The comparison of several cryptosystems using the elliptic curve: a report

Mai Manh Trung[1, 2], Le Phe Do[2], Do Trung Tuan[3], Thu Thuy Trieu[4], Nguyen Van Tanh[5], Ngo Quang Tri[1], Bui Van Cong[1]

[1]Department of Information System, Faculty of Information Technology, University of Economics Technology for Industries, Hanoi City, Vietnam
[2]Department of Computer Science, Faculty of Information Technology, University of Engineering and Technology, Vietnam National University, Hanoi City, Vietnam
[3]Department of Computational Science, Faculty of Information Technology, University of Science, Vietnam National University, Hanoi City, Vietnam
[4]Department of Informatics, Institute of Information Technology, Vietnam Academy of Science and Technology, Hanoi City, Vietnam
[5]Faculty of Applied Sciences, International School, Vietnam National University, Hanoi City, Vietnam

Article Info

Article history:

Received Sep 18, 2023
Revised May 25, 2024
Accepted Jun 4, 2024

Keywords:

Affine cryptography
Data security
Symmetric key
The cryptosystem
The elliptic curve
Vigenère cryptography
Generating point

ABSTRACT

The elliptic curve cryptosystem (ECC) has several applications in Information Security, especially in cryptography with two main activities including encrypting and decrypting. There were several solutions of different research teams which propose various forms of the elliptic curve cryptosystem on cryptographic sector. In the paper, we proposed a solution for applying the elliptic curve on cryptography which is based on these proposals as well as basic idea about the elliptic curve cryptosystem. We also make comparison between our proposal and other listed solution in the same application of the elliptic curve for designing encryption and decryption algorithms. The comparison results are based on parameters such as time consumption (t), RAM consumption (MB), source code size (Bytes), and computational complexity.

Corresponding Author:

Mai Manh Trung
Department of Information System, Faculty of Information Technology, University of Economics Technology for Industries
Hanoi City, Vietnam
Email: mmtrung@uneti.edu.vn

1. INTRODUCTION

The elliptic curve cryptography (ECC) is one of several cryptosystems which have widespread use currently. For example, several enterprises in the United States such as CloudFlare use the ECC at huge scale to protect the confidentiality of the hypertext transfer protocol secure (HTTPS) connections between customers and transmission between data centers. In 1987, according to [1], an algorithm was proposed to divide an integer into prime factors using elliptic curve. The cost of this algorithm is averagely low which is better than the notation O(n2). This cost is lower than the polynomial sieving algorithm and general numerical field sieving algorithm which means it reaches the third lowest.

In the aspect of cryptography, the first paper about the application of elliptic curve in cryptography was published in 1985 [2], followed by the proposal of a cryptosystem based on elliptic curve cryptography in 1987 [3]. After that, there are several papers for researching cryptography from the elliptic curve about theoretical ideas and practical applications. As a result, the ECC has expanded its applications and currently, it is considered a security standard.

Elliptic curve is a contemporary and advanced method employed in the domain of ECC. ECC is frequently used to bolster the security of public communication networks [4] and to grant access to the modern digital era (MDE) to those with verified identities. Users of MDE utilize several technologies, including social media [5], cloud computing [6], the internet of things (IoT) sector [7], and data mining with privacy preservation [8]. The multicore wireless sensor network utilized an efficient authenticated ECC technique [9] to construct secure IoT for logistic regression [10]. In addition, ECC is utilized in blockchain to safeguard the royalties of digital authors [11], [12], in distributed group key management [13], in smart home applications [14], in an efficient hardware implementation [15], and in a novel m-commerce data security mechanism [16]. Ensuring the security and privacy of users is of utmost importance for the entire system, irrespective of the individual technologies used. The study of cryptography is essential due to the susceptibility of data transmission and information transfer to data theft and attacks over unsecured networks. Hence, it is crucial to obtain expertise in cryptography. Cryptography is the act of using keys to encode documents and communications, ensuring that only the intended receivers can decipher and handle them. Digital signatures, cryptographic data integrity, and authentication methods rely on the recipient and sender's addresses, along with mathematical operations to authenticate the signature.

In terms of key features, the cryptosystems are classified into two categories such as symmetric or asymmetric cryptosystems. As we know, the Rivest–Shamir–Adleman (RSA) public key cryptography has been widely applied in practice, but ECC can potentially replace RSA with better security and higher processing speed. Table 1 provides the key lengths for symmetric key cryptography, ECC cryptography, and RSA cryptography. Table 2 compares the key sizes of RSA and ECC cryptography, showing that RSA key sizes are much larger than ECC key sizes, yet they offer similar levels of security.

