PROPOSED ENCRYPTION BASED ON ELLIPTIC CURVES WITH LARGE PRIME NUMBERS

Mai Manh Trung[1,*], Le Phe Do[2]

DOI: https://doi.org/10.57001/huih5804.2026.081

ABSTRACT

The paper proposes an encryption method based on elliptic curves using large prime numbers to enhance security and performance. The approach focuses on selecting appropriate parameters over finite fields and improving key generation, encryption, and point computation on the curve to reduce computational cost. The system is designed to resist common attacks such as brute-force and the elliptic curve discrete logarithm problem. Experimental results show that the proposed solution achieves good performance while ensuring a high level of security, making it suitable for resource-constrained environments such as mobile devices and IoT systems. Compared to traditional methods, the proposed model demonstrates advantages in both speed and practical applicability.

Keywords: *ECC, VECC Cryptosystem, Large prime numbers, Cryptographic standards.*

[1]School of Information and Communications Technology, Hanoi University of Industry, Vietnam

[2]Faculty of Information Technology, Thang Long University, Vietnam

[*]Email: trungmm@fit-haui.edu.vn

Received: 10/01/2026

Revised: 15/3/2026

Accepted: 30/3/2026

1. INTRODUCTION

Elliptic Curve Cryptography (ECC) was first proposed by Miller and Koblitz in the years 1985 - 1987 and has since become an important approach in public-key cryptography [1]. ECC is based on the algebraic structure of points on elliptic curves defined over finite fields, typically prime fields GF(p). Compared to traditional cryptosystems such as RSA, ECC can provide an equivalent level of security with significantly smaller key sizes, thereby reducing computational and storage costs [2].

In the early stages of research, many studies focused on constructing and analyzing ECC over finite fields with small prime numbers to support theoretical validation and experimental implementation. These studies demonstrated that the elliptic curve discrete logarithm problem (ECDLP) forms the foundation of the system's security, even when the field size is relatively small [3]. At the same time, parameter selection especially the choice of the prime number p has a direct impact on the security level and resistance to attacks [4].

However, as attack methods have become increasingly sophisticated, ECC systems using small prime numbers have revealed significant limitations in terms of security. Consequently, recent research trends have shifted toward the use of larger finite fields, along with algorithmic optimizations to ensure performance in resource-constrained environments such as IoT and embedded systems [5]. Early studies on small fields still play an important role, laying the foundation for the development of more secure modern ECC systems.

The U.S. National Institute of Standards and Technology (NIST) recommends that, to ensure adequate security, ECC systems should use a prime number p of at least 160 bits [8]. In this paper, an AECC-based elliptic curve cryptosystem using a 160-bit prime number p is proposed. The VECC algorithm in previous research [7] was primarily implemented over finite fields with small sizes to demonstrate the feasibility of the proposed model. This study proposes extending VECC over large prime fields according to the recommendations of NIST and the TCVN/ISO standards on ECC in order to enhance the security level.

2. MATHEMATICAL FOUNDATIONS OF ELLIPTIC CURVES

Elliptic curves over a finite field F_p (where p is a prime number) are commonly represented in the form of the reduced Weierstrass equation:

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

Where a, b ∈ Fp satisfy the condition $4a^3 + 27b^2 \neq 0 \pmod p$ to ensure that the curve is non-singular. The set of points (x,y) satisfying the equation, together with the point at infinity ∞, forms an Abelian group with point addition defined geometrically and algebraically. This group property serves as the foundation for constructing cryptographic algorithms based on ECC.

The addition of two points P and Q on an elliptic curve is defined using the chord-and-tangent rule. If $P \neq Q$, the line passing through the two points intersects the curve at a third point, which is then reflected across the x-axis to obtain P + Q. In the case where P = Q, point doubling is performed using the tangent line at that point. The corresponding algebraic formulas enable efficient computation on computers. Repeated point addition leads to scalar multiplication kP, which is the fundamental operation in ECC-based cryptosystems.

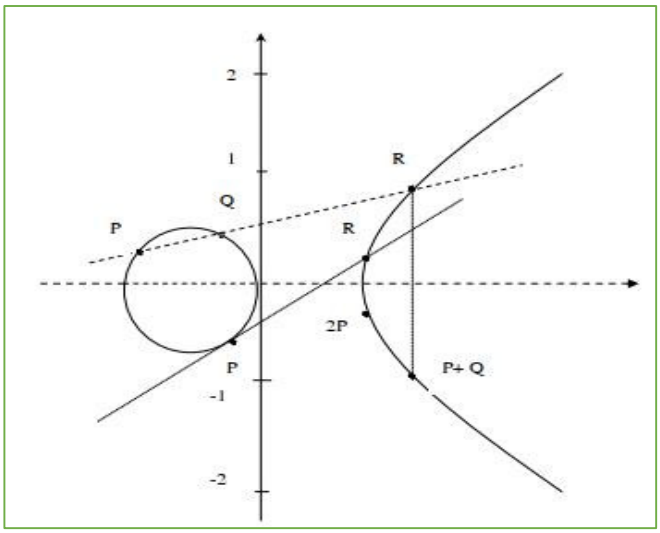

Figure 1. Summation of two points of an elliptic curve

The security of ECC is based on the difficulty of the Elliptic Curve Discrete Logarithm Problem (ECDLP), meaning that given two points P and Q, with Q = kP, it is computationally infeasible to determine the integer k when the field size is sufficiently large. Studies have shown that no efficient polynomial-time algorithm exists to solve the ECDLP for properly chosen curves. As a result, ECC provides a high level of security with smaller key sizes compared to traditional cryptosystems, making it well-suited for applications that require both efficiency and limited computational resources.

