Malaysian Journal of Mathematical Sciences 16(3): 483–499 (2022)
https://doi.org/10.47836/mjms.16.3.06

Malaysian Journal of Mathematical Sciences

Journal homepage: https://einspem.upm.edu.my/journal

Bézier Coefficients Matrix for ElGamal Elliptic Curve Cryptosystem

Ismail, N. H. M. and Misro, M. Y. *

School of Mathematical Sciences, Universiti Sains Malaysia, Malaysia

E-mail: yushalify@usm.my
**Corresponding author*

Received: 28 December 2021
Accepted: 14 July 2022

Abstract

It is well-known that cryptography is a branch of secrecy in science and mathematics, which usually preserves the confidentiality and authenticity of the information, where its growth is parallel with the rapid evolution of the internet and communication. As one of the prominent public key cryptosystems, the Elliptic Curve Cryptosystem (ECC) offers efficiency and complex mathematical operations with a smaller bit compared to other types of public key schemes. Throughout the evolution of cryptography, ElGamal Elliptic Curve Cryptosystem (ElGamal ECC) revolved from ElGamal public key scheme for user efficiency and privacy. In this study, an improved method will be introduced using ElGamal ECC as the foundation with the incorporation of the Bézier curve coefficient matrix, where the ElGamal ECC value is considered as the control point of the Bézier curve during the encryption and decryption processes. The proposed method is designed to develop a robust ciphertext system algorithm for better efficiency and to increase the level of protection in ElGamal ECC. In this paper, the performance of the proposed method is compared with the normal ElGamal ECC. The results of this study show that the proposed method offers no significant difference in terms of the implementation time during the encryption and decryption process. However, it does offer extra layers of protection when operated with complex mathematical operations.

Keywords: Bézier curve; elliptic curve; cryptosystem; encryption; decryption.

1 Introduction

Goldsmith [7] highlighted the evolution of recent technologies, especially in Information and Communication Technology (ICT) such as telephone, radio, and internet with wired connection are rapidly advancing towards more wireless devices. This advancement of technology required huge improvements in data privacy and security, especially since the current wireless technology has a higher transfer rate and faster connection. Here, the security defence in ICT is crucial since it deals with the risk of attacks and data breaches, remarkably when the data are related to confidential information.

Moreover, Stinson and Paterson [17] defined cryptography as the study of secure communication techniques that allow only mutually trusted parties to view its content. In addition, Diffie and Hellman first established their public key cryptography concept in 1976 to preserve the authentication of messages in an unsecured network [4]. In the same year, ElGamal [5] instituted his public key cryptosystem and signature scheme to the world where the system introduced had some limitations since it is only based on discrete logarithms. Furthermore, Koblitz [9] and Miller [11] were the first to initiate the idea of applying elliptic curve group over a finite field in public key cryptography concept, which is then widely established as Elliptic Curve Cryptography (ECC). Later, Koblitz developed the ElGamal Elliptic Curve Cryptosystem (ElGamal ECC) as an improvement of the ElGamal public key scheme through the elliptic curve group.

Many studies and methods had been developed by researchers to increase the level of security in cryptography such as the application of Bézier curve in cryptosystem by Ghadi and Al-Rammahi (2020) [6], Srividya and Akhila (2014) [16], and Abdul Wahab and Satter Jaber (2016) [19]. However, these studies lack important information on how to embed the Bézier curve coefficient into the cryptosystem to improve its security level. In 2016, Abdul Wahab and Satter Jaber applied Bézier curve with a chaotic system technique to generate an encryption key for a secure communication [19]. Meanwhile, Ghadi and Al-Rammahi manipulated the formula of Menezes-Vanstone Elliptic Curve Cryptosystem (MVECC) with the mathematical notation of quadratic Bézier curve as their proposed method in [6].

Previous studies that have applied Bézier curve in their proposed cryptosystems used only the mathematical notation of Bézier curves. The usage of the messages as the control points of the Bézier curve is yet to be experimented. Therefore, this paper introduced the application of ElGamal ECC with an additional level of security that applies the values as the control points on Bézier curve, where it is not restricted based on specific numbers of data. This method also offers the flexibility for users to send different types of messages (alphabetical characters, numerical, or mixed). This study compares the implementation time and mathematical operations of the suggested method with the original ElGamal ECC. The proposed method also compares the method introduced in [6] since both methods used Bézier curve for encryption and decryption process.

This paper is structured as follows. In Section 2, the mathematical notation of ECC with its operations is introduced, alongside the background of Bézier curve with the Bézier coefficient matrix and its inverse. Section 2 also introduces the ElGamal ECC scheme. On the other hand, the proposed improved method using Bézier coefficient matrix is briefly demonstrated in Section 3. Next, three different examples of messages using the improved method are provided in Section 4. The comparison of our proposed method with existing methods is then discussed in Section 5. Finally, the implemented method will be summarised and concluded in Section 6.

2 Methodology

2.1 Elliptic Curve Cryptography (ECC)

In an Elliptic Curve, E is a graph of the equation such that

$$E : y^2 = x^3 + ax^2 + bx + c, \tag{1}$$

where a, b, c are constants that can satisfy users' appropriate set. However, for ECC, we specifically define p as a prime number, which yields

$$E : y^2 \equiv x^3 + ax + b \pmod{p}, \tag{2}$$

where $a, b \in F_p, p \neq 2, 3$, and satisfy the condition $4a^3 + 27b^2 \neq 0 \pmod{p}$. The elliptic curve group $E(F_p)$ is the set of all points (x, y) along Equation (2) with special point O at infinity as defined in Hankerson *et al.* (2004) [8].

