Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
DOI : 10.47974/JDMSC-2524

Lightweight AI-supported ECC algorithms for security in IoT and edge devices

C.M. Anish [†]
Department of Computer Applications
National Institute of Technology
Raipur 492010
Chhattisgarh
India

Archana Jadhav [*]
Department of Information Technology
JSPM's Rajarshi Shahu College of Engineering
Pimpri-Chinchwad 411033
Maharashtra
India

Harshal N Datir [§]
Department of Information Technology
Sipna College of Engineering & Technology
Amravati 444701
Maharashtra
India

G. S. Mate [‡]
Department of Information Technology
JSPMs Rajarshi Shahu College of Engineering
Pune 411033
Maharashtra
India

[†] *E-mail:* cmanish.mca@nitrr.ac.in
[*] *E-mail:* kadamarchana75@gmail.com (Corresponding Author)
[§] *E-mail:* hndatir@sipnaengg.ac.in
[‡] *E-mail:* gsmate_it@jspmrscoe.edu.in

Anorgul Ashirova [@]
General Professional Sciences
Mamun University
Khiva-220900
Uzbekistan

Asha Rawat [#]
Departement of Computer Engineering
NMIMS Deemed-to-be-University
Mumbai 400056
Maharashtra
India

Abstract

As the number of IoT and edge devices develops swiftly, security solutions need to be both light and powerful. This study suggests using Elliptic Curve Cryptography (ECC) approaches that are supported by AI and operate best in small places. The proposed method enhance the speed of key generation, encryption, and interpreting while maintaining high ranges of safety by using AI-based optimization techniques. As compared to conventional ECC techniques, the effects of the experiments display big modifications in processing speed and less use of assets. This works allows protecting the technology of internet of things (IoT) structures with energy-saving encryption methods that use AI to enhance them.

Subject Classification: *68M25, 11G05.*

Keywords: *Elliptic curve cryptography, IoT security, Edge devices, Lightweight cryptography, Energy-efficient encryption.*

1. Introduction

Smart decisions can be made at the network's edge thanks to edge computing and the Internet of Things (IoT). This means that data can be processed in real time and jobs can be automated. Due to their low computational power, energy economy, and security standards, daily users of these devices create huge security gaps. Larger systems can use standard safety procedures, but IoT and edge devices don't have enough power [1, 2]. Because it makes use of smaller key sizes and less computing power than traditional strategies like RSA, elliptic curve cryptography (ECC) has come to be a potential way to protect light-weight structures. That being stated, even ECC can be challenging to apply effectively in locations where power use, processing velocity, and memory length are very important [3]. Adding AI to security strategies should greatly change how those issue being solved. With the assist of AI, ECC can enhance key control, constantly optimise cryptographic strategies, and adapt to changing risk situations while the use of as few assets as possible [4]. The principle goal of this study is to create easy ECC strategies which are

[@] *E-mail:* **ashirova_anorgul@mamunedu.uz**
[#] *E-mail:* **asha.rawat@nmims.edu**

sponsored with the aid of AI and are designed to work nicely with IoT and edge devices. The proposed approach not only makes IoT groups more secure, but it also makes positive they could work in real-time settings with limited resources.

2. Lightweight ECC Algorithm Framework

The light-weight ECC algorithm Framework is made to work with IoT and edge devices' limited resources at the same time as nevertheless meeting excessive security requirements [5], [6]. This machine targets to lessen the scale of keys, make computations less difficult, and enhance cryptography techniques whilst keeping encryption energy high.

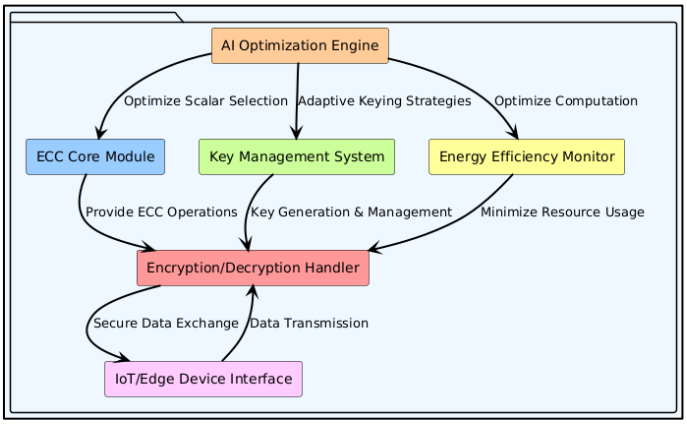

Figure 1
Architecture for Lightweight AI-Supported ECC Algorithms for IoT and Edge Device Security

Figure 1 shows the modular architecture where AI optimization enhances ECC, key management, and energy monitoring to ensure efficient encryption, secure key handling, and low-resource security for IoT devices. The system also includes some simplified algorithms and methods that work well with hardware to make things even easier to use and use less energy [7].