Table 1. Key size for symmetric key cryptography, ECC cryptography, and RSA public key cryptography [17]

Symmetric-key (bit)	ECC (bit)	RSA (bit)
64	128	700
80	160	1,024
128	256	2048-3072

Table 2. Comparison of RSA and ECC key sizes at equivalent security levels [18]

Time it takes	Key size		Key size ratio RSA: ECC
Key (unit: year)	RSA	ECC	
104	512	106	5:1
108	768	132	6:1
1011	1,024	160	7:1
1020	2,048	210	10:1
1078	21,000	600	35:1

The ECC is categorized as an asymmetric cryptosystem that employs distinct public and private keys in its operations [19]. Using ECC, [20] introduced an algorithm for encrypting photos with a public key. The encrypted images are represented as a matrix of pixels. In the context of symmetric cryptography, a single key is employed for both the process of encrypting and decrypting data [21], [22]. In their paper [23], the research team introduced an ECC for encrypting Vietnamese text using a symmetric single-value key. Subsequently, the research team [24] suggested employing ECC as a means of encrypting and decrypting data strings. The study conducted by paper [25] investigated the procedure of encrypting and decrypting Vietnamese texts using ECC with a symmetric key consisting of two parts. In the publication [26], the team introduced a cryptosystem that utilizes the elliptic curve for data encryption and decryption with a public key. This study assesses the implementation of ECC in the cryptosystem using several methods mentioned in paper [23]–[27]. The evaluation is based on multiple metrics, such as the length of the source code, memory capacity, and processing time.

2. MATHEMATICAL BASIS OF THE ELLIPTIC CURVE

In several studies, almost of them uses the elliptic curve equation as (1):

$$y^2 = x^3 + ax + b(mod\ p) \quad (1)$$

In (1), p represents a prime integer, while the variables a and b are constants. The variables x, y, a, and b are commonly chosen from a field, such as the set of real numbers (R), rational numbers (Q), complex numbers (C), or a finite field. An elliptic curve is said to be defined over the field K if K is a field and both a and b belong to K. A point (x, y) on the elliptic curve, where (x, y) is an element of the field K, is referred to as a K-rational point. The curve (1) has determinant $\Delta = 4a^3 + 27b^2$. This curve will degenerate and not have enough 3 distinct solutions when $\Delta = 0$. We only evaluate the elliptic curve in this paper which has $\Delta \neq 0$.

2.1. The addition of points in the curve

Consider two points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ on the curve defined by (1). The addition of these two points on the curve E is defined as (2):

$$P_3(x_3, y_3) = P_1(x_1, y_1) + P_2(x_2, y_2) \tag{2}$$

In which $P_3(x_3, y_3) = -P'_3(x_3, y'_3)$, point $P'_3(x_3, y'_3)$ is the junction point of curve E with the line that passes through points P1 and P2, as shown in Figure 1. Because 2 points $P_3(x_3, y_3)$ and $-P'_3(x_3, y'_3)$ lie on the curve E, (x_3, y_3) and (x_3, y'_3) must be compatible to the formula (1).

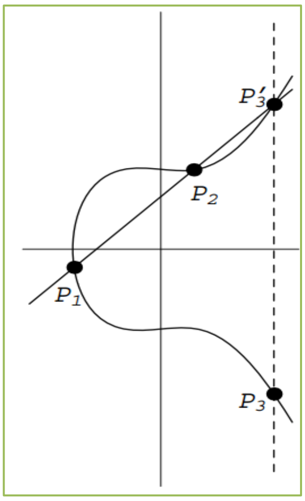

Figure 1. An illustrative example of addition on the elliptic curve

If $x_1 = x_2$ combined with condition $y_1 = -y_2$, we set $P_1 + P_2 = \infty$ (∞ is infinite point). In contrast, $P_1 + P_2 = P_3 = (x_3, y_3) \in E$ which $x_3 = \mu^2 - x_1 - x_2, y_3 = \mu(x_1 - x_3) - y_1$, and the value of μ is calculated in the formula (3):

$$\mu = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}, & \text{if } P_1 \neq P_2 \\ \frac{3x_1^2 + a}{2y_1}, & \text{if } P_1 = P_2 \end{cases} \tag{3}$$

If $P_1 \neq P_2$, it means $x_1 \neq x_2$, we have (4):

$$\begin{cases} x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2 \\ y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1 \end{cases} \tag{4}$$

If $P_1 = P_2$, it means $x_1 = x_2$, we have (5):

$$\begin{cases} x_3 = \left(\frac{3{x_1}^2 + a}{2y_1}\right)^2 - 2x_1 \\ y_3 = \left(\frac{3{x_1}^2 + a}{2y_1}\right)(x_1 - x_3) - y_1 \end{cases} \tag{5}$$

We argue that point (x_3, y_3) and point (x_3, -y_3) belong to the curve E. As the result, in the Cartesian coordinate system, both 3 points (x_1, y_1), (x_2, y_2) and (x_3, -y_3) also belong to one line.

