ADSUJSR 13 (2) (2025) 163 - 174
Adamawa State University Journal of Scientific Research (ADSUJSR)
ISSN: 2705-1900 (Online); ISSN: 2251-0702 (Print)
https://www.adsujsr.adsu.edu.ng

ELLIPTIC CURVE CRYPTO-SYSTEM USING BABY-STEP GIANT-STEP ALGORITHM

R. F. NDATUWONG and D. SAMAILA*
Department of Mathematics, Adamawa State University Mubi, Nigeria
**Corresponding author email: samaila506@adsu.edu.ng*

Received in November 2025
Accepted in December 2025

Abstract

Although the Elgamal Elliptic Curve Cryptosystem (EECC) has proven to be the most outstanding cryptographic system, the increasing complexity of cryptographic attacks necessitates the development and analysis of more secure cryptographic systems. One way to achieve that is through the exploration of other approaches to EECC security system. Hence, the topic- Baby-Step Giant-Step Algorithm using group theoretic approach. As most used cryptographic systems assume that the discrete logarithm is extremely difficult to compute, the more difficult it is, the more security it provides a data transfer. The algorithm we presented in this paper is based on a space-time trade-off. It is proven to be a simple modification of trial multiplication, the naive method of finding discrete logarithms. The results show that the best way to speed up the baby-step giant-step algorithm is to use an efficient table lookup scheme. Most especially, the hashing is done on the second component in the pair (j, αj). Hence, the baby-step giant-step algorithm could be used by an eavesdropper to derive the private key generated in Diffie-Hellman key exchange, when the modulus is a prime number p that is not too large.

Keywords: ElGamal elliptic curve, Discrete logarithm, Encryption, Decryption, Key exchange, Cyclic group, Baby-step Giant-step algorithm

Introduction

Communication security is one of many informatics parts with huge progress. Sensitive data are increasingly used in communication and that is the reason why security requirement is timelier and more important. The risk of obtaining data through an attack increases with enhancing the power of today's computers (Aranha et al, 2020).

Cryptography has been an essential aspect of information security, safeguarding data against unauthorized access and ensuring the integrity and confidentiality of communications. Cryptographic systems face constant threats from advancing computational capabilities and emerging cryptographic attacks. Over the years, various cryptographic techniques have been developed, some of which are Rivest – Shamir – Adleman (RSA), Digital Signature Algorithm (DSA), ElGamal Cryptosystem (EC), ElGamal Elliptic Curve Cryptosystem (EECC) etc, among which the Elgamal Elliptic Curve Cryptosystem (EECC) stands out due to its robustness and efficiency.

Elliptic Curve Cryptography (ECC), on which EECC is based, utilizes the mathematical properties of elliptic curves to provide security (Koblitz, 2019). ECC is a relatively recent development in the field of cryptography, introduced independently by Neal Koblitz and Victor Miller in the mid-1980s (Miller, 1986). EECC offers enhanced security with shorter key lengths, making them resistant to many current and foreseeable cryptographic attacks (Johnson & Menezes, 2019).

The use of elliptic curves over finite fields has proven to be advantageous due to the smaller key sizes required for equivalent security levels compared to traditional systems like RSA and DSA. This results in faster computations and reduced storage requirements, which are critical in resource-constrained environments such as mobile devices and embedded systems (Hankerson, Menezes, & Vanstone, 2020). The elliptic curve discrete logarithm problem (ECDLP) forms the basis of ECC's security, requiring exponential time to solve on classical computers. ECC has gained widespread adoption in various applications, including secure communication protocols and digital signatures (Menezes, et al, 1997).

The Elgamal cryptosystem, introduced by Taher Elgamal in 1985, is a public-key cryptosystem based on the Diffie-Hellman key exchange. It allows for both encryption and digital signatures, making it versatile for various cryptographic applications. When combined with ECC, the Elgamal cryptosystem benefits from the security and efficiency of elliptic curves, leading to the development of EECC.

One of the key advantages of ECC is its ability to provide high levels of security with smaller key sizes. For instance, a 256-bit key in ECC is as secure as a 3072-bit key in RSA (Bos & Costello, 2014). This efficiency is particularly important in environments where computational power and storage are limited. Consequently, EECC is well-suited for applications in modern communication systems, including secure messaging, financial transactions, and digital signatures (Blake, et al, 2018). The Elgamal Elliptic Curve Cryptosystem has been extensively studied in theoretical contexts, but using group theoretic approach provides an opportunity for further research. The theoretical underpinnings of EECC provide insights into its potential vulnerabilities and the measures needed to mitigate them (Biehl, et al, 2021). For instance, differential fault analysis and side-channel attacks are notable threats that require careful consideration in both the design and implementation stages (Lange & Shparlinski, 2015).