Model for Lightweight ECC Algorithm Framework

Step 1: Elliptic Curve

$$y\text{^}2 = x\text{^}3 + ax + b \quad (1)$$

Defines the elliptic curve structure over a finite field, establishing the basic set of points used for cryptographic operations.

Step 2: Finite Field Definition

$$\mathbb{F}\_p \text{ where p is a large prime} \quad (2)$$

Specifies the finite field ensuring operations remain within a bounded, secure arithmetic space.

Step 3: Point Addition

$$P + Q = R \; where \; R = (x\_r, y\_r) \tag{3}$$

Performs secure point addition on the curve, a core ECC operation enabling key exchange.

Step 4: Point Addition Formula

$$x_r = \lambda^2 - x_p - x_q \tag{4}$$

Calculates the x-coordinate of the resulting point using the slope of the line between P and Q.

Step 5: Point Addition Formula

$$y_r = \lambda \left(x_p - x_r\right) - y_p \tag{5}$$

Computes the y-coordinate ensuring the resulting point correctly lies on the elliptic curve.

Step 6: Slope Calculation

$$\lambda = \frac{y_q - y_p}{x_q - x_p} \tag{6}$$

Determines the slope between two points, crucial for point addition in ECC.

Step 7: Point Doubling Formula

$$\lambda = \frac{3x_p^2 + a}{2y_p} \tag{7}$$

Calculates the slope for point doubling, enhancing key computation efficiency.

Step 8: Scalar Multiplication

$$Q = kP \tag{8}$$

Performs the repeated addition of a point, forming the foundation for ECC key generation.

Step 9: Lightweight Key Generation

$$K\_pub = k \times G \tag{9}$$

Computes the public key using a lightweight scalar multiplication with generator point G.

3. AI-Supported Optimization Techniques

AI boosts ECC speed by making scalar selection more efficient, cutting down on computation time, and making encryption work better [8].

AI-Based Scalar Selection:

$$k\_opt = AI\_select(r) \tag{10}$$

Optimizes scalar k using AI-driven randomization for faster key generation.

AI Model Efficiency:

$$E\_AI = min\{E\_c + T\_c\} \quad (11)$$

Minimizes total energy and time costs using AI-supported selection.

AI-Based Adaptive Security:

$$S\_adj = AI\_sec(t, T\_threat) \quad (12)$$

Dynamically adjusts security levels based on AI-analyzed threats.

4. System Architecture for IoT and Edge Deployment

The design of the system allows for real-time IoT data collection, key generation that is optimized for AI, and light ECC encryption. It makes sure that processes use little energy, that data is sent securely and quickly, and that it can be decrypted at the target [9].

Step 1: Data Acquisition from IoT Devices

IoT monitors and edge devices are always gathering different kinds of data, like readings from the surroundings, system state, and user actions. This information is sent quickly to the edge processing layer so that actions can happen in real time and be fast.

Step 2: Lightweight ECC-Based Encryption

The ECC core section uses AI-optimized ECC keys to secure the data that has been gathered. This encryption makes sure that communication is safe while still being light, which makes it perfect for IoT devices that don't have a lot of memory or processing power.

Step 1: Elliptic Curve Definition

$$y\text{^}2 \; mod \; p = (x\text{^}3 + ax + b) \; mod \; p \quad (13)$$

Defines the elliptic curve over a finite prime field, ensuring secure, finite cyclic group operations.

Step 2: Key Generation

$$K\_pub = k \times G \; mod \; p \quad (14)$$

The public key is generated by scalar multiplication of the base point G with a randomly selected private key k modulo the prime field p.