Addition with points P, P_1, P_2, and P_3 on the curve E satisfies the properties of a group:

- Commutativity: $P_1 + P_2 = P_2 + P_1$;
- Identity element: $P + \infty = P$;
- Inverse element: for each point P, there exists a point P' such that $P + P' = \infty$; and
- Associativity: $(P_1 + P_2) + P_3 = P_1 + (P_2 + P_3)$.

2.2. The scalar multiplication of point in the elliptic curve

For $n \in N \setminus \{0\}$, the scalar multiplication of a point P on the curve E is defined as the sum of the point P with itself n times:

$$P \rightarrow nP = \underbrace{\mathrm{P} + \mathrm{P} + \cdots + \mathrm{P}}_{\text{n times}} = \mathrm{Q} \tag{6}$$

We use the double-and-add method to optimize scalar multiplication. First, express the number $n = n_0 + 2n_1 + 2^2n_2 + \cdots + 2^mn_m$ with $[n_0 \ldots n_m] \in \{0,1\}$.

When working with elliptic curves, the process of multiplying a point in the Cartesian coordinate system by a constant is not a straightforward operation of multiplying each coordinate of the point by the same constant. Equation (10) defines multiplication as the process of iteratively adding a number to itself. In order to calculate the product 3P, we first find the sum of P and P to produce 2P, and then add 2P to P to obtain 3P.

In order to determine the value of 2P, we employ the method of drawing a tangent line from point P to the curve. Simultaneously, the point at which this line intersects with the curve is referred to as -2P. The point 2P is the image of -2P obtained by reflecting it across the horizontal axis of the coordinate plane.

In order to obtain the point 3P, we create a line segment that connects points 2P and P. Next, we determine the coordinates of the intersection point -3P. Ultimately, we ascertain the symmetrical position of -3P by mirroring it across the horizontal axis of the coordinate system. Figure 2 depicts the procedure of multiplication in the elliptic curve.

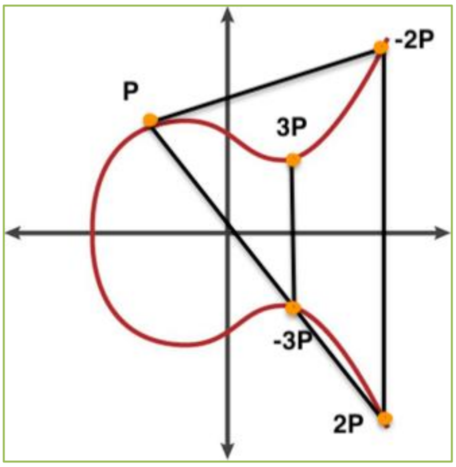

Figure 2. An illustrative example of scalar multiplication on an elliptic curve

3. DETAILS OF PROPOSED CRYPTOGRAPHIC ALGORITHMS IN THE ELLIPTIC CURVE

This section presents a detailed explanation of the research results, along by a thorough analysis and conversation. The results can be represented using figures, graphs, tables, and other visual aids to help readers understand [14], [15]. The debate can be structured into several sub-sections as required.

3.1. Description of algorithm for generating series

This algorithm necessitates generating a series of vectors known as *Si*. The steps for this are detailed in Algorithm 1. The outcome of this procedure is then applied to the ECC output.

Algorithm 1. Generating series [9]
Input: Parameters of the elliptic curve
Output: Series of bits
Step 1:
- Calculate the number of points (n) contained within the elliptical form.
- The point q is the point that generates the following equation.
- Calculate the collection of points on an elliptic curve utilizing the generator point q.

Step 2:
- Transform the numerical value n into the ternary numeral system.
- Let m represent the number of characters that transforms the number of points to the base 3.