The adoption of EECC in various industries underscores its relevance and practicality. Financial institutions, for example, leverage ECC-based cryptographic systems to secure transactions and protect sensitive information. The efficiency of EECC also makes it an attractive option for securing communications in the Internet of Things (IoT) ecosystem, where devices are often resource-constrained (Silverman, 2016).

The ongoing research in elliptic curve cryptography is driven by the need to stay ahead of potential threats and improve the security of cryptographic systems. Innovations in mathematical algorithms and computational techniques continue to enhance the performance and security of EECC. For instance, advances in fast arithmetic operations on elliptic curves have led to more efficient implementations of EECC, making it viable for a wide range of applications (Hankerson, et al, 2020).

EECC has proven to be the most outstanding cryptographic system, there is still need to improve it security. One way to do that is through the exploration of other approaches to EECC security system. Hence, the paper is channeled towards the baby-step giant-step algorithm using group theoretical approach.

Group theory principles have been integrated into the ElGamal cryptosystem to enhance its security and efficiency. By leveraging the algebraic structures of groups, researchers can develop more robust cryptographic protocols and algorithms. In the context of ElGamal cryptosystem, group theory principles are applied to analyze the mathematical properties of the underlying multiplicative group modulo, a large prime number ‘p’. These insights aid in optimizing key generation, encryption, and decryption operations, thereby strengthening the security of the ElGamal cryptosystem.

Integrating group theory principles into the ElGamal cryptosystem may introduce new security implications and vulnerabilities. While group theoretic approaches can enhance the security of cryptographic systems by leveraging algebraic properties, they also require careful analysis to ensure that the introduced modifications do not weaken the overall security posture. Security implications introduced by group theoretic approaches need to be thoroughly evaluated to ascertain their impact on the cryptographic system's resilience against attacks (Smith, 2019).

In the context of elliptic curve cryptography (ECC), the discrete logarithm problem (DLP) remains a fundamental security assumption. ECC relies on the difficulty of solving the DLP over elliptic curve groups, which are inherently different from traditional multiplicative groups modulo prime numbers. Group theory provides a framework for analyzing the computational complexity of the DLP in the context of elliptic curves and offers insights into the security implications of using elliptic curve groups in cryptographic protocols.

Efficient algorithms play a crucial role in cryptographic systems, influencing factors such as performance, scalability, and resource utilization. In practical applications, cryptographic algorithms need to operate efficiently to meet the demands of real-time communication, large-scale data processing, and resource-constrained environments. Efficient algorithms not only enhance the user experience by minimizing computation time and resource overhead but also enable the deployment of cryptographic solutions in diverse computing environments, including embedded systems, mobile devices, and cloud computing platforms (Bernstein et al, 2017).

Group theory provides a theoretical foundation for understanding the structure and properties of mathematical objects, which can be exploited to design more efficient cryptographic algorithms. Algorithmic optimizations based on group theoretic insights can lead to significant improvements in key generation, encryption, decryption, and other cryptographic operations (Gupta, et al, 2020).

The theoretical exploration of EECC involves a detailed analysis of its mathematical properties, security features, and potential vulnerabilities. This includes studying the algebraic structure of elliptic curves, the computational complexity of ECDLP, and the impact of various cryptographic attacks. Such an in-depth analysis provides valuable insights for developing more secure and efficient cryptographic systems (Johnson & Menezes, 2019). Understanding the theoretical aspects of EECC not only enhances its practical implementations but also contributes to the broader field of cryptography. Researchers continue to explore new mathematical techniques and cryptographic protocols based on elliptic curves, driving innovation and improving security (Silverman, 2016).

The importance of secure communication and data protection cannot be overstated in today's digital age. As cyber threats continue to evolve, the need for robust cryptographic systems like EECC becomes increasingly critical. The theoretical exploration of EECC not only advances our understanding of elliptic curve cryptography but also contributes to the development of secure cryptographic protocols that can withstand emerging threats (Hankerson, et al, 2020). Continued research and development in this area are essential for enhancing the security and performance of cryptographic systems in an increasingly digital world (Koblitz, 2019; Stallings, 2017).

ElGamal Elliptic Curve

The ElGamal elliptic curve cryptosystem (ECC) represents a significant advancement in the field of cryptography, combining the robustness of the ElGamal encryption scheme with the efficiency and security of elliptic curve mathematics. Elliptic Curve Cryptography (ECC) has gained widespread recognition for its ability to provide equivalent security with smaller key sizes compared to traditional cryptosystems like RSA, making it particularly advantageous in environments with limited computational resources (Sun, 2018).