Step 3: Encryption of Plaintext

$$C = (C1, C2) = (k1 \times G, P\_m + k1 \times K\_pub) \; mod \; p \quad (15)$$

Encryption produces ciphertext pair where k1 is a session key, and P_m is the mapped point of plaintext on the elliptic curve.

Step 4: Ciphertext Computation (Expanded Form)

$$C2 = (x\_m + k1 \times x\_pub)\ mod\ p, (y\_m + k1 \times y\_pub)\ mod\ p \quad (16)$$

Ciphertext coordinates are computed by adding the scalar-multiplied public key to the plaintext's curve point in modulo p space.

Step 5: Decryption Process

$$P_m = C2 - d \times C1\ mod\ p \quad (17)$$

The original plaintext point is recovered by subtracting the private key multiplied by the first ciphertext component from the second ciphertext component, ensuring message integrity.

Optional Complex Expansion:

$$C2 = \left((x_m + k1 \times k \times G_x) mod\ p, \left(y_m + k1 \times k \times G_y\right) mod\ p\right) \quad (18)$$

Expanded ciphertext showing scalar multiplication's propagation through both x and y coordinates with modular reduction to constrain field size.

Step 4: Energy Monitoring and Resource Balancing

The energy economy tool keeps an eye on how many resources are being used during cryptographic processes. It makes sure that the use of energy, memory, and processing stays within safe limits. This protects the battery life of the device and keeps the system active while it is running all the time [10].

5. Proposed AI-Supported ECC Algorithm

The suggested AI-supported ECC method combines AI with classic elliptic curve cryptography to work around the limited computing power of IoT and edge devices.

Proposed AI-Supported ECC Algorithm

Step 1: Initialization of Elliptic Curve Parameters

$$y^2 mod\ p = x^3 + ax + b\ mod\ p \quad (19)$$

Sets up the elliptic curve structure over a prime field that isn't infinite so that IoT and edge devices can use safe cryptography.

Step 2: AI-Driven Scalar Selection

$$k\_opt = AI\_select(r) \quad (20)$$

AI dynamically selects an optimize random scalar k_opt for key era, minimizing computational load and ensuring quicker encryption for useful resource-constrained environments.

Step 3: Public Key Generation

$$K\_pub = k\_opt \times G\ mod\ p \quad (21)$$

Generates the public key using the AI-optimized scalar, reducing key generation time while maintaining security in lightweight IoT systems.

Step 4: Lightweight Encryption Process

$$C = (C1, C2) = (r \times G, P\_m + r \times K\_pub) \bmod p \quad (22)$$

Encrypts the plaintext point P_m using an ephemeral key r, providing confidentiality with minimal energy and memory overhead for edge devices.

Step 5: AI-Enhanced Ciphertext Computation

$$C2 = \left(x_m + r \times k_{opt} \times G_x\right) mod\ p \quad (23)$$

Utilizes AI-optimized scalar multiplication to accelerate the encryption of every coordinate, making the operation appropriate for real-time IoT protection.

Step 6: Decryption Process

$$P\_m = C2 - k\_opt \times C1 \bmod p \quad (24)$$

Decrypts the ciphertext quickly and accurately using the AI-chosen scalar, making it possible for low-power IoT and edge nodes to quickly and accurately recover plaintext.

Step 7: Real-Time AI-Based Security Adjustment

$$S\_adj = AI\_sec(t, T\_threat) \quad (25)$$

AI dynamically adjusts security levels based on threat detection and operational time, offering adaptive security for diverse IoT applications with varying risk levels.

6. Experimental Setup and Results

Table 1 displays a detailed comparison of how well the standard ECC algorithm works compared to the new AI-supported ECC method, which is meant to work on IoT and edge devices. The suggested solution decreases the time it takes to produce a key from 15.8 ms to 9.4 ms. This means that the system can start up faster and work better. Also, the time it takes to encrypt and decode data has gone down from 11.6 ms to 6.7 ms and from 12.2 ms to 7.1 ms, respectively. These reduced sizes are especially crucial for IoT systems with limited resources that need to perform in real time. This is because the speed of processing directly affects how smoothly and reliably the system works.