Two points are defined as $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ such that $P \neq Q$, where these points lie along the elliptic curve E as defined in Equation (2). The summation of P and Q results in a new point, R, which also lies on the elliptic curve E as in Equation (3):

$$P + Q = R = (x_3, y_3), \tag{3}$$

such that

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1}, \tag{4}$$

$$x_3 \equiv (\lambda^2 - x_1 - x_2) \pmod{p}, \tag{5}$$

$$y_3 \equiv (\lambda(x_1 - x_3) - y_1) \pmod{p}. \tag{6}$$

Let k be an integer with point $P = (x_1, y_1)$ that lies on E. The scalar multiplication of the elliptic curve can be defined as follows

$$kP = \underbrace{P + P + \ldots + P}_{k-times}, \tag{7}$$

where point P will be added to itself k multiple times. For example, the scalar multiplication of $9P$ can be written as $9P = 2(2(2P)) + P$ as mentioned in Al-Saffar and Said (2015) [13].

Let $P = (x, y)$, then the inverse of the point P is Q, which can be defined as follows:

$$Q = -P = (x, -y), \tag{8}$$

such that $P + Q = O$.

Dawahdeh *et al.* (2016) [3] defined the number of points on the curve as the order of an elliptic curve and can be denoted as $\#E$. Note that, Trappe and Washington (2006) [18] suggested that the value of $\#E$ is includes a point at infinity to assure the elements on the elliptic curve occur along the y-axis. In the selection process of based point in ECC, Al-Saffar and Said (2013) [1] concluded that all points on E can be a base point if $\#E$ is a prime number.

2.2 ElGamal Elliptic Curve Cryptosystem (ElGamal ECC)

ElGamal ECC is the evolution of ElGamal public key cryptosystem, which is based on elliptic curve over a finite prime field introduced by Koblitz (1987) [9]. Let $E(F_p)$ be an elliptic curve group over a finite prime field with publicly known point D such that D lies on E. Firstly, both parties (user A and user B) need to choose their respective private keys, v_a and v_b such that $v_a, v_b \in [1, p-1]$. Then, user B will generate his public key, P_B using his private key, v_b such that $P_B = v_b D$. Secretly, both users need to calculate the symmetric key, K such that $K = v_a P_B = v_b P_A$. Then, message, M will be encrypted; and ciphertext message, C will be sent to user B together with his public key, P_A by user A as follows:

$$C = \{m, P_A\} = \{(x_1, y_1), (x_2, y_2)\},$$

where $m = M + K$ and $P_A = v_a D$. Upon receiving the ciphertext, C, user B will calculate K from P_A and decrypt the ciphertext into message, m as follows:

$$m - K = M + K - K = M. \blacksquare$$

2.3 Bézier Curve

In Marsh (2006) [10] and Safaruddin and Misro (2021) [12], Bézier polynomial of degree n over parameter t can be defined explicitly as follows:

$$P(t) = \sum_{i=0}^{n} P_i B_i^n(t), \qquad 0 \leq t \leq 1, \tag{9}$$

such that P_i are the list of control points and $B_i^n(t)$ are the Bernstein basis functions where

$$B_i^n(t) = \left(\begin{array}{c} n \\ i \end{array} \right) t^i (1-t)^{n-i}, \qquad 0 \leq t \leq 1,$$

with $i = 0, 1, \ldots, n$. The degree of the polynomial will be based on the number of control points defined on the curve. A polynomial, $P(t)$ of degree n over interval $t \in [0, 1]$ can be written as monomial form as shown below:

$$P(t) = A_0 + A_1 t + \cdots + A_n t^n, \tag{10}$$

where A_i are the coefficients of the polynomial. Salomon (2007) [15] highlighted that $P(t)$ can be illustrated in the matrix form as follows

$$P(t) = TMP_i,$$

$$P(t) = \begin{bmatrix} 1 & t & \ldots & t^{n-1} & t^n \end{bmatrix} \begin{bmatrix} a_0 & b_0 & \ldots & z_0 \\ a_1 & b_1 & \ldots & z_1 \\ \vdots & \vdots & \ddots & \vdots \\ a_n & b_n & \ldots & z_n \end{bmatrix} \begin{bmatrix} P_0 \\ \vdots \\ P_n \end{bmatrix}, \tag{11}$$

where T consists of monomial basis, M is the Bézier coefficient matrix, and P_i are control points of the curve.

Based on Equation (11), the matrices coefficient of Linear Bézier (LB), Quadratic Bézier (QB), and Cubic Bézier (CB) curves and its inverse are given as follows

$$LB = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}, \quad LB^{-1} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}. \tag{12}$$

$$QB = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 2 & 0 \\ 1 & -2 & 1 \end{bmatrix}, \quad QB^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & \frac{1}{2} & 0 \\ 1 & 1 & 1 \end{bmatrix}. \tag{13}$$

$$CB = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix}, \quad CB^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & \frac{1}{3} & 0 & 0 \\ 1 & \frac{2}{3} & \frac{1}{3} & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \tag{14}$$

3 The Proposed Method

This section introduces a modification of the ElGamal ECC. This modification will improve the level of security in the ElGamal cryptosystem and maintain the system's efficiency. The coefficient matrix of the Bézier curve will be embedded into ElGamal ECC on this proposed method. This technique also offers flexibility for users to send a secret plaintext regardless in terms of text messages or numbers/coordinates. Multiple levels of security are available for this algorithm, depending on the type of messages involved. All messages will have to go through two levels of security, which are ElGamal ECC and Bézier curve. In contrast, additional security will be applied if the user plans to send a text message by converting each character of the message into decimal values via American Standard Code for Information Interchange (ASCII).