The ElGamal cryptosystem, initially proposed by Taher ElGamal in 1985, is a public key cryptosystem that leverages the difficulty of the discrete logarithm problem for secure communications (Zhang & Liu, 2017). When integrated with ECC, the resulting ElGamal ECC benefits from the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP), thus enhancing security.

This chapter provides a detailed literature review of the ElGamal elliptic curve cryptosystem, encompassing both theoretical insights and empirical findings. By synthesizing recent research, it aims to present a holistic understanding of the ElGamal ECC's capabilities, challenges, and future directions. The integration of group theoretical concepts with empirical evidence underscores the cryptosystem's potential to address the evolving security needs of the digital age.

Theoretical Framework

A thorough literature review was conducted to establish a theoretical framework for understanding the ElGamal ECC. Key sources included seminal papers, textbooks, and recent research articles on elliptic curves, cryptography, and the ElGamal encryption scheme. This review helped identify critical aspects of the ElGamal ECC, such as its theoretical and mathematical foundations, security assumptions, and other important concepts that will help in understanding and fulfilling the research objectives.

Overview of Cryptographic Systems

Cryptography is the science of securing communication in the presence of adversaries. It provides techniques to ensure confidentiality, integrity, authenticity, and non-repudiation of data. Cryptographic systems are broadly classified into symmetric-key cryptography and asymmetric-key cryptography.

Symmetric-Key Cryptography: In symmetric systems, both the sender and receiver share the same secret key for encryption and decryption. Example: Advanced Encryption Standard (AES). Limitation: The need to securely share the key makes it impractical for large-scale systems (Meneze et al, 1997).

Asymmetric-Key Cryptography: Introduced by Diffie and Hellman in 1976, asymmetric cryptography uses two keys: a public key for encryption and a private key for decryption. It eliminates the need for secure key exchange. Example: RSA, ElGamal, and Elliptic Curve Cryptography (ECC). Asymmetric systems, though computationally expensive, are essential for modern secure communication.

Key Concepts in Public-Key Cryptography

Public-key cryptography relies on mathematical problems that are computationally hard to solve (Koblitz, 1987; Miller, 1985). The three most widely used problems are:

i. **Integer Factorization Problem (IFP):** Used in RSA, this problem involves factoring a large composite number into its prime factors. It is computationally infeasible for large numbers. Example: Given $n = pq$, where p and q are large primes, finding p and q is hard.

ii. **Discrete Logarithm Problem (DLP):** Used in Diffie-Hellman and ElGamal cryptosystems. Given $g^x \bmod p = hq$, finding x is computationally infeasible when p is large.

iii. **Elliptic Curve Discrete Logarithm Problem (ECDLP):** A variant of DLP, where the problem is defined over the group of points on an elliptic curve. This is the foundation of ECC.

Overview of the ElGamal Cryptosystem

The ElGamal cryptosystem is a widely used public-key encryption scheme that provides both encryption and digital signature functionalities. It is based on the hardness of the Discrete Logarithm Problem (DLP) in a cyclic group. The ElGamal cryptosystem can be adapted to different algebraic structures, including finite fields and elliptic curve groups, making it a versatile and foundational cryptosystem in modern cryptography.

Key Components of the ElGamal Cryptosystem

The ElGamal cryptosystem consists of the following components:

i. **Cyclic Group**: A group G with a generator g, where the group operation is either multiplication (in finite fields) or elliptic curve addition (in elliptic curve groups).

ii. **Private Key**: A randomly chosen integer x, where $x \in \{1, 2, \ldots, |G| - 1\}$, serves as the private key.

iii. **Public Key**: The public key is derived from the private key as $h = g^x$, where g is the generator.

The cryptosystem operates securely under the assumption that the Discrete Logarithm Problem (DLP) or Elliptic Curve Discrete Logarithm Problem (ECDLP) is computationally infeasible in the chosen group.

Review of Existing Literatures on Key Generation in ElGamal ECC

Empirical research has focused on optimizing key generation techniques in ElGamal ECC. Studies have explored various methods for efficient elliptic curve point multiplication, crucial for generating secure keys. For instance, Smith and Brown (2019) examined the performance of different key generation algorithms, highlighting improvements in computational efficiency and security.

Recent empirical research on the ElGamal elliptic curve cryptosystem (ECC) has focused extensively on its performance and security metrics.

Ali, Khan, and Hussain (2018) conducted a study to evaluate the security of ElGamal ECC implementations against various cryptographic attacks. Their research highlighted the robustness of the cryptosystem, particularly when large key sizes were employed, making it resistant to common attack vectors.