Step 3:
- Generate a matrix (M) with a total of (n+1)*m elements. Therefore, the number of rows is represented by (n+1), whereas the number of columns and the number of elements in each row are both denoted by m.

$$M = \left(a_{0,0}a_{0,1} \dots \dots a_{0,m}\ a_{1,0}a_{1,1} \dots \dots a_{1,m}\ a_{2,0}a_{2,1} \dots \dots a_{2,m}\ \dots \dots\ a_{n,0}a_{n,1} \dots \dots a_{n,m}\ \right)$$

Step 4:
- Shift each element from the current position of the matrix created in Step 3 to the right position: [$a_{i,0}$ $a_{i,1}$ $a_{i,2}$….$a_{i,m-1}$]→[$a_{i,m-1}$ $a_{i,0}$ $a_{i,1}$ $a_{i,2}$ ….$a_{i,m-2}$]

Step 5:
- The generating series is S: [S_0 = [$a_{0,m-1}$ $a_{0,0}$ $a_{0,1}$ $a_{0,2}$...$a_{0,m-2}$], S_1 = [$a_{1,m-1}$ $a_{1,0}$ $a_{1,1}$ $a_{1,2}$...$a_{2,m-2}$], …, S_n = [$a_{n,m-1}$ $a_{n,0}$ $a_{n,1}$ $a_{n,2}$...$a_{n,m-2}$]]

3.2. Caesar elliptic curve cryptography cryptosystem in the elliptic curve

3.2.1. Theoretical basis

The Caesar elliptic curve cryptography (CECC) cryptosystem combines the mathematical operations of the elliptic curve with an improved Caesar shift cipher. This CECC cryptosystem also relies on the algorithm for generating data series [9] to create an encryption algorithm that uses the elliptic curve on a finite field with a symmetric key to encrypt Vietnamese text. By integrating the properties of elliptic curves, the system ensures enhanced security and efficiency.

3.2.2. Description of algorithms

The CECC cryptosystem is represented in Algorithm 2 and Algorithm 3. Algorithm 2 is the encryption algorithm, and Algorithm 3 is the decryption algorithm. The output of Algorithm 2 is the ciphertext and serves as the input for Algorithm 3. The algorithm for encrypting of the CECC.

Algorithm 2. Encryption of the CECC [8]

```
BEGIN
Input: P={$p_i$} i=1...l; Key K;
        Do
                Begin
                        Input (a, b, p);
                End;
        While (($4a^3+27b^2$) mod p=0)
        n = Generating points' number in the Elliptic Curve;
        q = Generating points' set in the Elliptic Curve;
        Assign n points to n corresponding characters from generating points q;
        Assign each character of plaintext to each Elliptic Curve point;
        i=1;
        While (i<=l) do
                Begin
                        Define point $p_i$ in a table that has n points and n characters;
                        $C_i$=[($p_i$+K) mod (n)]q;
                        Generate a serie of bit from $C_i$
                        i=i+1;
                End;
Output: The ciphertext C;
END.
```

The algorithm for decrypting of the CECC. This algorithm takes as input the ciphertext received from Algorithm 2 and the parameters of the elliptic curve equation, and the key K. The decryption process follows steps to calculate the total number of points on the elliptic curve and find the generator point. Then, it iterates through each character corresponding to the points on this curve. It uses $P_i = [(C_i - K)\ mod\ (n)]q$ to retrieve the original plaintext.

Algorithm 3. Decryption of the CECC [8]

```
BEGIN
Input: $C=\{C_i\}$ i=1...l; Key K;
        a is a parameter of the Elliptic Curve;
        b is a parameter of the Elliptic Curve;
        p is a parameter of the Elliptic Curve;
n = Generating points' number in the Elliptic Curve;
q = Generating points' set in the Elliptic Curve;
i = 1;
while (i<=l) do
        Begin
                Define m characters of ciphertext;
                Shift a character to the left from the current position;
                Convert number into decimal system;
                Attach the cipher points in the Elliptic Curve;
                Define position $C_i$ in table of points in the Elliptic Curve;
                $P_i=[(C_i-K)\ mod\ (n)]q$;
                Attach plain point in the Elliptic Curve;
                Define the plaintext from the plain points attached in the Elliptic Curve;
                i=i+1;
        End;
Output: The plaintext P;
END.
```

3.3. The affine elliptic curve cryptography cryptosystem in the elliptic curve

3.3.1. Theoretical basis

The affine elliptic curve cryptography (AECC) cryptosystem integrates the mathematical operations of the elliptic curve with an improved affine cipher. Additionally, it employs the algorithm for generating data series [9] and exploits the theoretical principles of the affine cipher to create a cryptographic algorithm that is based on the elliptic curve over a finite field. This approach uses a symmetric key.