Supposed two parties (user A and user B) communicate some confidential information through an unsecured network. Firstly, an elliptic function E with the domain parameters $\{a, b, p, DP\}$ was agreed upon between the two parties publicly such that a, b are the coefficients of E and p is the prime number that satisfies Equation (2). At the same time, DP is the generator point. Note that if the order of the elliptic curve, $\#E$, is a prime number, then every point on E can be chosen as the generator point, DP. In order to send user B (the receiver) the message, M, both parties need to choose a random private key from the interval $[1, p-1]$. Assume user A's private key is v_a while user B is v_b such that $v_a, v_b \in [1, p-1]$. Hence, the public key for each user can be defined as follows:

$$P_A = v_a DP, \tag{15}$$

$$P_B = v_b DP. \tag{16}$$

From each other's public key, each user can compute the secret key (symmetric key), $K = (x, y)$ by multiplying their private key with the other party's public key, given by

$$K = v_a P_B = v_b P_A = v_a v_b DP = (x, y). \tag{17}$$

Encryption

Supposed user A wants to send a message to user B such that the message is in numerical, text, or the combination text and numerical form. Note that if the messages are in the form of alphabetical characters or the combination between numbers and characters, user A firstly needs to convert the message into decimal value via ASCII and form a new set of numbers, $M =$

$(m_1, m_2, m_3, m_4, \ldots, m_u)$, which consists of the decimal values of the messages. On the other hand, if user A plans to send a numerical message such as coordinates or some sets of numbers, user A can directly form M where $M = (m_1, m_2, m_3, m_4, \ldots, m_u)$ are the list of numerical messages. To avoid confusion for the second party, this method also offers users to differentiate whether the message was supposed to be sent in text form or numerical form, where the final ciphertext will be sent with an alphabet "S" (string) to the user B.

Since the ECC requires two-dimensional coordinates, the list of messages, M, needs to be partitioned into groups of two. These groups are denoted as g_i. Meanwhile, the collection of the groups is defined as G. User A needs to partition M into groups of two such that $G = (g_1, g_2, g_3, \ldots, g_f)$, where $g_1 = (m_1, m_2)$, $g_2 = (m_3, m_4)$ until $g_f = (m_{u-1}, m_u)$. The purpose of this partition of two is to apply ElGamal ECC on G to produce the first ciphertext, C_E. Note that C_E consists of O_f, which are the values of each G, and is added with the symmetric key, K.

$$\begin{aligned} C_E &= G + K \\ O_1 &= g_1 + K \\ &= (m_1, m_2) + (x, y) \\ &= (m_1 + x, m_2 + y) \\ &= (c_1, c_2). \end{aligned}$$

By applying the above steps of ElGamal ECC onto each element G, C_E can be written as $C_E = (c_1, c_2, c_3, c_4, \ldots, c_{u-1}, c_u)$. After user A obtained the ciphertext C_E, user A needs to undergo Algorithm 1 to generate the final ciphertext, F, and send it to user B. In Algorithm 1, a Cubic Bézier (CB) coefficient matrix will be used for the additional protection. CB coefficient matrix is a 4×4 matrix requiring four control points. Therefore, C_E will be partitioned into groups of four, where each of the groups will be a dot product with the CB coefficient matrix. The collections of the groups are defined as $\bar{C_F}$ such that $\bar{C_F} = (C_1, C_2, \ldots, C_N)$ is for messages with lengths of multiple of 4. Meanwhile, $C_F = (\bar{C_F}, C_U)$ is the collections of the groups for lengths that are not of multiple of 4 where C_U is the remaining group elements of F which consist of a group of three, two or c_u, last element of C_E. Correspondingly, C_U will be a dot product with Quadratic Bézier (QB) or Linear Bézier (LB) coefficient matrix, depending on the number of elements, while c_u will be multiplied with p.

Algorithm 1 Encryption by using Bézier coefficient for final ciphertext F.

```
INPUT: $C_E$
OUTPUT: $F$
procedure BÉZIER ENCRYPTION (final encryption)
    if $l \pmod 4 = 0$ then                                        ▷ $l$ Length of $C_E$
        $\bar{C_F} \leftarrow C_E$                                 ▷ $\bar{C_F} = (C_1, C_2, \ldots, C_N)$
        $F \leftarrow \sum_{i=0}^{3} B_i^3(t) \cdot \bar{C_F}$      ▷ By Eq. (14)
    else
        $C_F \leftarrow C_E$                                       ▷ $C_F = (\bar{C_F}, C_U)$
        if $l \pmod 4 = 1$ then
            $F \leftarrow (\sum_{i=0}^{3} B_i^3(t) \cdot \bar{C_F}, p * c_u)$
        else $l \pmod 4 \neq 1$ or 0
            $F \leftarrow (\sum_{i=0}^{3} B_i^3(t) \cdot \bar{C_F}, \sum_{i=0}^{n-1} B_i^{n-1}(t) \cdot C_U)$   ▷ By Eq. (12) or (13)
```

Note that the results of Algorithm 1 will be the final ciphertext that will be sent to user B, which are $F = (f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, \dots, f_{u-3}, f_{u-2}, f_{u-1}, f_u)$ for numerical message and $F = (f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, \dots, f_{u-3}, f_{u-2}, f_{u-1}, f_u, S)$ for text or combination messages. Here, "S" is added to differentiate the types of messages received by user B.