Smith and Brown (2019) investigated the efficiency of different key generation techniques in ElGamal ECC. Their study compared several algorithms for elliptic curve point multiplication, which is central to the key generation process. The results demonstrated that optimized algorithms significantly reduce computation time without compromising security, suggesting practical improvements for real-world applications.

The application of ElGamal ECC in secure communications was empirically tested by Nguyen and Othman (2019). They implemented the cryptosystem in a simulated network environment to assess its performance in encrypting and decrypting data. The findings confirmed that ElGamal ECC provides strong encryption capabilities with minimal overhead, making it suitable for high-security communication channels.

In another empirical study, Patel and Modi (2021) conducted a comprehensive analysis of the ElGamal ECC's resilience against side-channel attacks. They tested various implementation strategies to determine which provided the best protection against these attacks. Their results indicated that certain coding practices and hardware configurations could enhance the cryptosystem's security against physical attack vectors.

Gupta and Yadav (2020) explored the practical implementation of ElGamal ECC in mobile devices. Their research focused on optimizing the cryptosystem for constrained environments with limited computational resources. The study demonstrated that, with appropriate optimizations, ElGamal ECC could be efficiently implemented on mobile platforms without significant performance degradation.

Wang, Zhang, and Liu (2016) examined the use of precomputation techniques to improve the efficiency of ElGamal ECC operations. Their empirical results showed that precomputation could reduce the time required for encryption and decryption processes, making the cryptosystem more suitable for applications requiring fast cryptographic operations.

The integration of ElGamal ECC in blockchain technology was empirically tested by Li and Chen (2020). They implemented the cryptosystem within a blockchain framework to evaluate its effectiveness in securing transactions. The study found that ElGamal ECC provided strong security guarantees while maintaining the performance requirements of the blockchain environment.

Sun (2018) conducted an empirical study on the scalability of ElGamal ECC for large-scale applications. The research focused on the cryptosystem's ability to handle increased loads without performance bottlenecks. The findings suggested that ElGamal ECC could scale effectively, provided that sufficient computational resources were available.

In a study by Chen (2015), the focus was on the empirical analysis of ElGamal ECC's resistance to quantum attacks. Although the cryptosystem showed some vulnerability to potential quantum computing advancements, the research highlighted ongoing efforts to develop quantum-resistant variants of ElGamal ECC, which could mitigate these risks in the future.

Zhang and Liu (2017) explored the empirical performance of ElGamal ECC in cloud computing environments. Their research aimed to determine the cryptosystem's suitability for securing data in cloud storage and processing. The results indicated that ElGamal ECC could provide robust security for cloud applications, although optimization for specific cloud architectures was necessary. An empirical evaluation of ElGamal ECC's was also used in secure voting systems. They implemented the cryptosystem in a prototype electronic voting system and assessed its ability to ensure vote integrity and confidentiality. The study concluded that ElGamal ECC could effectively secure electronic voting processes, enhancing the trustworthiness of election results.

The performance of ElGamal ECC in securing Internet of Things (IoT) devices was empirically analyzed by Smith and Brown (2019). Their research focused on the cryptosystem's ability to provide lightweight security for resource-constrained IoT devices. The findings demonstrated that, with appropriate adjustments, ElGamal ECC could be a viable security solution for the IoT ecosystem.

Ali, Khan, and Hussain (2018) extended their research to evaluate the cross-platform performance of ElGamal ECC. They implemented the cryptosystem on various operating systems and hardware configurations to assess its versatility. The results indicated that ElGamal ECC performed consistently well across different platforms, highlighting its adaptability.

Patel and Modi (2021) examined the energy efficiency of ElGamal ECC in battery-powered devices. Their empirical study measured the cryptosystem's energy consumption during key cryptographic operations. The findings suggested that, while ElGamal ECC required significant energy, optimization techniques could reduce power consumption, making it more suitable for energy-constrained environments.

Finally, Gupta and Yadav (2020) investigated the use of ElGamal ECC in securing smart grid communications. Their empirical research focused on the cryptosystem's ability to protect data transmitted between smart grid components. The study concluded that ElGamal ECC provided strong encryption and was capable of securing smart grid infrastructure against cyber threats.

An empirical study by Liu, et al, 2020, explored the implementation of ElGamal ECC in securing wireless sensor networks (WSNs). Their research focused on evaluating the cryptosystem's performance in terms of energy consumption, latency, and security. The results indicated that ElGamal ECC could effectively secure WSNs with manageable energy consumption and acceptable latency, providing a feasible security solution for these networks.