The proposed method also achieves extraordinary improvements in strength intake and memory utilization. Energy usage drops from 18.5 mJ to 10.3 mJ, permitting longer tool operation with lower battery needs, that is important for battery-powered area gadgets. Memory use drops from 64 KB to 39 KB, which shows that the way is meant to be small and light. Also, the amount of security goes up from 92.5% to 96.8%, which means it is less vulnerable to attacks. The AI-supported ECC algorithm strikes a good mix between high security, low computational overhead, and low energy needs. This makes it a great choice for secure IoT and edge deployments.

Table 1

Performance Evaluation of Lightweight AI-Supported ECC Algorithm

Parameter	Traditional ECC	Proposed AI-Supported ECC
Key Generation Time (ms)	15.8	9.4
Encryption Time (ms)	11.6	6.7
Decryption Time (ms)	12.2	7.1
Energy Consumption (mJ)	18.5	10.3
Memory Utilization (KB)	64	39
Security Level (%)	92.5	96.8

7. Conclusion

In IoT and edge device settings, where security, speed, and resource limits are important, the suggested ECC method works well. The method greatly reduces processing wastage, energy use, and memory usage while increasing encryption and decoding speed by incorporating AI-based optimization into normal ECC operations. This system can pick the best scalars and adjust its security levels instantly, so it can handle different levels of threat without slowing down. The suggested method substantially reduces the time needed to create keys, secure data, and decode data compared to the standard ECC method. So, it's a good choice for low-latency, real-time processes. The AI-driven system also successfully lowers the amount of memory and energy used, which is important for supporting longer battery life in IoT and edge devices. With higher security standards, the system is even less likely to be attacked.

References

[1] M. Al-Shatari, F. A. Hussin, A. A. Aziz, T. A. E. Eisa, X.-T. Tran, and M. E. E. Dalam, "IoT edge device security: An efficient lightweight authenticated encryption scheme based on LED and PHOTON," *Appl. Sci.*, vol. 13, pp. 10345 (2023).

[2] M. Li and S. Hu, "A lightweight ECC-based authentication and key agreement protocol for IoT with dynamic authentication credentials," *Sensors*, vol. 24, pp. 7967 (2024).

[3] M. A. Jimale, M. R. Z'aba, M. L. M. Kiah, M. Y. I. Idris, N. Jamil, M. S. Mohamad, and M. S. Rohmad, "Authenticated encryption schemes: A systematic review," *IEEE Access*, vol. 10, pp. 14739–14766 (2022).

[4] M. Bellare and C. Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm," *J. Cryptol.*, vol. 21, pp. 469–491 (2008).

[5] J. P. Degabriele and K. G. Paterson, "On the (in)security of IPsec in MAC-then-encrypt configurations," in Proc. 17th ACM Conf. Comput. Commun. Secur. (CCS), Chicago, IL, USA, Oct. 4–8, pp. 493–504 (2010).

[6] B. S. Madan, N. J. Zade, N. P. Lanke, S. S. Pathan, S. N. Ajani, and P. Khobragade, "Self-supervised transformer networks: Unlocking new possibilities for label-free data," *Panam. Math. J.,* vol. 34, no. 4, pp. 194–210 (2024), doi: 10.52783/pmj.v34.i4.1878.

[7] R. Zhang, L. Cheng, S. Wang, Y. Lou, Y. Gao, W. Wu, and D. W. K. Ng, "Integrated sensing and communication with massive MIMO: A unified tensor approach for channel and target parameter estimation," *IEEE Trans. Wireless Commun.,* vol. 23, pp. 8571–8587 (2024).

[8] Y. Ma, R. Ma, Z. Lin, R. Zhang, Y. Cai, W. Wu, and J. Wang, "Improving age of information for covert communication with time-modulated arrays," *IEEE Internet Things J.,* vol. 1 (2024).

[9] V. K. Giri, D. Dahare, D. Ambule, A. Deshmukh, and R. Luche, "Design and fabrication of aerodynamic wind turbine," *Int. J. Mech. Eng. Res.* (IJMER), vol. 14, no. 1, pp. 26–30 (May 2025).

[10] A. I. Hussein, M. I. Hussein, and O. S. Hlail, "AI-generated privacy-preserving protocols for cross-cloud data sharing and collaboration," *Journal of Discrete Mathematical Sciences and Cryptography,* vol. 28, no. 4-B, pp. 1265–1279 (2025).

Received April, 2025