Decryption

Upon receiving ciphertext, F from user A, user B will calculate the symmetric key, $K = (x, y)$ as Equation 17 before undergo Algorithm 2 to generate first level plaintext, P_F from F such that $P_F = (q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, \dots, q_u)$. Note that if the ciphertext, F consist of an alphabet "S", the alphabet must be removed from F and ASCII will be used at the end of the decryption process in order to get the original message. As shown in Algorithm 1, the ciphertext, F, needs to be partitioned into groups of four. In Algorithm 2, the collection of the groups will be defined as $\bar{H}$, such that $\bar{H} = (H_1, H_2, \dots, H_N)$ for ciphertext with lengths of multiple of 4. On the other hand, $H = (\bar{H}, H_U)$ is the collections of the group where the lengths are not multiples of 4 and H_U is a group of remaining elements of F which is a group of three, two or single element, f_u.

Algorithm 2 Decryption by Bézier coefficient for the first ciphertext P_F.

```
1:  INPUT: $F$
2:  OUTPUT: $P_F$
3:  procedure Bézier decryption (first decryption)
4:      if $L \pmod 4 = 0$ then                                          ▷ $L$ is the length of $F$
5:          $\bar{H} \leftarrow F$                                           ▷ $\bar{H} = (H_1, H_2, \dots, H_N)$
6:          $P_F \leftarrow \sum_{i=0}^{3} B_i^3(t)^{-1} \cdot \bar{H}$                      ▷ By Eq. (14)
7:      else if $L \pmod 4 = 1$ then
8:          $H \leftarrow F$                                               ▷ $H = (\bar{H}, H_U)$
9:          $P_F \leftarrow (\sum_{i=0}^{3} B_i^3(t)^{-1} \cdot \bar{H}, \frac{f_u}{p})$
10:     else if $L \pmod 4 \neq 1$ or $0$ then
11:         $H \leftarrow F$
12:         $P_F \leftarrow (\sum_{i=0}^{3} B_i^3(t)^{-1} \cdot \bar{H}, \sum_{i=0}^{n-1} B_i^{n-1}(t)^{-1} \cdot H_U)$   ▷ By Eq. (12) or (13)
```

Finally, after P_F was discovered, user B needs to do the final decryption, which is related to ElGamal ECC such that, P_F needs to be partitioned into P_G. It is groups of two such that $P_G = (g_1, g_2, g_3, \dots, g_f)$, where $g_1 = (q_1, q_2), g_2 = (q_3, q_4)$ until $g_f = (q_{u-1}, q_u)$ before the summation of the inverse symmetric key. Using Equation (8), the inverse of symmetric key, K can be written as

$$K^{-1} = (x, -y).$$

Finally, user B can read message, M as follows:

$$\begin{aligned} M &= P_G + K^{-1} \\ M_1 &= g_1 + K^{-1} \\ &= (q_1, q_2) + (x, -y) \\ &= (q_1 + x, q_2 - y) \\ &= (m_1, m_2). \end{aligned}$$

By applying above steps on each element P_G, finally, user B will be able to read the message, M such that $M = (m, m_2, m_3, m_4, \dots, m_u)$. Note that if the ciphertext, F received consists of the letter "S", user B needs to convert the value of M into characters in ASCII.

4 Implementation Examples for Proposed Method

Supposed two parties (Alice and Bob) accede in using the proposed elliptic curve function, E for their communication via unsecured network such that

$$E : y^2 = x^3 + x + 2547 \pmod{3023},$$

where $a = 1, b = 2547$, and $p = 3023$, satisfying the condition $4a^3 + 27b^2 = 4(1^3) + 27(2547^2) = 175154647 \pmod{3023} = 2027 \neq 0$ and $\#E = 3083$. Thus, they set $DP = (2237, 2480)$ to be their base point.

Key Generation

Alice: Chooses the private key, $v_a = 2313 \in [1, 3022]$ and computes her public key, $P_A = v_a DP = 2313(2237, 2480) = (934, 29)$.

Bob: Chooses the private key, $v_b = 1236 \in [1, 3022]$ and computes his public key, $P_B = v_b DP = 1236(2237, 2480) = (1713, 1709)$.

Symmetric key (secretly by both), K: By using their own private key and other party's public key, the symmetric key, K can be obtained as follows: $K = v_a P_B = v_b P_A = (2537, 1632)$.

