Mathematical Journal of Modelling and Forecasting
Vol. 3, No. 2, December 2025, pp. 18-25

Implementation of Elgamal Elliptic Curve-Based Cryptographic Systems (ECEG) in Text Encryption

Kenanga Dylla Oktariza[1,*] , Meira Parma Dewi[2]

[1,2]Departement of Mathematics, Universitas Negeri Padang, Indonesia
[*]oktarizakenangadylla@gmail.com

Abstract. Modern cryptography continues to evolve to meet the challenges of information security in the digital age. One of the most widely used algorithms is ElGamal's Elliptic Curve (ECEG). This public-key cryptography-based encryption method offers a high level of security with better computational efficiency than classical algorithms. This study implements the ECEG algorithm in the text encoding process to ensure data confidentiality and integrity. The encryption process is performed using the recipient's public key, while decryption is performed using the corresponding private key. The entire encryption and decryption process in this algorithm can be understood mathematically through operations on elliptical curves. The purpose of this study is to determine the peculiarities of each elliptical curve selected on an ECEG. This study uses a curves $y^2 = (x^3 + 8x + 25) \bmod 37$ generates all the points on the curve that can be used as a generator that will represent all the letters in the alphabet and numbers from 0 to 9.

Keywords: Cryptographic, Elgamal, Elliptic Curve, Text Encryption.

1 Introduction

Cryptography is a branch of mathematical science that maintains the confidentiality of information through the processes of encryption and decryption. Cryptography comes from the Greek: "cryptós" means "secret", while "gráphein" means "writing". So, cryptography means "secret writing"[1]. In its application, cryptography uses mathematical concepts in the creation and application of keys. Plaintext is readable information, while ciphertext is an encrypted form of that information [2]. The terms used in this study are encryption, decryption, key, plaintext, and ciphertext. Encryption is the process of encoding a native message into a message that cannot be interpreted as the original [3]. Keys are parameters used for enciphering and deciphering transformations [4]. Plaintext is a message or information that will be delivered in an easy-to-read format or in its original form. *Ciphertext* is information that has been encrypted [5]. In the realm of modern cryptography, ECEG is considered to have a high level of security due to the complexity of its mathematical calculations In the sphere of contemporary cryptography, ECEG is deemed to have a high level of security due to the complexity of its mathematical calculations [6]. Demonstrates the effectiveness of the ECEG algorithm in a previous study that aimed to test and compare the ECEG authentication algorithm based on Fiat-Shamir and the Elliptic Curve Diffie-Hellman protocol, utilising Hash Message Authentication Code (HMAC). The parameters for this test are the computation time, delay, memory usage, and communication cost of the authentication algorithm. The results of the experiment showed that the Diffie-Hellman Elliptic Curve algorithm based on Hash Message Authentication Code (ECDH-HMAC) had the lowest compute time, delay, and memory usage, and the Fiat-Shamir-based ECEG algorithm had the lowest communication cost value [7]. Furthermore, in another study, the results of the ECEG algorithm encryption and decryption time test on different elliptical curve parameters showed that the Secp192r1 parameters were 28.9% and 77.1% faster than Secp256r1 and Secp521r1 in the encryption process, while in the decryption process, 27.9% and 73.5% faster than Secp256r1 and Secp521r1. The difference in the encryption and decryption process times of each different parameter is due to the amount of overhead determined by the length of the message character and the parameter values used in the ECEG algorithm [8]. In the previous study, users had to create public keys and private keys first to encrypt and decrypt [9]. The steps that will be taken are to conduct experiments with four types of curves in the text encryption process, and then we will see the peculiarities of each curve.