3. VECC ALGORITHM

The VECC algorithm is based on the fundamental operations of elliptic curves. This proposed VECC algorithm does not rely on data sequence generation methods as in [6], where such processes can provide higher security. However, those approaches have limitations, including (i) increased computation time and (ii) higher memory consumption. Instead, this algorithm adopts the idea of the Vigenère cipher as a foundation to construct an encryption scheme by utilizing elliptic curves over finite fields, where the positions of points on the elliptic curve are used to encode data [7].

3.1. VECC Encryption Algorithm

Algorithm 1. VECC Encryption [7]

BEGIN
Input: P = {p_i} i= 1..l; Key K = {k_j} j= 1..d;
Do
Begin
Input (a, b, p);
End;
While ($4a^3 + 27b^2 = 0$)
n = Total number of points on the elliptic curve;
q = Generator point on the elliptic curve;
Assign n points to n corresponding characters starting from the generator point q;
Map plaintext characters to points on the elliptic curve;
i = 1;
j = 1;
While (i<=l) and (j<=d) Do
Begin
Determine the positions of p_i and k_j in the table containing n points and n characters;
$C_i = [(p_i + k_j) \bmod (n)]q$;
Determine the encoded point on the elliptic curve;
Determine the corresponding ciphertext character from the encoded point;
i = i + 1;
if (j >= d)
Begin
j = 1;
End;
else
Begin
j = j + 1;
End;
Output: Ciphertext C;
END.

The result of this process is a sequence of points in which each point represents a character in the ciphertext. These points are referred to as ciphertext points. Finally, a predefined mapping table between points and characters established from the generator point is used to determine each ciphertext character from the corresponding ciphertext point. The final result is the encrypted text string.

3.2. VECC Decryption Algorithm

Algorithm 2. VECC Decryption [7]

BEGIN

Input: C = {C_i} i= 1..l; Key K = {k_j} j= 1..d;

a, b, p: elliptic curve parameters;

Compute n = total number of points on the elliptic curve;

Determine q = generator point of the elliptic curve;

while (i<=l)and (j<=d) Do

Begin

Determine the positions of C_i, k_j in the table of points on the elliptic curve;

$P_i = [(C_i - k_j) \bmod (n)]q$;

Display the corresponding point on the elliptic;

Display the plaintext character corresponding to the point on the elliptic curve;

i = i + 1;

if (j >= d)

Begin

j = 1;

End;

else

Begin

j = j + 1;

End;

End;

Output: Plaintext P;

END.

4. PROPOSED ENCRYPTION USING THE VECC CRYPTOSYSTEM WITH LARGE PRIME NUMBERS

4.1. Basis of the Proposal

According to the U.S. National Institute of Standards and Technology (NIST) [8], as well as the Vietnamese National Standard TCVN 12852-5:2020 and ISO/IEC 15946-5:2017, Information Technology – Security Techniques - Elliptic Curve Cryptography [9], Part 5: Elliptic Curve Generation recommends the use of elliptic curves over prime fields with sizes of 160 bits, 192 bits, 224 bits, 256 bits, 384 bits, and 512 bits.

To ensure the security and practical reliability of the cryptosystem, elliptic curves over prime fields with the above sizes should be used. This paper proposes an elliptic curve scheme with a large prime number of 160 bits. The equation of the elliptic curve is given as follows:

$y^2 = x^3 + 3x + 17$
(mod p = 1452046121366725933991673688168680114377396846591) (2)

The total number of points on the elliptic curve is determined as: NP = |E(Fp)| = 145204612136672593399167536834962608405830622623 03. The total number of points on the curve is a prime number.