Liu, et al, 2020, also conducted an empirical analysis of ElGamal ECC in the context of secure multimedia transmissions. They implemented the cryptosystem to encrypt video and audio streams, assessing its impact on quality and performance. Their findings showed that ElGamal ECC maintained high levels of security while introducing minimal latency and degradation in multimedia quality, making it suitable for secure streaming applications.

In an empirical evaluation, Robinson et al. (2018) assessed the application of ElGamal ECC for securing electronic health records (EHRs). They implemented the cryptosystem in a simulated healthcare environment to test its effectiveness in protecting patient data. The study concluded that ElGamal ECC provided robust encryption for EHRs, ensuring patient privacy and data integrity without significant performance overhead.

Another study by Ahmed and Mustafa (2020) examined the use of ElGamal ECC in securing vehicular ad hoc networks (VANETs). Their research focused on evaluating the cryptosystem's ability to protect communications between vehicles and infrastructure. The empirical results demonstrated that ElGamal ECC could secure VANET communications effectively, offering a viable solution for enhancing the security of intelligent transportation systems.

Finally, an empirical study by Patel et al, 2021, investigated the performance of ElGamal ECC in securing cloud-based e-commerce platforms. They implemented the cryptosystem to protect sensitive transaction data in a cloud environment. The study found that ElGamal ECC provided strong security for e-commerce transactions, with minimal impact on system performance, ensuring safe and efficient online commerce operations.

METHODOLOGY

The general structure of the ElGamal Elliptic Curve Cryptosystem (EECC), which include; the algebraic structures in ECC, the integration of group theory in EECC and the EECC algorithms using group theoretic approach were considered and presented in this section.

The Elliptic Curve Group

An elliptic curve E over a prime field F_p is defined by the equation: $E: y^2 \equiv x^3 + ax + b(\mod p)$, with $p \neq 2,3,\ a,b \in F_p$, and must satisfy the condition $4a^3 + 27b^2 \neq 0(\mod p)$. The set of points (x, y) that satisfy this equation, along with a special point at infinity (O), form an elliptic curve group $E(F_p)$ (Koblitz, 1987).

ElGamal Elliptic Curve Key Exchange Protocol

The Elliptic Curve Diffie-Hellman (ECDH) is a key exchange protocol that allows two parties to securely share a common secret over an insecure channel as follows:

Algorithm:

Setup:

- Choose an elliptic curve E defined over a finite field F_p.
- Select a base point G on the curve with a large prime order n.

Key Generation:

- Alice chooses a private key d_A randomly from $\{1,2,...,n-1\}$ and computes her public key: $P_A = d_A G$.
- Bob chooses a private key d_B randomly from $\{1,2,...,n-1\}$ and computes his public key: $P_B = d_B G$.

Key Exchange:

- Alice sends P_A to Bob, and Bob sends P_B.

Shared Secret Computation:

- Alice computes the shared secret: $S = d_A.P_B = d_A(d_B G)$.
- Bob computes the shared secret: $S = d_B.P_A = d_B(d_A.G)$.

Since elliptic curve scalar multiplication is commutative, both will compute the same shared secret.

ElGamal Elliptic Curve Encryption Scheme

The ElGamal encryption scheme is a public key cryptosystem which relies on the Diffie-Hellman key exchange for its security. The basic steps for ElGamal encryption over elliptic curves are:

Key Generation:

a. Choose an elliptic curve (E) defined over a finite field F_q.

b. Select a base point (P) on the curve.

c. Generate a private key (d), which is a random integer.

d. Compute the public key ($Q = dP$), where (dP) denotes scalar multiplication of point (P) by (d).

Encryption

a. To encrypt a message (m) (represented as a point (M) on the elliptic curve):

b. Choose a random integer (k).

c. Compute ($C_1 = kP$).

d. Compute ($C_2 = M + kQ$).

e. The cipher text is the pair (C_1, C_2).

Decryption

a. To decrypt the ciphertext (C_1, C_2).

b. Compute ($S = dC_1 = dkP$).

c. Compute ($M = C_2 - S$).

ElGamal Elliptic Curve Signature Scheme

In addition to encryption, the ElGamal cryptosystem can be used for digital signatures. The signing process involves generating a pair of values that prove the authenticity of a message, while the verification process ensures the signature is valid.

Signing:

- Alice wishes to sign a message m
- She selects a random $k \in \{1,2,...,|G|-1\} \ni \gcd(k,|G|) = 1$
- She computes:

$$r = g^k \bmod p$$

$$s = (m - xr)k^{-1} \bmod (p-1)$$

where k^{-1} is the modular inverse of $k \bmod (p-1)$

- The signature is the pair (r, s)

Verification:

- To verify the signature (r, s) on the message m, Bob computes:

$$v_1 = g^m \bmod p$$

$$v_2 = r^s \bullet h^r \bmod p$$

- If $v_1 = v_2$, the signature is valid.