4.1 Numerical Messages

Encryption (Alice)

Supposed Alice needs to send a secret message, M to Bob such that

$$M = (567, 78, 2000, 1004, 877, 991, 10, 6).$$

Since the message is in numerical form, she may directly separate the numbers into groups of two, G and get the first ciphertext, C_E such that $C_E = G + K$, where

$$G = (g_1, g_2, g_3, g_4)$$
$$g_1 = (567, 78)$$
$$\vdots$$
$$g_4 = (10, 6).$$

Then,

$$O_1 = (567, 78) + (2537, 1632)$$
$$\vdots$$
$$O_4 = (10, 6) + (2537, 1632).$$

Hence, $C_E = (2010, 708, 494, 335, 890, 1355, 197, 1570)$, which is the first ciphertext. By applying Algorithm 1, Alice will calculate the final ciphertext, F and sends it to Bob for decryption:

$$\bar{C_F} = (C_1, C_2);$$
$$C_1 = (2010, 708, 494, 335),$$
$$C_2 = (890, 1355, 197, 1570).$$

Therefore,

$$\begin{aligned}
F &= \sum_{i=0}^{3} B_i^3(t) \cdot \bar{C_F}, \\
F &= CB \cdot \bar{C_F}, \\
F &= CB \cdot (C_1, C_2), \\
F &= (2010, -3906, 3264, -1033, 890, 1395, -4869, 4154).
\end{aligned}$$

Decryption (Bob)

Upon receiving the ciphertext, F from Alice, Bob first needs to calculate the symmetric key, K and apply the Algorithm 2 in order to obtain the first plaintext, P_F as given below:

$$\begin{aligned}
\bar{H} &= (H_1, H_2); \\
H_1 &= (2010, -3906, 3264, -1033), \\
H_2 &= (890, 1395, -4869, 4154).
\end{aligned}$$

Then,

$$\begin{aligned}
P_F &= \sum_{i=0}^{3} B_i^3(t)^{-1} \cdot \bar{H}, \\
P_F &= CB^{-1} \cdot \bar{H}, \\
P_F &= CB^{-1} \cdot (H_1, H_2), \\
P_F &= (2010, 708, 494, 335, 890, 1355, 197, 1570).
\end{aligned}$$

Therefore, using Equation (8), the inverse of K can be written as

$$K^{-1} = (2537, -1632).$$

Then, Bob needs to separate P_F into groups of two, P_G and calculate it using ElGamal ECC to get the message, M

$$\begin{aligned}
P_G &= (g_1, g_2, g_3, g_4); \\
g_1 &= (2010, 708) \\
&\vdots \\
g_4 &= (197, 1570)
\end{aligned}$$

such that

$$\begin{aligned}
M &= P_G + K^{-1}; \\
M_1 &= (2010, 708) + (2537, -1632) \\
&= (2010 + 2537, 708 - 1632) \\
&= (567, 78) \\
&\vdots \\
M_4 &= (197, 1570) + (2537, -1632) \\
&= (197 + 2537, 1570 - 1632) \\
&= (10, 6).
\end{aligned}$$

Hence, Bob will get Alice's message, which is

$$M = (567, 78, 2000, 1004, 877, 991, 10, 6). \quad \blacksquare$$

4.2 Text Message

Encryption (Alice)

Suppose Alice wishes to share a secret plaintext *Computer* with Bob via an unsecured network. First, she should convert each character of the text message *Computer* into decimal values of ASCII, as given below:

$$Computer \xrightarrow{ASCII} 67, 111, 109, 112, 117, 116, 101, 114.$$

Then, let $M = (67, 111, 109, 112, 117, 116, 101, 114)$ be a set of text messages in decimal values of ASCII. Next, M is separated into groups of two, G, in which the second ciphertext, C_E is calculated given by $C_E = G + K$, where

$$G = (g_1, g_2, g_3, g_4);$$
$$g_1 = (67, 111)$$
$$\vdots$$
$$g_4 = (101, 114).$$

Thus,

$$O_1 = (67, 111) + (2537, 1632)$$
$$\vdots$$
$$O_4 = (101, 114) + (2537, 1632).$$

Hence, $C_E = (2660, 472, 2606, 2558, 1050, 94, 1967, 674)$ will be the second ciphertext. Next, Alice needs to undergo Algorithm 1 and add the character "S" at the end of F before sending it to Bob as an alert of a text message.

$$\begin{aligned}\bar{C_F} &= (C_1, C_2);\\ C_1 &= (2660, 472, 2606, 2558),\\ C_2 &= (1050, 94, 1967, 674).\end{aligned}$$

Then,

$$\begin{aligned}F &= \sum_{i=0}^{3} B_i^3(t) \cdot \bar{C_F},\\ F &= CB \cdot \bar{C_F},\\ F &= CB \cdot (C_1, C_2),\\ F &= (2660, -6564, 12966, -6504, 1050, -2868, 8487, -5995, S).\end{aligned}$$

Decryption (Bob)

When Bob receives the ciphertext F from Alice, Bob needs to calculate K before undergoing Algorithm 2 in order to get the first level plaintext, P_F such that

$$\begin{aligned}\bar{H} &= (H_1, H_2);\\ H_1 &= (2660, -6564, 12966, -6504),\\ H_2 &= (1050, -2868, 8487, -5995).\end{aligned}$$

Then,

$$\begin{aligned}
P_F &= \sum_{i=0}^{3} B_i^3(t)^{-1} \cdot \bar{H}, \\
P_F &= CB^{-1} \cdot \bar{H}, \\
P_F &= CB^{-1} \cdot (H_1, H_2), \\
P_F &= (2660, 472, 2606, 2558, 1050, 94, 1967, 674).
\end{aligned}$$