2 Research Methods

ECEG is a development of the ElGamal algorithm. ElGamal's algorithm consists of three processes, namely the key formation process, the encryption process and the decryption process [10]. ECEG is one of the public key cryptographic algorithms based on elliptic curves and is known to have a high level of security with relatively low computational resource requirements. This study aims to implement the ECEG algorithm in the text coding process to maintain information confidentiality. The encryption process is performed using a public key to convert the original message into an incomprehensible ciphertext, which can only be decrypted with the corresponding private key. Decryption is done by using the appropriate private key to return the message to its original form. This entire cryptographic process can be explained and analysed mathematically through point operations on elliptical curves, such as point summing and scalar multiplication. In a private key setting, two parties share a confidential piece of information called a key, which they use to communicate secretly with each other. The sending party uses the key to encrypt (or "shuffle") the message before it is sent, and the recipient uses the same key to decrypt (or "shuffle") and recover the message after it has been received [11]. ECEG utilises an elliptical curve in the encryption and decryption process. An elliptical curve is a curve with a general shape of the equation:

$$y^2 = x^3 + ax + b$$

$4a^3 + 27b^2 \neq 0$ with the condition. Each different value of a and b gives a different elliptical curve [12]. Elliptic Curve Cryptography (ECC) uses a generator point on the elliptic curve, which can be doubled and added multiple times to create complex keys. The process of point doubling increases the difficulty of deriving the private key from the public key, thus enhancing the security of the key exchange[13]. The value (x,y) obtained from the two points P and Q on the elliptical curve E(k) requires the operation of summing and doubling the points [14]. ECC offers greater security with smaller key sizes by leveraging point doubling and scalar multiplication on elliptic curves, making it highly suitable for blockchain technology [15]. The use of ECC in cloud environments enables secure key exchange and data encryption through complex operations like point doubling, enhancing cloud data confidentiality [16]. ECEG algorithms utilise the mathematical properties of elliptic curves to provide robust encryption and decryption mechanisms, with point doubling significantly complicating unauthorised key recovery [17].

The flowchart of the ECEG Encryption process:

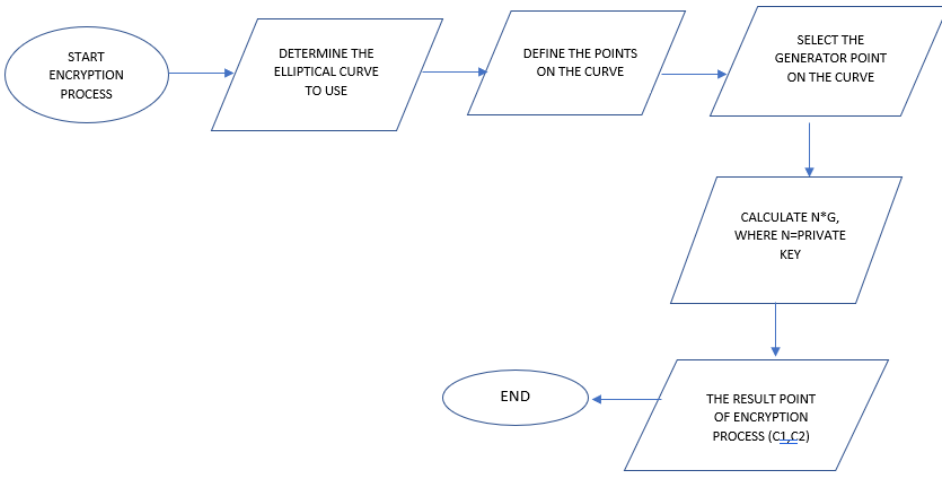

Fig 1. Flowchart of the ECEG Encryption Process

This flowchart illustrates the encryption process using ECEG in structured steps. It starts with the "START ENCRYPTION PROCESS" step. Next, we will determine the elliptical curve to use. After that, define the points on the curve. Select generator points on the curve to represent the 26 letters of the alphabet and numbers to be used. Then, calculate with N*G, when N is the private key. The result of the encryption process is (C_1, C_2). After that, end the process.

