017-Janua, pp. 243–246, 2017, doi: 10.1109/CIC.2017.00040.

[7] K. M. (Mat), "Capitalizing on the business value of the internet of things: The time to act is now," 2015.

[8] A. Amrani and N. A. Rafalia, "Lightweight Secure Scheme for IoT-Cloud Convergence based on Elliptic Curve," vol. 96, no. 1, pp. 144–155, 2019.

[9] K. Sha, W. Wei, T. Andrew Yang, Z. Wang, and W. Shi, "On security challenges and open issues in Internet of Things," Futur. Gener. Comput. Syst., vol. 83, pp. 326–337, 2018, doi: 10.1016/j.future.2018.01.059.

[10] S. M. M. R. M. Shamim Hossain, Ghulam Muhammad and A. A. Wadood Abdul, Abdulhameed Alelaiwi, "Toward End-to-End Biomet rics-Based Security for IoT Infrastructure," 2016.

[11] Y. Shou, H. Guyennet, and M. Lehsaini, "Parallel scalar multiplication on elliptic curves in wireless sensor networks," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7730 LNCS, pp. 300–314, 2013, doi: 10.1007/978-3-642-35668-1_21.

[12] M. S. Albahri, M. Benaissa, and Z. U. A. Khan, "Parallel Implementation of ECC Point Multiplication on a Homogeneous Multi-Core Microcontroller," Proc. - 12th Int. Conf. Mob. Ad-Hoc Sens. Networks, MSN 2016, no. 1, pp. 386–389, 2017, doi: 10.1109/MSN.2016.070.

[13] Y. Faye, H. Guyennet, S. Yanbo, and I. Niang, "Accelerated precomputation points–based scalar reduction on elliptic curve cryptography for wireless sensor networks," Int. J. Commun. Syst., vol. 30, no. 16, pp. 1–11, 2017, doi: 10.1002/dac.3327.

[14] S. Thiraviya Regina Rajam and S. Britto Ramesh Kumar, "Enhanced elliptic curve cryptography," Indian J. Sci. Technol., vol. 8, no. 26, 2015, doi: 10.17485/ijst/2015/v8i26/80444.

[15] P. K. Dhillon and S. Kalra, "Elliptic Curve Cryptography for Real Time Embedded Systems in IoT Networks," 2016 5th Int. Conf. Wirel. Networks Embed. Syst., pp. 1–6, 2016.

[16] M. Q. Hong, P. Y. Wang, and W. B. Zhao, "Homomorphic Encryption Scheme Based on Elliptic Curve Cryptography for Privacy Protection of Cloud Computing," in Proceedings - 2nd IEEE International Conference on Big Data Security on Cloud, IEEE BigDataSecurity 2016, 2nd IEEE International Conference on High Performance and Smart Computing, IEEE HPSC 2016 and IEEE International Conference on Intelligent Data and S, 2016, no. September, pp. 152–157, doi: 10.1109/BigDataSecurity-HPSC-IDS.2016.51.

[17] J. R. Shaikh, M. Nenova, G. Iliev, and Z. Valkova-Jarvis, "Analysis of standard elliptic curves for the implementation of elliptic curve cryptography in resource-constrained E-commerce applications," 2017 IEEE Int. Conf. Microwaves, Antennas, Commun. Electron. Syst. COMCAS 2017, vol. 2017-Novem, pp. 1–4, 2017, doi: 10.1109/COMCAS.2017.8244805.

[18] H. Liu, Q. Dong, and Y. Li, "Efficient ECC scalar multiplication algorithm based on symmetric ternary in wireless sensor networks," Prog. Electromagn. Res. Symp., vol. 2017-Novem, pp. 879–885, 2017, doi: 10.1109/PIERS-FALL.2017.8293258.

[19] A. Mullai and K. Mani, "Enhancing the security in RSA and elliptic curve cryptography based on addition chain using simplified Swarm Optimization and Particle Swarm Optimization for mobile devices," Int. J. Inf. Technol., vol. 13, no. 2, pp. 551–564, 2021, doi: 10.1007/s41870-019-00413-8.

[20] K. Javeed, X. Wang, and M. Scott, "High performance hardware support for elliptic curve cryptography over general prime field," Microprocess. Microsyst., vol. 51, pp. 331–342, 2017, doi: 10.1016/j.micpro.2016.12.005.

[21] M. Hasan, P. Biswas, M. T. I. Bilash, and M. A. Z. Dipto, "Smart home systems: Overview and comparative analysis," Proc. - 2018 4th IEEE Int. Conf. Res. Comput. Intell. Commun. Networks, ICRCICN 2018, pp. 264–268, 2018, doi: 10.1109/ICRCICN.2018.8718722.

[22] N. York, B. Heidelberg, H. Kong, L. Milan, and P. Tokyo, Guide to Elliptic Curv e Cryptography Springer. Springer-Verlag New York, Inc., 2004.