3.3.2. Description of algorithms

The AECC cryptosystem is represented in Algorithms 4 and 5. Algorithm 4 is the encryption algorithm, and Algorithm 5 is the decryption algorithm. The output of Algorithm 4 is the ciphertext and serves as the input for Algorithm 5. The algorithm for encrypting of the AECC.

Algorithm 4. Encryption of the AECC [11]

```
BEGIN
Input: $P=\{p_i\}$ i=1...l;
        Do
                Begin
                        Input (a, b, p);
                End;
        While (($4a^3+27b^2$) mod p=0)
        n = Generating points' number of the Elliptic Curve;
        q = Generating points in the Elliptic Curve;
        Assign n points to n corresponding characters from generating points q;
        Assign each character of plaintext to each point of the Elliptic Curve;
        Do
                Begin
                        Input K(u, v);
                End;
        While (UCLN(u, n) $\neq$ 1)
        i=1;
        While (i<=l) do
                Begin
                        Define $p_i$ in a table that has n points and n characters;
                        $C_i=[(u*P_i+v)\ mod\ (n)]q$;
                        Generate a serie of bit from $C_i$
                        i=i+1;
                End;
Output: The ciphertext C;
END.
```

The algorithm for decrypting of the AECC. This algorithm takes as input the ciphertext received from Algorithm 4 and the parameters of the elliptic curve equation, and the key K being a pair of values u and v. The decryption process follows steps to calculate the total number of points on the elliptic curve and find the generator point. Then, it iterates through each character corresponding to the points on this curve. It uses $Pi = [u - 1(Ci - v)\ mod\ (n)]q$ to retrieve the original plaintext.

Algorithm 5. Decryption of the AECC [11]

```
BEGIN
Input: $C=\{C_i\}$ i=1...l; Key K;
         a is a parameter of the Elliptic Curve;
         b is a parameter of the Elliptic Curve;
         p is a parameter of the Elliptic Curve;
n = Generating points' number in the Elliptic Curve;
q = Generating points' set in the Elliptic Curve;
        while (i<=l) do
                Begin
                        Define m characters of ciphertext;
                        Shift a character to the left from the current position;
                        Convert number into decimal system;
                        Attach the cipher points in the Elliptic Curve;
                        Define position $C_i$ in table of points in the Elliptic Curve;
                        $P_i=[u^{-1}(C_i-v) \bmod (n)]q$;
                        Attach plain point in the Elliptic Curve;
                        Define the plaintext from the plain points attached
                        in the Elliptic Curve;
                        i=i+1;
                End;
Output: The plaintext P;
END.
```

3.4. The Vigenère elliptic curve cryptography cryptosystem in the elliptic curve

3.4.1. Theoretical basis

The Vigenère elliptic curve cryptography (VECC) cryptosystem is founded upon the fundamental concept of the algebraic theorem of the elliptic curve. The VECC cryptosystem does not rely on algorithms for creating data series, such as those mentioned in references [8]–[10]. However, it is worth noting that algorithms for producing data series have the potential to enhance density. Nevertheless, the drawback lies in the fact that it requires a significant amount of time and memory. The study employs the concept of utilizing elliptic curve cryptography on a limited field to secure data [12].

3.4.2. Description of algorithms

The VECC cryptosystem is represented in Algorithms 6 and 7. Algorithm 6 is the encryption algorithm, and Algorithm 7 is the decryption algorithm. The output of Algorithm 6 is the ciphertext and serves as the input for Algorithm 7. The algorithm for encrypting of the VECC.

Algorithm 6. Encryption of the VECC [12]

```
BEGIN
Input: $P=\{p_i\}$i=1...l; Key $K=\{k_j\}$j=1...d;
        Do
                Begin
                        Input (a, b, p);
                End;
        While (($4a^3+27b^2$) mod p=0)
        n represents the quantity of producing points in the Curve, while q represents the
        generating points in the Elliptic Curve;
        Assign n points to n corresponding characters from generating points q;
        Assign each character of plaintext to each point of the Elliptic Curve;
        i=1;    j=1;
        While (i<=l) and (j<=d) do
           Begin
                Define $p_i$ and $k_j$ in a table that has n points and n characters;
                $C_i=[(p_i+k_j) \bmod (n)]q$;
                Define cipher points in the Elliptic Curve;
                Define ciphertext form these cipher points;
                i=i+1;
                if (j>=d)
                        Begin
                                j=1;
                        End;
                else
                        Begin
                                j=j+1;
                        End;
Output: The ciphertext C;
END.
```

The algorithm for decrypting of the VECC. This algorithm takes as input the ciphertext received from Algorithm 6 and the parameters of the elliptic curve equation, and the key K being a string of characters. The decryption process follows steps to calculate the total number of points on the elliptic curve and find the generator point. Then, it iterates through each character corresponding to the points on this curve. It uses $Pi\ = [(Ci - kj)\ mod\ (n)]q$ to retrieve the original plaintext.