Therefore, P_F will be the first ciphertext that Bob receives. Then, using Equation (8), the inverse of K can be written as

$$K^{-1} = (2537, -1632).$$

Then, Bob needs to separate P_F into groups of two, P_G and calculate it by using ElGamal ECC to get the message, M, as follows:

$$\begin{aligned}
P_G &= (g_1, g_2, g_3, g_4); \\
g_1 &= (2660, 472) \\
&\vdots \\
g_4 &= (1967, 674),
\end{aligned}$$

such that

$$\begin{aligned}
M &= P_G + K^{-1}; \\
M_1 &= (2660, 472) + (2537, -1632) \\
&= (2660 + 2537, 472 - 1632) \\
&= (67, 111) \\
&\vdots \\
M_4 &= (1967, 674) + (2537, -1632) \\
&= (1967 + 2537, 674 - 1632) \\
&= (101, 114).
\end{aligned}$$

Thus, $M = (67, 111, 109, 112, 117, 116, 101, 114)$. Note that this M is not the final answer since this message was meant to be a text message. Subsequently, using ASCII, Bob will then convert the number in the set M into a text message

$$67, 111, 109, 112, 117, 116, 101, 114 \xrightarrow{ASCII} Computer. \quad \blacksquare$$

4.3 Combination of Numerical and Text Message

Encryption (Alice)

Suppose Alice wants to send *Covid19* to Bob such that the plaintext is the combination of text message and numerical message. First, Alice needs to convert the plaintext into M, a set of decimal values of the message converted via ASCII

$$Covid19 \xrightarrow{ASCII} 67, 111, 118, 105, 100, 49, 57,$$

where $M = (67, 111, 118, 105, 100, 49, 57)$. Then, M is separated into groups of two, G and the second ciphertext is calculated, C_E such that $C_E = G + K$, where

$$G = (g_1, g_2, g_3, 57);$$
$$g_1 = (67, 111)$$
$$\vdots$$
$$g_3 = (100, 49).$$

Thus,

$$O_1 = (67, 111) + (2537, 1632)$$
$$\vdots$$
$$O_3 = (100, 49) + (2537, 1632),$$

while

$$O_4 = 57 + 2537.$$

Therefore, $C_E = (2660, 472, 2127, 1906, 2196, 2198, 2594)$ will be the second ciphertext for this message. By applying Algorithm 1, Alice sends F to Bob such that

$$\begin{aligned} C_F &= (C_1, C_2); \\ C_1 &= (2660, 472, 2127, 1906), \\ C_2 &= (2196, 2198, 2594). \end{aligned}$$

Then, since C_2 consists of 3 numbers, F can be written as follows, with "S" at the end.

$$\begin{aligned} F &= (\sum_{i=0}^{3} B_i^3(t) \cdot C_1, \sum_{i=0}^{2} B_i^2(t) \cdot C_2), \\ F &= (CB \cdot C_1, QB \cdot C_2), \\ F &= (2660, -6564, 11529, -5719, 2196, 4, 394, S). \end{aligned}$$

***Decryption* (*Bob*)**

Upon receiving the ciphertext, F, from Alice, Bob applies Algorithm 2 in order to generate the first level plaintext, P_F, which can be written as follows:

$$H = (H_1, H_2);$$

such that

$$\begin{aligned} H_1 &= (2660, -6564, 11529, -5719), \\ H_2 &= (2196, 4, 394). \end{aligned}$$

Hence, each set of H needs to undergo dot product with the inverse of Bézier coefficient in order to get the first plaintext, P_F, given below:

$$\begin{aligned} P_F &= (\sum_{i=0}^{3} B_i^3(t)^{-1} \cdot H_1, \sum_{i=0}^{2} B_i^2(t)^{-1} \cdot H_2), \\ P_F &= (CB^{-1} \cdot H_1, QB^{-1} \cdot H_2), \\ P_F &= (2660, 472, 2127, 1906, 2196, 2198, 2594). \end{aligned}$$

Then, after calculating the symmetric key, K, as in Equation (8), the inverse of K can be written as

$$K^{-1} = (2537, -1632).$$

Correspondingly, Bob needs to separate P_F into groups of two, P_G with one leftover element before using ElGamal ECC to get the message, M:

$$P_G = (g_1, g_2, g_3, 2594);$$

$$g_1 = (2660, 472)$$

$$\vdots$$

$$g_3 = (2196, 2198),$$

such that

$$\begin{aligned} M &= P_G + K^{-1}; \\ M_1 &= (2660, 472) + (2537, -1632) \\ &= (2660 + 2537, 472 - 1632) \\ &= (67, 111) \\ &\vdots \\ M_3 &= (2196, 2198) + (2537, -1632) \\ &= (2196 + 2537, 2198 - 1632) \\ &= (100, 49) \end{aligned}$$

with

$$M_4 = 2594 - 2537 = 57.$$

Hence, $M = (67, 111, 118, 105, 100, 49, 57)$. Since M is not the final answer, Bob needs to convert M via ASCII to read the message

$$67, 111, 118, 105, 100, 49, 57 \xrightarrow{ASCII} Covid19. \quad \blacksquare$$

5 Result and Discussion

Several comparisons are made in this section to discuss our proposed method by analysing it against the implementation time and required operations of the normal ElGamal ECC. This simulation uses Wolfram Mathematica (Version 12.3.1 for Mac OS X x86/ 64-bit). The results of ElGamal ECC [1] and Ghadi and Al-Rammahi's MVECC (GAMVECC) in [6] will be compared.