Elliptic Curve Discrete Logarithm Problem (ECDLP)

The security of ECC is based on the intractability of the Elliptic Curve Discrete Logarithm Problem (ECDLP), which is defined as follows:

- Given two points P and $Q = kP$ on an elliptic curve $E(F_q)$, find the scalar k.

The hardness of ECDLP ensures the security of ElGamal ECC.

The best-known algorithms for solving the ECDLP, such as Pollard's rho method or index calculus, have a sub-exponential time complexity of $O(\sqrt{n})$, where n is the size of the group. This makes ECC highly secure even with smaller key sizes compared to other cryptosystems like RSA. ECC offers equivalent security with much smaller key sizes: A 256-bit ECC key provides the same security as a 3072-bit RSA key. This efficiency makes ECC ideal for resource-constrained environments like IoT devices and mobile systems.

Discrete Logarithm Assumption

The Discrete Logarithm Assumption states that the ECDLP is computationally infeasible in the group of points on an elliptic curve over a finite field (Koblitz,1987). Formally:

- Given $P, Q \in E(F_q)$, it is computationally infeasible to determine k such that Q = kP, where k is uniformly distributed in [0, n-1] and n is the group order.

This assumption is essential for algorithms like Elliptic Curve Diffie-Hellman (ECDH): A secure key exchange protocol and Elliptic Curve Digital Signature Algorithm (ECDSA): A widely used signature algorithm.

Solving the Elliptic Curve Discrete Logarithm Problem (ECDLP)

The procedures for solving two logarithmic problems are presented in this section.

a. The discrete logarithm problem (DLP) is defined as follows:

➢ Given a prime p, a generator g of a cyclic group Z_{p^*}, and an element $h \in Z_{p^*}$, find an integer x (if it exists) such that:

$$g^x \equiv h \bmod p.$$

b. The elliptic curve discrete logarithm problem (ECDLP) is defined as follows:

- Given an elliptic curve E over a finite field $E(F_p)$, a point $P \in E(F_p)$, and another point $Q \in E(F_p)$, find an integer k (if it exists) such that: Q = kP.

The implementation of these procedures is therefore presented in the next section.

RESULTS AND DISCUSSION

Presenting the Baby-Step Giant-Step Method

One important branch of group theory in mathematics is the baby-step giant-step that meet in the middle algorithm for computing some discrete algorithm or order of an element in a finite Abelian group. The discrete logarithm problem plays a very important role in public key cryptography.

Most used cryptographic systems assume that the discrete log is extremely difficult to compute; the more difficult it is, the more security it provides a data transfer. One way to increase the difficulty of the discrete log problem is to base the cryptosystem on a larger group. The algorithm we presented in this paper is based on a space-time tradeoff. It is a simple modification of trial multiplication, the naive method of finding discrete logarithms.

Now, given a cyclic group G of order *m* and generator *α*, let *β* be an arbitrary element of the group. The problem here is to find an integer *n* such that

$$\alpha^n = \beta$$

The baby-step giant-step method is based on rewriting *n* as follows:

$$n = ix + j \quad \text{where } 0 \le i \le x,\ 0 \le j \le x$$

$$n = [\sqrt{m}\,]$$

Therefore, we have:

$\alpha^n = \beta$ which implies that $\alpha^{ix+j} = \beta$ so that $\alpha^j = \beta(\alpha^{-x})^i$.

The algorithm is to precompute α^j for all *j*, then fixes the integer *x* and try the values of *i* in the right-hand side of the congruence, in the manner of trial multiplication. It tests to see if the congruence is satisfied for all values of *j*, using the precomputed values of α^j.

The Algorithm

Input: A cyclic group *G* of order *m*, with a generator *α* and an element *β*.
Output: A value *n* satisfying $\alpha^n = \beta$

- x ← Ceilling ($\sqrt{m}$)
- For all *j*, 0 ≤ *j* < *x*:
  - Compute α^j and keep the pair (j, α^j) in a table
- Compute α^{-x}
- Set $\gamma \leftarrow \beta$
- For all *i*, 0 ≤ *i* < *x*:
  - Check if γ is the second component (α^j) of any pair in the table
  - If yes, return $ix + j$.
  - Otherwise, $\gamma \leftarrow \gamma \bullet \alpha^{-x}$.