Algorithm 7. Decryption of the VECC [12]

```
BEGIN
Input: C={$C_i$} i=1...l; Key K={$k_j$} j=1...d;
        a are parameters of the Curve;
        b are parameters of the Curve;
        p are parameters of the Curve;
n = Generating points’ number in the Elliptic Curve;
q = Generating points’ set in the Elliptic Curve;
while (i<=l) and (j<=d) do
       Begin
              Specify the values of $C_i$ and $k_j$ within a tabular representation of points on
              the Elliptic Curve;
              $P_i=[(C_i-k_j)\ mod\ (n)]q$;
              Attach plain point in the Elliptic Curve;
              Define the plaintext from the plain points attached in the Elliptic Curve;
              i=i+1;
              if (j>=d)
                     Begin
                            j=1;
                     End;
              else
                     Begin
                            j=j+1;
                     End;
       End;
Output: The plaintext P;
END.
```

3.5. Evaluation of protective efficiency of cryptosystems using the elliptic curve

The effectiveness of a cryptographic system that uses the elliptic curve depends on the computational complexity of discrete logarithm functions on that curve. At now, there are no algorithms that have a computational complexity greater than that of exponential functions. Since the elliptic curve lacks a division equation, we can ascertain the value of n in the equation $P = nQ$ by incrementally modifying it from 1 to n-1, using the supplied points P and Q. We go through each value of n until we find a number that makes the statement $P = nQ$ accurate. Determining the value of n inside the range of polynomials is impossible. Therefore, since we have the equation $P = nQ$, we can employ the geometric representation to establish a line that connects points P and Q, consequently establishing points (n-1) Q. Afterwards, we determine the value of n by repeatedly using a recursive method.

3.6. Comparison of cryptosystems in the elliptic curve

The proposed elliptic curve equation is $y^2 = x^3 – 2 + 3\ (mod\ 137)$ and it has 131 points. A sends a plaintext message called "*MYCOMPUTER*" to B. To secure the transmission, A encrypts this message using each of the listed algorithms. When the message is encrypted by an algorithm from a cryptosystem, it must be decrypted by another algorithm from the same cryptosystem. The metrics for measurement include time consumption, RAM usage, source code size, and computational complexity. We use the stopwatch class, employing the *Start()* method to begin timing and the *Stop()* method to end the execution of the algorithm. To determine the memory usage while running the algorithm, we use the process class along with the *GetCurrentProcess()* method and the *PrivateMemorySize64* property. The test results are presented in Table 3. The parameters of the elliptic curve with the algorithms are the same, on the same hardware device (8 GB RAM, Intel Core i7 Chip, 512 GB HDD), and in the same programming environment, the algorithmic complexity is the same. However, the VECC algorithm processes faster and consumes less memory.

Table 3. Compare cryptographic algorithms on elliptic curve

No	Cryptosystem	Consumption of time (t)	Consumption of RAM (MB)	Size of source code (Byte)	Computational complexity
1	CECC cryptosystem [8]	21	30.47 MB	36.50	$O(n * logn)$
2	AECC cryptosystem [11]	22	51.35 MB	35.98	$O(n * logn)$
3	VECC cryptosystem [12]	18	29.12 MB	35.06	$O(n * logn)$
4	ECC cryptosystem from [9]	35	31.42 MB	34.58	$O(n * logn)$
5	ECC cryptosystem from [10]	60	52.65 MB	106.94	$O(n * logn)$

4. CONCLUSION

The elliptic curve serves a vital role in cryptography applications. The robustness and level of protection provided by each suggested cryptosystem are greatly influenced by the parameters and the genesis point of the elliptic curve. Moreover, the secure key K has a substantial impact on the strength and level of security in this particular setting. We have performed an analysis and comparison between our concept and another research on the implementation of elliptic curve cryptosystems. Before evaluating the efficiency of our computer, we run all cryptographic algorithms using C# source code. This enables us to measure the amount of time it takes for various algorithms to process and ascertain their computational complexity. The results suggest that the VECC cryptosystem demonstrates remarkable efficiency, with little time and RAM memory usage, as well as a small source code size. Remarkably, all of these benefits are attained regardless of the fact that the computational complexity stays constant, as denoted by $O(n * logn)$.