Table 1 shows the implementation time for normal ElGamal ECC on Wolfram Mathematica versus Al-Saffar's on MATLAB (version 7.10.0.499/ 32-bits). As mentioned in [1], cryptosystem offers different implementation times depending on the input size, length of algorithms or code, as well as the type of language and processors used throughout the encryption and decryption.

Table 1: The implementation time of encryption and decryption for normal ElGamal ECC using Wolfram Mathematica (Author's) and MATLAB (Al-Saffar's).

	ElGamal ECC (Author's)		ElGamal ECC (Al-Saffar's)	
Messages/ Times	Encryption Time (Sec)	Decryption Time (Sec)	Encryption Time (Sec)	Decryption Time (Sec)
(8228,5025)	0.190054	0.113603	0.292592	0.176354
(8219,7676)	0.184969	0.105382	0.290878	0.170969
(7570,7470)	0.186887	0.105246	0.283419	0.203476

Table 2 illustrates the implementation time on GAMVECC in [6] using different platforms. Based on Tables 1 and 2, the time taken for encryption and decryption on Wolfram Mathematica (Version 12.3.1 for Mac OS X x86/ 64-bit) is much faster than MATLAB (version 7.10.0.499/ 32-bits) by Al-Saffar and MATLAB R2018b (version 9.5.0.944444/ 64-bits) in [6]. Note that Tables 1 and 2 are presented to compare the gaps between the two platforms before continuing to examine the proposed method on Wolfram Mathematica.

Table 2: The implementation time of encryption and decryption process for MVECC using Wolfram Mathematica (Author's) and MATLAB (GAMVECC).

	GAMVECC (Author's)		GAMVECC	
Text characters/ Times	Encryption Time (Sec)	Decryption Time (Sec)	Encryption Time (Sec)	Decryption Time (Sec)
1000-bits	0.039579	0.019957	16.9943	5.2962
3000-bits	0.040894	0.021078	59.5675	18.8306
5000-bits	0.042974	0.022233	85.3577	26.5149

Table 3 illustrates the time taken for the messages to be encrypted and decrypted between the proposed method and the original ElGamal ECC via Wolfram Mathematica. Based on the results in Table 3, it is concluded that there is no significant difference between the time taken for encryption and decryption between both methods, where two of the messages took a slightly faster time to decrypt, and the other message took a faster time to encrypt. This concludes that the proposed method suggests a higher level of security with no significant difference in time performance compared to the normal ElGamal ECC.

Table 3: The time implementation in encryption and decryption between proposed method and ElGamal ECC by Wolfram Mathematica.

	Proposed method		ElGamal ECC	
Messages/ Times	Encryption Time (Sec)	Decryption Time (Sec)	Encryption Time (Sec)	Decryption Time (Sec)
(8228,5025)	0.186777	0.113733	0.190054	0.113603
(8219,7676)	0.187144	0.103956	0.184969	0.105382
(7570,7470)	0.188981	0.104205	0.186887	0.105246

In contrast to Table 3, Table 4 shows distinguishable results between the currently proposed method and GAMVECC. After studying the comparison between the two methods, it is concluded that there is a huge gap between the time taken for encryption and decryption, where the proposed method consumes a longer time for encryption and decryption. Furthermore, although the proposed method requires a longer implementation time, it is known that the implementation time for the original ElGamal ECC is significantly longer compared to the original MVECC in [1]. This confirms the validity of the comparison in Table 4 since the gap between both methods is well established. In addition, this method offers a higher level of security with a lower mathematical complexity compared to Ghadi's method.

Table 4: The time implementation for the proposed method and Ghadi's MVECC in encryption and decryption process via Wolfram Mathematica.

	Proposed method		GAMVECC	
Text characters/Times	Encryption Time (Sec)	Decryption Time (Sec)	Encryption Time (Sec)	Decryption Time (Sec)
1000-bits	0.188085	0.123242	0.039579	0.019957
3000-bits	0.195034	0.125143	0.040894	0.021078
5000-bits	0.193191	0.12756	0.042974	0.022233

Since the proposed method offers two different workflows between text and numerical messages, Table 5 presents the justification for the proposition of this improved method. It is known that numbers can be converted into decimal values via ASCII. However, in this paper, the authors purposely separate the workflows of encryption and decryption, where only text and combination messages will undergo ASCII. Excluding ASCII conversion for numerical messages is shown to be more beneficial for the proposed method, where the implementation time is much faster as opposed to the inclusion of the ASCII conversion. It also shows that workflow without ASCII for numerical messages offers a smaller size of bytes compared to the workflow with ASCII.

Table 5: The time implementation and size of bytes for numerical messages via ASCII.

Proposed Method	General		Numerical ASCII	
Size(bytes)	64,116		67,068	
Messages	Encryption Time (Sec)	Decryption Time (Sec)	Encryption Time (Sec)	Decryption Time (Sec)
(8228,5025)	0.186777	0.113733	0.195252	0.121745
(8219,7676)	0.187144	0.103956	0.192871	0.111378
(7570,7470)	0.188981	0.104205	0.199272	0.111613

Table 6: The required operations for the ElGamal ECC, GAMVECC, and the proposed method.