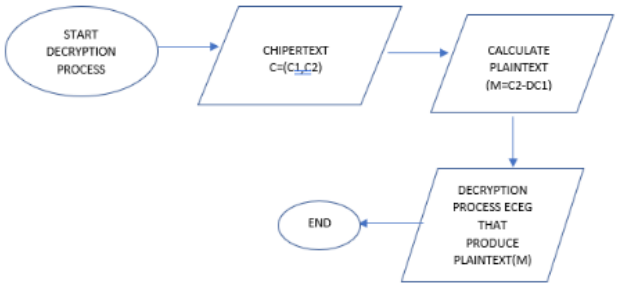

Fig 2. Flowchart Decryption Process

This flowchart illustrates the decryption process using ECEG. The process begins with the "START DECRYPTION PROCESS" step, where the user is asked to enter the ciphertext in the form (C1, C2). Next, the decryption step is performed by calculating the plaintext of M using the formula M = C2 − DC1, where D is the private key. Once the ECEG decryption process is complete, the result is plaintext. This flowchart provides a clear and systematic overview of the steps required to decrypt messages encrypted using the ECEG method, making it easier to understand the decryption workflow in cryptography. Here are examples of plain text, keys, and curves to use:

Plaintext: AYO KITA WISUDA SEKARANG

Curve Elite for ECEG : $y^2 = x^3 + 8x + 25 \pmod{37}$

Elements needed : G = (36 , 4) , k = 4 , d = 2

3 Research Methods

3.1 Encryption Process Using ECEG with curve $y^2 = x^3 + 8x + 25 \pmod{37}$

The encryption process begins by determining the curve to be used. The elliptical curve to be selected is a curve with $a = 8, b = 25, p = 37$.

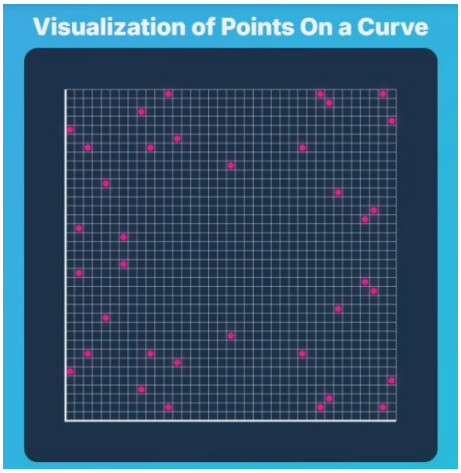

Fig 3. Visualisation of Points on the Elliptic Curve

Fig 3 describes the points on a curve. That distribution of these dots is represented by pink. The output of this visualisation is obtained through an HTML program.

In this implementation, we'll select the point $P = (36.4)$ and its multiplication as the generator that will represent each letter and number. More details are shown in the following table:

Table 1. Element Generator, Curve Point, and Representation Point of Curve.

Element generator	Curve Point	Representation
P	(∞,∞)	(Space)
P	(36,4)	A
2P	(1, 21)	B
3P	(26, 7)	C
4P	(28, 1)	D
5P	(35, 1)	E
6P	(12, 31)	F
7P	(29, 2)	G
8P	(34, 23)	H
9P	(11, 36)	I
10P	(30, 12)	J
11P	(18, 9)	K
12P	(9, 7)	L
13P	(8, 34)	M
14P	(0, 5)	N
15P	(2, 30)	O
16P	(33, 22)	P
17P	(4, 26)	Q
18P	(6, 17)	R
19P	(6, 20)	S
20P	(4, 11)	T
21P	(33, 15)	U
22P	(2, 7)	V
23P	(0, 32)	W
24P	(8, 3)	X
25P	(9, 30)	Y
26P	(18, 28)	Z
27P	(30, 25)	0
28P	(11, 1)	1
29P	(34, 14)	2
30P	(29, 35)	3
31P	(12, 6)	4
32P	(35, 36)	5
33P	(28, 36)	6
34P	(26, 30)	7
35P	(1, 16)	8
36P	(36,33)	9

Table 1 shows the generator elements and their representations. Point P = (36.4) as the representation of the letter A, point 2P = Q = (1.21) as the letter B, and so on 26P = (18.28) as the representation of the letter Z. Next 27P = (30.25) as the representation of the number 0, and so on until 36P = (36.33) represents the number 9.