Implementing the Baby-Step Giant-Step Algorithm

In this case, we take *p* as large integer, and use the Baby-Step Giant-Step Algorithm as follows:

i. Precompute Baby Steps:

Set $m = \sqrt{p-1} = \sqrt{16} = 4.$

Compute $g^j \bmod 17$ for j = 0, 1, 2, 3:

Table 1: Baby Step Method for Solving DLP.

j	$g^j \bmod 17$
0	1
1	3
2	9
3	27mod17=10

ii. Compute Giant Steps:Now compute

$h \bullet g^{-im} \bmod p$ for i = 0, 1, 2, …

We first compute $g^{-m} \bmod 17$:

$$g^{-m} \equiv g^{p-1-m} \bmod 17 = 3^{12} \bmod 17 = 15.$$

Compute giant steps $h \bullet g^{-im} \bmod 17$:

Table 2: Giant Step Method for Solving DLP.

i	$h \bullet g^{-im} \bmod 17$
0	13
1	$13 \bullet 15 \bmod 17 = 8$
2	$8 \bullet 15 \bmod 17 = 1$

Match baby steps with giant steps: $g^3 \equiv h \bullet g^{-2m}$, so x = 4.

Solving the Elliptic Curve Discrete Logarithm Problem (ECDLP)

Let: $E: y^2 \equiv x^3 + x + 1 \bmod 23, P = (3,10)$ and $Q = (9,7)$. We need to find k such that $Q = kP$.

Step 1: Verify Points

Check that P = (3, 10) and Q = (9, 7) lie on the curve $y^2 \equiv x^3 + x + 1 \bmod 23$:

i. For P = (3, 10):

$$y^2 \equiv 10^2 \equiv 100 \bmod 23 = 8,$$

$$x^3 + x + 1 \equiv 3^3 + 3 + 1 \equiv 27 + 3 + 1 \bmod 23 = 31 \bmod 23.$$

So P lies on the curve.

ii. For Q = (9, 7):

$$y^2 \equiv 7^2 \equiv 49 \bmod 23 = 3,$$

$$x^3 + x + 1 \equiv 9^3 + 9 + 1 \equiv 729 + 9 + 1 \bmod 23 = 3.$$

So Q lies on the curve.

Step 2: Compute Multiples of P:

Using the elliptic curve point addition formulas, compute successive multiples of P until Q is reached.

Compute Multiples of P = (3, 10):

i. 1P = (3, 10).

ii. 2P = P + P:

$$\lambda = \frac{3(3^2)+1}{2(10)} \bmod 23 = \frac{28}{20} \bmod 23.$$

The modular inverse of 20 modulo 23 is 7, so:

$$\lambda = 5 \bullet 7 \bmod 23 = 35 \bmod 23 = 12.$$

Compute x_3, y_3:

$$x_3 = 12^2 - 2(3) \bmod 23 = 144 - 6 \bmod 23 = 15.$$

$$y_3 = 12(3-15) - 10 \bmod 23 = 12(-12) - 10 \bmod 23 = -154\mathrm{m}$$

So, 2P = (15, 17).

iii. 3P = 2P + P = (15, 17) + (3, 10):

We follow the point addition formula to compute 3P.

Continue this process until Q = (9, 7) is reached.

Equivalently, one can use the point addition formula:

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \bmod(p).$$

For G + G, where P = Q = (0, 1), use point doubling:

$$\lambda = \frac{3x_1^{\ 2} + a}{2y_1} \bmod(p),$$

$$\lambda = \frac{3(0)^2 + 1}{2(1)} \bmod(13) = 7.$$

Compute

$$x_3 = \lambda^2 - 2x_1 \bmod(p), y_3 = \lambda(x_1 - x_3) - y_1 \bmod(p).$$

Repeat this process to generate the cyclic subgroup.

Step 3: Find k:

After successive computations, we find that Q = 5P.

Therefore, k = 5.

- For DLP, techniques like trial and error or the Baby-Step Giant-Step Algorithm can solve the problem efficiently.

- For ECDLP, successive computation of multiples of a point using elliptic curve group operations is the standard approach.

These examples demonstrate the computational complexity of both DLP and ECDLP, which forms the basis of their cryptographic security.

Conclusion

The results in this paper show that the best way to speed up the baby-step giant-step algorithm is to use an efficient table lookup scheme. Most especially, the hashing is done on the second component in the pair (j, α^j), and to perform the check in step 1 of the main loop, γ is hashed and the resulting memory address checked. This seems to be better since hash tables are known for retrieving and adding elements in O(1) time (constant time). This does not slow down the overall baby-step giant-step algorithm.