ACKNOWLEDGEMENTS

The Vietnam Academy of Science and Technology (VAST) is providing funding for this research with grant number CSCL02.05/22-23.

REFERENCES

[1] H. W. Lenstra Jr, "Factoring integers with elliptic curves," *Annals of mathematics*, vol. 126, no. 3, pp. 649–673, 1987, doi: 10.2307/1971363.

[2] V. S. Miller, "Use of elliptic curves in cryptography," in *Advances in Cryptology — CRYPTO '85 Proceedings*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 417–426.

[3] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[4] R. Qazi, K. N. Qureshi, F. Bashir, N. U. Islam, S. Iqbal, and A. Arshad, "Security protocol using elliptic curve cryptography algorithm for wireless sensor networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 547–566, Jan. 2021, doi: 10.1007/s12652-020-02020-z.

[5] V. Bhuse, "Review of end-to-end encryption for social media," *International Conference on Cyber Warfare and Security*, vol. 18, no. 1, pp. 35–37, 2023.

[6] A. Chhabra and S. Arora, "An elliptic curve cryptography based encryption scheme for securing the cloud against eavesdropping attacks," in *2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC)*, Oct. 2017, pp. 243–246, doi: 10.1109/CIC.2017.00040.

[7] X. Zhao, D. Li, and H. Li, "Practical three-factor authentication protocol based on elliptic curve cryptography for industrial internet of things," *Sensors*, vol. 22, no. 19, Oct. 2022, doi: 10.3390/s22197510.

[8] B. Murugeshwari, D. Selvaraj, K. Sudharson, and S. Radhika, "Data mining with privacy protection using precise elliptical curve cryptography," *Intelligent Automation & Soft Computing*, vol. 35, no. 1, pp. 839–851, 2023, doi: 10.32604/iasc.2023.028548.

[9] E. T. Oladipupo *et al.*, "An efficient authenticated elliptic curve cryptography scheme for multicore wireless sensor network," *IEEE Access*, vol. 11, pp. 1306–1323, 2023, doi: 10.1109/ACCESS.2022.3233632.

[10] J. R. Arunkumar, S. Velmurugan, B. Chinnaiah, G. Charulatha, M. Ramkumar Prabhu, and A. Prabhu Chakkaravarthy, "Logistic regression with elliptical curve cryptography to establish secure IoT," *Computer Systems Science and Engineering*, vol. 45, no. 3, pp. 2635–2645, 2023, doi: 10.32604/csse.2023.031605.

[11] M. I. S. Ayasy and A. M. Barmawi, "Protecting author royalty of digital assets using Blockchain and elliptic curve cryptography," in *2022 International Conference on Green Energy, Computing and Sustainable Technology (GECOST)*, Oct. 2022, pp. 86–92, doi: 10.1109/GECOST55694.2022.10010412.

[12] D. M. Sharma, S. K. Shandilya, and S. C. Satapathy, "Maximizing Blockchain security: Merkle tree hash values generated through advanced vectorized elliptic curve cryptography mechanisms," *Concurrency and Computation: Practice and Experience*, vol. 35, no. 23, Oct. 2023, doi: 10.1002/cpe.7829.

[13] C. Anitha, S. S. Priscila, M. R, and B. Balakrishnan, "An efficient secure routing and hierarchical approach to elliptic curve cryptography combined with distributed group key management," in *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Mar. 2023, pp. 2208–2212, doi: 10.1109/ICACCS57279.2023.10113088.

[14] Z. Y. M. Yusoff, M. K. Ishak, L. A. B. Rahim, and O. Ali, "Elliptic curve cryptography based security on MQTT system for smart home application," in *2022 19th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2022, pp. 1–4, doi: 10.1109/ECTI-CON54298.2022.9795478.

[15] T. Kudithi and S. R, "An efficient hardware implementation of the elliptic curve cryptographic processor over prime field," *International Journal of Circuit Theory and Applications*, vol. 48, no. 8, pp. 1256–1273, Aug. 2020, doi: 10.1002/cta.2759.

[16] B. P. kavin and S. Ganapathy, "A novel M-commerce data security mechanism using elliptic curve cryptography," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 10, pp. 847–851, Aug. 2019, doi: 10.35940/ijitee.J9039.0881019.

[17] S. S. Kumar, "Elliptic curve cryptography for constrained devices," Dissertation, Faculty of Electrical Engineering and Information Technology, Ruhr-University Bochum, 2006.