Next, we calculate C1 with the formula C1 = kP = 4P. So we get C1 = (28, 1), which represents the letter D. After that, we find C2 using the formula: C2 = M + Kq. First example of the plaintext used is "AYOKITAWISUDASEKARANG"

For the first plain text M1 = A Obtained. The following table will display the results of the ciphertext calculation $(\boldsymbol{C}) = [\boldsymbol{C_1}, \boldsymbol{C_2}), (\boldsymbol{C_1}, \boldsymbol{C_2})]$

Table 2. Result and Representation Point of Curve For Example 1.

Plaintext (M)	Point of Plaintext	(C)	Representation of (C)
A	(36,4)	[(28, 1), (11, 36)]	DI
Y	(9,30)	[(28, 1), (28, 36)]	D6
O	(2, 30)	[(28, 1), (0, 32)]	DW
K	(18, 9)	[(28, 1), (6, 20)]	DS
I	(11, 36)	[(28, 1), (4, 26)]	DQ
T	(4, 11)	[(28, 1), (4, 11)]	D1
A	(36, 4)	[(28, 1), (11, 36)]	DI
W	(0, 32)	[(28, 1), (12, 6)]	D4
I	(11, 36)	[(28, 1), (4, 26)]	DQ
S	(6, 20)	[(28, 1), (30, 25)]	D0
U	(33, 15)	[(28, 1), (34, 14)]	D2
D	(28, 1)	[(28, 1), (9, 7)]	DL
A	(36, 4)	[(28, 1), (11, 36)]	DI
S	(6, 20)	[(28, 1), (30, 25)]	D0
E	(35, 1)	[(28, 1), (8, 34)]	DM
K	(18,9)	[(28, 1), (6, 20)]	DS
A	(36,4)	[(28, 1), (11, 36)]	DI
R	(6,17)	[(28, 1), (18, 28)]	DZ
A	(36,4)	[(28, 1), (11, 36)]	DI
N	(0,5)	[(28, 1), (2, 7)]	DV
G	(29,2)	[(28, 1), (26, 30)]	D7

The table 2 show couple points (C) = (C1, C2). (C1, C2) and the letter that representative of the point.

Fig 4. Output Insert ECEG plain text

Fig 4 shows the output of the ECEG HTML program. The generator point used is the point P = (36.4). The public key parameter is the point 2P = (1.21) with k = 4, For example, we will include only 1 instance of text, which is "AYOKITAWISUDASEKARANG".

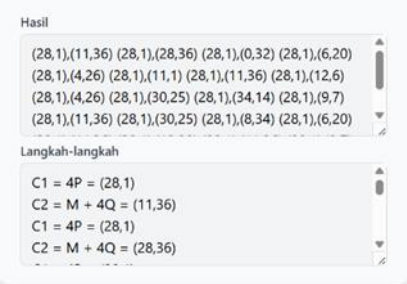

Fig 5. Output ciphertext $(\boldsymbol{C}) = [\boldsymbol{C_1}, \boldsymbol{C_2}), (\boldsymbol{C_1}, \boldsymbol{C_2})]$

Fig 5. **s**how the output of the ciphertext $(\boldsymbol{C}) = [\boldsymbol{C_1}, \boldsymbol{C_2}), (\boldsymbol{C_1}, \boldsymbol{C_2})]$ in html program. Ciphertext point set is {[(28,1),(11,36)] , [(28,1),(28,36)] [(28,1),(0,32)] , [(28,1),(6,20)] , [(28,1),(4,26)] , [(28,1),(11,1)] [(28,1),(11,36)] , [(28,1),(12,6)] , [(28,1),(4,26)] , [(28,1),(30,25)] , [(28,1),(34,14)] , [(28,1),(9,7)] [(28,1),(11,36)] , [(28,1),(30,25)] , [(28,1),(8,34)] , [(28,1),(6,20)] , [(28,1),(11,36)] , [(28,1),(18,28)] [(28,1),(11,36)] , [(28,1),(2,7)] , [(28,1),(2,30)]}

The second example of the plaintext used is "ANNA".

Using the same calculation method as the first example and the representation of the letters follows Table 2, the ciphertext results are obtained as follows:

Table 3. Result and Representation Point of Curve for Example 2.