It is also deduced that the baby-step giant-step algorithm could be used by an eavesdropper to derive the private key generated in Diffie-Hellman key exchange, when the modulus is a prime number *p* that is not too large. The result also shows that the baby-step giant-step algorithm is a generic algorithm that can work for every finite cyclic group G. It is not necessary to know the exact order of group *G* in advance. The algorithm still works if *n* is merely an upper bound on the group order. Hence, we conclude that the algorithm is used better for groups whose order is a prime.

Conflicting Interest

The authors have declared no conflict of interest regarding the manuscript

References

Ahmed, M., & Mustafa, S. (2020). Secure Communications in Vehicular Ad Hoc Networks Using Elliptic Curve Cryptography. *Journal of Network and Computer Applications*, 149, 102481

Ali, A., Khan, M., & Hussain, S. (2018). Improved Security Analysis of Elliptic Curve Cryptography. *Journal of Information Security and Applications*, 41, 112-123.

Aranha, D. F., Dahab, R., López, J., & Oliveira, L. B. (2020). Efficient implementation of elliptic curve cryptography. *Journal of Cryptographic Engineering*, 10(1), 1-20.

Bernstein, D. J., Lange, T., & Rezaeian Farashahi, R. (2017). *Post-quantum cryptography*: Supersingular isogeny-based cryptosystems. Springer.

Biehl, I., Meyer, B., & Müller, V. (2021). Differential fault attacks on elliptic curve cryptosystems. *Advances in Cryptology—CRYPTO 2000*, 131-146.

Blake, I., Seroussi, G., & Smart, N. (2018). Elliptic Curves in Cryptography. *Cambridge University Press.*

Bos, J. W., Costello, C., Naehrig, M., & Stebila, D. (2014). Elliptic curve cryptography in practice. *Advances in Cryptology – Public Key Cryptography.*

Chen, J. (2015). Elliptic Curve Cryptography and Its Applications. *International Journal of Advanced Computer Science and Applications,* 6(7), 45-51.

Gupta, S., & Yadav, R. (2020). Analysis of Elliptic Curve Cryptography and Its Implementations. *International Journal of Computer Applications*, 177(13), 1-5.

Hankerson, D., Menezes, A., & Vanstone, S. (2020). *Guide to Elliptic Curve Cryptography*. Springer Science & Business Media.

Johnson, D. B., & Menezes, A. J. (2019). *Elliptic curve DSA (ECDSA):* An enhanced DSA. Digital Signature Standard (DSS).

Koblitz, N. (2019). Elliptic Curve Cryptosystems. *Mathematics of Computation*, 48(177), 203–209.

Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177), 203–209.

Lange, T., &Shparlinski, I. E. (2015). Applications of computational number theory to elliptic curve cryptography. *Computational Number Theory and Modern Cryptography*, 103-128

Li, W., & Chen, Y. (2020). Blockchain-Based Secure Data Sharing Using Elliptic Curve Cryptography. *IEEE Access*, 8, 19175-19183.

Liu, Y., Zhang, J., & Wang, X. (2020). Advancements in Elliptic Curve Cryptography. *Journal of Cryptography and Information Security*, 12(2), 214-231.

Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (1997). *Handbook of Applied Cryptography*. CRC Press.

Miller, V. S. (1985). Use of Elliptic Curves in Cryptography. *Advances in Cryptology—* CRYPTO 1985. Nguyen, H., & Othman, M. (2019). Elliptic Curve Cryptography for Blockchain Technology. *Journal of Network and Computer Applications*, 140, 102-408.

Patel, V., & Modi, N. (2021). Elliptic Curve Cryptography: Security Analysis and Performance Enhancement. *Journal of Cryptographic Engineering*, 11(2), 125-138.

Robinson, L., Thompson, A., & Miller, D. (2018). Protecting Electronic Health Records with ElGamal Elliptic Curve Cryptosystem. *Journal of Medical Systems*, 42(12), 233.

Silverman, J. H. (2016). *The Arithmetic of Elliptic Curves*. Springer Science & Business Media

Smith, R., & Brown, T. (2019). Key Generation Techniques in Elliptic Curve Cryptography. *Cryptography and Information Security*, 5(2), 67-75.

Stallings, W. (2017). *Cryptography and Network Security: Principles and Practice*. Pearson.

Sun, Y. (2018). Elliptic Curve Discrete Logarithm Problem and Its Applications. *Cryptologia*, 42(4), 319

-335.

Wang, H., Zhang, J., & Liu, K. (2016). Efficient Algorithms for Elliptic Curve Cryptography. *Journal of Cryptology*, 29(2), 235-264.

Zhang, X., & Liu, J. (2017). ElGamal Cryptosystem with Elliptic Curves. *Journal of Information Security*, 8(2), 123-134.