[18] D. Mahto, D. A. Khan, and D. K. Yadav, "Security analysis of elliptic curve cryptography and RSA," in *Proceedings of the world congress on engineering*, 2016, vol. 1, pp. 419–422.

[19] R. Bhanot and R. Hans, "A review and comparative analysis of various encryption algorithms," *International Journal of Security and Its Applications*, vol. 9, no. 4, pp. 289–306, 2015.

[20] H. Liang, G. Zhang, W. Hou, P. Huang, B. Liu, and S. Li, "A novel asymmetric hyperchaotic image encryption scheme based on elliptic curve cryptography," *Applied Sciences*, vol. 11, no. 12, Jun. 2021, doi: 10.3390/app11125691.

[21] M. Al Saadi and B. Kumar, "A review on elliptic curve cryptography," *International Journal of Future Generation Communication and Networking*, vol. 13, no. 3, pp. 1597–1601, 2020.

[22] G. Zhang, W. Ding, and L. Li, "Image encryption algorithm based on tent delay-sine cascade with logistic map," *Symmetry*, vol. 12, no. 3, Mar. 2020, doi: 10.3390/sym12030355.

[23] M. T. Mai, P. D. Le, T. T. Le, and T. P. A. Dao, “Proposing an elliptic curve cryptosystem with the symmetric key for Vietnamese text encryption and decryption,” *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 3, pp. 4158–4162, 2020.

[24] F. Amounas and E. H. El Kinani, “ECC encryption and decryption with a data sequence,” *Applied mathematical sciences*, vol. 6, no. 10, pp. 5039–5047, 2021.

[25] D. S. Kumar, C. H. Suneetha, and A. Chandrasekhar, “Encryption of data using elliptic curve over finite fields,” *arXiv preprint arXiv:1202.1895*, 2012.

[26] M. M. Trung, L. P. Do, and D. T. Tuan, “Building elliptic curve cryptography with public key To encrypt Vietnamese text,” *Journal of Science and Technology on Information security*, vol. 1, no. 15, pp. 119–126, Jun. 2022, doi: 10.54654/isj.v1i15.854.

[27] M. M. Trung, L. P. Do, D. T. Tuan, N. Van Tanh, and N. Q. Tri, “Design a cryptosystem using elliptic curves cryptography and Vigenère symmetry key,” *International Journal of Electrical and Computer Engineering*, vol. 13, no. 2, pp. 1734–1743, Apr. 2023, doi: 10.11591/ijece.v13i2.pp1734-1743.

BIOGRAPHIES OF AUTHORS

Mai Manh Trung graduated from Vietnam National University, Hanoi. Currently a lecturer at the University of Economics Technology for Industries and researcher at the University of Engineering and Technology, Vietnam National University, Hanoi. His research area includes cryptography, lightweight cryptography, security for IoT networks, artificial intelligence, and application programming. He can be contacted at email: mmtrung@uneti.edu.vn.

Le Phe Do currently a lecturer at Faculty of Information Technology, University of Technology - Vietnam National University, Hanoi. His research area includes advanced mathematics, statistical probability, cryptography, light cryptography, and information security. He can be contacted at email: dolp@vnu.edu.vn.

Do Trung Tuan a lecturer in University of Science - Vietnam National University, Hanoi. His research area includes database, data science, data mining, and data security. He can be contacted at email: tuandt@vnu.edu.vn.

Thu Thuy Trieu received M.Sc. degree from University of Engineering and Technology, Vietnam National University in 2011. As a research scientist at the Department of Telematics, Institute of Information Technology, Vietnam Academy of Science and Technology since 2007. She has taken part in a variety of academic endeavors, including research and teaching. She can be contacted at email: thuytrieu@ioit.ac.vn.

Nguyen Van Tanh graduated from Hanoi University of Science and Technology, Vietnam. Lecturer and researcher at the International School – Vietnam National University, Hanoi. Author of much research on information security, security for IoT networks and many scientific publications in the field of IT applications. He can be contacted at email: Tanhnv@vnu.edu.vn.

Ngo Quang Tri graduated from Hanoi University of Science and Technology. As a researcher, the author publishes many scientific articles in the fields of information technology, information security and computer network communication. Currently working in the Information Technology field at 1 Faculty of Information Technology – University of Economics Technology for Industries (UNETI), Vietnam. He can be contacted at email: nqtri@uneti.edu.vn.

Bui Van Cong currently a lecturer at the University of Economics Technology for Industries (UNETI) and researcher at Post and Telecommunications Institute of Technology, Hanoi City, Vietnam. He can be contacted at email: bvcong@uneti.edu.vn.