Plaintext (M)	Point of Plaintext	(C)	Representation of (C)
A	(36,4)	[(28, 1), (11, 36)]	DI
N	(0,5)	[(28, 1), (2, 7)]	DV
N	(0, 5)	[(28, 1), (2, 7)]	DV
A	(36,4)	[(28, 1), (11, 36)]	DI

Table 2 shows a couple of points (C) = (C1, C2). (C1, C2) and the letter that represents the point. Text example 2 shows that, if there is a repetition of the letters, then the repeating letters will result in the same ciphertext.

3.2 Decryption Process Using ECEG with curve $y^2 = x^3 + 8x + 25 \pmod{37}$

For text example 1, in this decryption process, we'll restore the message (C1, C2) to its original plain text. Plaintext can be calculated with the formula:

$$M = C2 - dC1$$

For the DJ plaintext, the result is:

$$M = (11.36) - 2 \cdot (28.1) = (36.4) = A$$

So, the DJ plaintext generates plain text from the letter B. All the results of the first stage of decryption using ECEG will be displayed in the table below:

Table 4. Result and Representation Point of Curve for Example 1.

Ciphertext	(C1, C2)	Plaintext
DI	[(28, 1), (11, 36)]	A
D6	[(28, 1), (28, 36)]	Y
DW	[(28, 1), (0, 32)]	O
DS	[(28, 1), (6, 20)]	K
DQ	[(28, 1), (4, 26)]	I
D1	[(28, 1), (4, 11)]	T
DI	[(28, 1), (11, 36)]	A

Ciphertext	(C1, C2)	Plaintext
D4	[(28, 1), (12, 6)]	W
DQ	[(28, 1), (4, 26)]	I
D0	[(28, 1), (30, 25)]	S
D2	[(28, 1), (34, 14)]	U
DL	[(28, 1), (9, 7)]	D
AT	[(28, 1), (11, 36)]	A
D0	[(28, 1), (30, 25)]	S
DM	[(28, 1), (8, 34)]	E
DS	[(28, 1), (6, 20)]	K
AT	[(28, 1), (11, 36)]	A
DZ	[(28, 1), (18, 28)]	R
AT	[(28, 1), (11, 36)]	A
DV	[(28, 1), (2, 7)]	N
D7	[(28, 1), (26, 30)]	G

From Table 4, we can see plaintext from sample text 1. The same method is also used for text example 2, so that it also produces plaintext “ANNA”. The following is the output of the HTML program as a result of the ECEG decryption:

Fig 6. Output Insert ECEG plain text

Fig 6 indicates the ECEG decryptor. Plaintext used to be input text. Then, to generate plain text, click on the decryption button. To remove the input text again, click the Clean button. The results of the implementation demonstrate that the ECEG algorithm can accurately encode and render text, providing a high level of security. With its efficiency and strong mathematical foundation, ECEG is a relevant choice for modern cryptographic applications, especially in resource-constrained systems.

Conclusion

The conclusions obtained from this study are as follows: The elliptic curve $y^2 = x^3 + 8x + 25 (\mathrm{mod} 37)$ demonstrates distinctive properties that make it suitable for use in an ElGamal Elliptic Curve–based Cryptographic System (ECEG). The curve successfully generates a complete set of points that can act as generator elements representing all alphabetical characters (A–Z) and numerical digits (0–9). This enables the mapping of plaintext characters into unique points on the curve as part of the encryption process.

The implementation results show that both encryption and decryption stages can be executed correctly and consistently. The system is capable of converting plaintext into ciphertext and recovering the plaintext accurately through elliptic curve point operations. In addition, the experiment confirms a deterministic behaviour of the ECEG algorithm: identical plaintext characters always produce identical cipher text pairs (C_1, C_2), as illustrated in Example 2. This characteristic indicates correct algorithmic implementation but also

highlights a potential vulnerability for real-world applications, where randomness should be incorporated to prevent pattern recognition attacks.