ElGamal ECC		GAMVECC		Proposed method	
Encryption	Decryption	Encryption	Decryption	Encryption	Decryption
2 Add	2 Add + 1 Inv	6 Mult + 4 Add + 4 Sub	6 Mult + 8 Sub + 2 Inv	2 Add + 1 Dot Product	2 Add + 1 Inv + 1 Dot Product

*Note that the number of mathematical operations for the proposed method varies depending on the length of the message involved

Finally, Table 6 summarised the mathematical complexity between all three methods in the encryption and decryption process. This table presents the number of operations involved in the proposed method together with ElGamal ECC and GAMVECC. Based on Table 6, it can be concluded that the proposed method possesses more complex computational operations compared to the original ElGamal ECC and GAMVECC. Note that, the number of mathematical operations for the proposed method varies depending on the length of the message involved. The proposed method can be deduced to be more efficient as it offers extra operation with one Dot Product operation for each encryption and decryption compared to the original ElGamal ECC.

6 Conclusions

In this paper, a new technique of ElGamal ECC combined with Bézier curve coefficient matrix has been introduced to provide a safer and rather efficient information exchanger for the encryption and decryption processes. The proposed method is regarded to be more flexible as it allows users to send messages in various forms. The proposed method also suggested differentiating the type of messages to obtain more accurate results, especially when dealing with numerical messages. This is because the numerical messages can directly use the proposed method without converting it to ASCII form. This proposed technique is believed to have a strong level of security with lower mathematical complexity since it offers a few additional operations with no significant difference in the implementation time compared to the original method of ElGamal ECC. However, extra layers of security consume more hardware storage due to a large number of key sizes. Thus, this has become a limitation of the proposed method. The proposed method is potentially exposed to side-channel attacks when it converts messages into decimal values using ASCII. However, the authentication of the messages is expected to be well preserved. For future studies, we would like to suggest incorporating the Bézier curve coefficient matrix in the MVECC scheme. Different types of Bézier basis functions, especially trigonometric [2] and fractional [14] could potentially be explored.

Acknowledgement This research was supported by Universiti Sains Malaysia under Short Term Grant (Khas) (304/PMATHS/6315587) and the School of Mathematical Sciences, Universiti Sains Malaysia. The authors are very grateful to the anonymous referees for their valuable suggestions.

Conflicts of Interest The authors declare no conflict of interest.

References

[1] N. H. Al-Saffar & M. R. M. Said (2013). On the mathematical complexity and the time implementation of proposed variants of elliptic curves cryptosystems. *International Journal of Cryptology Research*, *4*(1), 42–54.

[2] M. Ammad, M. Y. Misro & A. Ramli (2022). A novel generalized trigonometric Bézier curve: properties, continuity conditions and applications to the curve modeling. *Mathematics and Computers in Simulation*, *194*, 744–763. https://doi.org/10.1016/j.matcom.2021.12.011.

[3] Z. E. Dawahdeh, S. N. Yaakob & R. R. B. Othman (2016). A new modification for Menezes-Vanstone elliptic curve cryptosystem. *Journal of Theoretical and Applied Information Technology*, *85*(3), 290–297.

[4] W. Diffie & M. Hellman (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, *22*(6), 644–654. https://doi.org/10.1109/TIT.1976.1055638.

[5] T. ElGamal (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, *31*(4), 469–472. https://doi.org/10.1109/TIT.1985.1057074.

[6] D. M. Ghadi & A.-R. Adil (2020). Improvement of Menezes-Vanstone elliptic curve cryptosystem based on quadratic Bézier curve technique. *Journal of Computer Science*, *16*(5), 715–722. https://doi.org/10.3844/jcssp.2020.715.722.

[7] A. Goldsmith (2005). *Wireless communications*. Cambridge University Press, Cambridge, UK.

[8] D. Hankerson, A. J. Menezes & S. Vanstone (2004). *Guide to elliptic curve cryptography*. Springer, New York, NY.

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

[10] D. Marsh (2005). *Applied geometry for computer graphics and CAD*. Springer, London.

[11] V. S. Miller (1985). Use of elliptic curves in cryptography. In *Williams, H.C. (eds) Advances in Cryptology – CRYPTO'85 Proceedings. CRYPTO 1985. Lecture Notes in Computer Science*, pp. 417–426. Springer, Berlin, Heidelberg.

[12] M. Safaruddin & M. Misro (2021). Multi-objectives path planning using Bézier curve. *Malaysian Journal of Mathematical Sciences*, *15*(1), 45–59.

[13] N. F. H. A. Saffar & M. R. M. Said (2015). Speeding up the elliptic curve scalar multiplication using the window-w non adjacent form. *Malaysian Journal of Mathematical Sciences*, *9*(1), 91–110.

[14] S. A. A. A. Said Mad Zain, M. Y. Misro & K. T. Miura (2021). Generalized fractional Bézier curve with shape parameters. *Mathematics*, *9*(17), 2141. https://doi.org/10.3390/math9172141.

[15] D. Salomon (2007). *Curves and surfaces for computer graphics*. Springer, New York, NY.

[16] B. Srividya & S. Akhila (2014). Novel cryptosystem based on Bézier curve using GF (p m). In *International Conference on Circuits, Communication, Control and Computing*, pp. 304–307. IEEE, Bangalore, India.

[17] D. R. Stinson & M. Paterson (2019). *Cryptography: theory and practice*. CRC Press, Florida, US.

[18] W. Trappe & L. Washington (2006). *Introduction to cryptography with coding theory*. Pearson Prentice Hall, New Jersey, US.

[19] H. B. A. Wahab & T. A. Jaber (2016). Using Chebyshev polynomial and quadratic Bézier curve for secure information exchange. *Engineering and Technology Journal*, *34*(5), 666–674.