Overall, this study demonstrates that the selected curve provides a functional structure for character representation and secure point manipulation, proving its viability for text-based encryption. The successful implementation reinforces the effectiveness of ECEG as a modern cryptographic method offering both mathematical robustness and computational efficiency.

References

[1] P. Kriptografi, "Pengantar Kriptografi," in *Bab-1 Pengantar Kriptografi*, Bandung, 2025, ch. Bab 1, pp. 1–16.

[2] R. Munir, *Kriptografi*. Bandung: Informatika, 2019.

[3] W. Haryono, "Teori Kriptografi dan Aplikasi," *Eureka Media Aksara*, 2024.

[4] S. I. Lestariningati, "Presentasi: Rekayasa Internet - Pengantar Kriptografi," *Sist. Komput.*, 2018.

[5] J. Sasongko, "Pengamanan Data Informasi menggunakan Kriptografi Klasik," vol. X, no. 3, pp. 160–167, 2005.

[6] U. W. Latifah and P. W. Prasetyo, "Implementasi Kriptografi Kurva Eliptik Elgamal Di Lapangan Galois Prima Pada Proses Enkripsi Dan Dekripsi Berbantuan Software Python," *J. Fundam. Math. Appl.*, vol. 4, no. 1, pp. 45–60, 2021, doi: 10.14710/jfma.v4i1.9278.

[7] D. Perdana, P. Purwiko, F. Dewanta, and F. Afianti, "Analisa Penggunaan Elliptic Curve Cryptography pada Sistem Autentikasi pada Internet of Things," vol. 8, no. 1, pp. 42–49, 2022.

[8] N. CANDRA, "Implementasi Algoritma Elliptic Curve Cryptography (Ecc) Dengan End-To-End Encryption Pada Aplikasi Chat Berbasis Mobile," *Stikespanakkukang.Ac.Id*, 2021, [Online]. Available: https://stikespanakkukang.ac.id/assets/uploads/alumni/8a827536b6809e5871a87340e2594ad8.pdf

[9] F. Alfiah, R. Sudarji, and D. T. Al Fatah, "Aplikasi Kriptografi Dengan Menggunakan Algoritma Elgamal Berbasis Java Desktop Pada Pt. Wahana Indo Trada Nissan Jatake," *ADI Bisnis Digit. Interdisiplin J.*, vol. 1, no. 1, pp. 22–34, 2020, doi: 10.34306/abdi.v1i1.114.

[10] D. Ariyus, *Pengantar Ilmu Kriptografi*, no. November. 2008. [Online]. Available: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Pengantar+Kriptografi#0

[11] J. Katz and Y. Lindell, "Introduction to modern cryptography," *Introd. to Mod. Cryptogr.*, pp. 1–527, 2007, doi: 10.1201/9781420010756.

[12] R. Munir, "Elliptic Curve Cryptography (ECC) (Bagian 2)," no. Bagian 2.

[13] GeeksforGeeks, "Blockchain - Elliptic Curve Cryptography," GeeksforGeeks (online article). Accessed: Sep. 25, 2025. [Online]. Available: https://www.geeksforgeeks.org/blockchain-elliptic-curve-cryptography/

[14] R. Afgani, "Public-key Cryptography Elliptic Curves (Kurva Eliptik) Elliptic Curves Cryptography Rizal Afgani Elliptic Curves Cryptography," 2019.

[15] Xiangqi Ruan, "Research on Elliptic Curve Cryptography in Blockchain of Bitcoin," 2025-9-25: SciTePress, 2024, pp. 369–373. [Online]. Available: https://www.scitepress.org/Papers/2024/135242/

[16] A. Ogunleye, G.O., Akinsanya, "Factor Systems in a Cloud Computing Environment Using Elliptic Curve Cryptography," *Iraqi J. Sci.*, vol. 63, no. 7, pp. 3212–3224, 2022, doi: 10.24996/ijs.2022.63.7.40, ISSN: 0067-2904.

[17] S. Ullah, "Elliptic Curve Cryptography: Applications, challenges and solutions," *J. Comput. Secur. Innov.*, 2023, [Online]. Available: https://www.sciencedirect.com/science/article/pii/SXXXX