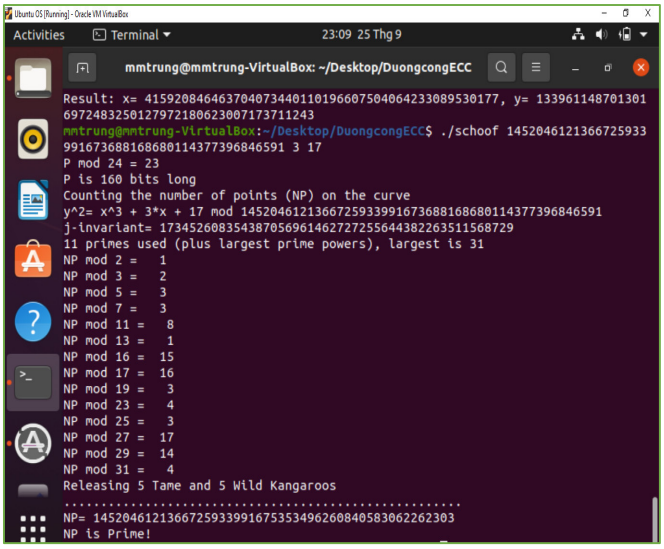

Figure 2. Experimental environment for computing the total number of points on the elliptic curve

As shown in Figure 2, the result is obtained from running the program on the open-source Ubuntu operating system, version Ubuntu 22.04.

4.2. Applying The VECC Algorithm

To perform encryption with the initial plaintext P = "cake" and key K = "done", using the elliptic curve parameters in equation (2), we select a = 3 and b = 17, with the prime number p = 145204612136672593399167368816868011437739 6846591. The generator point is chosen as: q = (569557196776379466754762268181294281256267175372, 701857423514298794065989459648164364342982892341).

Table 1. characters corresponding to points generated from the generator point q of equation (2)

Character/ Index	Point
a	(569557196776379466754762268181294281256267175372, 701857423514298794065989459648164364342982892341)
b	(1174991732670464645014017983012687311537860275016, 830598146533707890402121631387554567569778635718)
c	(1592760367622599208107355543171300984333206957 91, 208289701002424479461940834817959320319307155184)
d	(189755783596209521814462600303002715468724619 66, 601688388967696097062945536317482720122026809929)
e	(362942879791976215388178925535093400153749798084, 875259234924510488330379124064328760278964790003)
f	(116030632827947106129366922756556414671411413459 5, 100393651143419330368956669023959820318324568131 9)
g	(125187144354402479083403597858724230127164447495 1, 876763685484231533876198395698096024595554811942)
h	(458418540088347660151084698270147325644718087044, 433837537463107326452967996850535438565450149754)
i	(116666005755765813883058997392821063890826431747 8, 106209320568406494037651662139214622804086524270 0)
j	(525622826749772322749599095406221854701127118145, 117524460573947853153858398616613208013377224629 3)
k	(990685996707120479445508594227295401065491916739, 304637333858276881611057524053500054940595891029)
l	(113704274379402393307080996978892521553548232025 1, 135421408318369016698528280090867737656652721882 6)
m	(598899577098812947478133518507841330222722090161, 526865549257065815872918492373838699438808122155)
n	(114293276212822039183154913883221958545108868527 8, 949018172034019723538068923361815754416163242602)
o	(114574952042237542279202262713829943629471707931 3, 120887439739041408617343101191576093850515095431 6)
…	……
NP-1	(569557196776379466754762268181294281256267175372, 750188697852427139925684228520515750034413954250)
NP	∞

With the key K = "done", the key consists of (n) characters and is selected randomly. In this case, the illustrative key "done" is used. Each character of the key corresponds to a point on the elliptic curve. These are represented as shown in Table 2.

Table 2. Key characters corresponding to points on the elliptic curve (2)

Character	Point
d	(189755783596209521814462600303002715468724619 66, 601688388967696097062945536317482720122026809929)
o	(114574952042237542279202262713829943629471707931 3, 120887439739041408617343101191576093850515095431 6)
n	(114293276212822039183154913883221958545108868527 8, 949018172034019723538068923361815754416163242602)
e	(362942879791976215388178925535093400153749798084, 875259234924510488330379124064328760278964790003)

Apply the encryption function to obtain the ciphertext. By determining the corresponding points according to Table 1, the plaintext characters corresponding to those points are obtained, as shown in Table 3.

Table 3. Plaintext characters corresponding to points on the elliptic curve (2)

Plaintext	Position of the point on E
c	(1592760367622599208107355543171300984333206957 91, 208289701002424479461940834817959320319307155184)
a	(569557196776379466754762268181294281256267175372, 701857423514 298794065989459648164364342982892341)
k	(990685996707120479445508594227295401065491916739, 304637333858276881611057524053500054940595891029)
e	(362942879791976215388178925535093400153749798084, 875259234924510488330379124064328760278964790003)

(i) Consider the character "c": the corresponding point P_1 of "c" is 3q, associated with the point (1592760367622599208107355543171300984333206957 91, 208289701002424479461940834817959320319307155184) we obtain:

C_1 = [(3 + 4) mod 145204612136672593399167535349626084058306226230 3]q = 7(5695571967763794667547622681812942812562670 1753 72,701857423514298794065989459648164364342982892341) = (12518714435440247908340359785872423012716444749 51, 876763685484231533876198395698096024595554811942) which corresponds to the character "g".

(ii) Similarly, consider the character "a": the corresponding point P_2 of "a" is 1q, associated with the point (5695571967763794667547622681812942812562671751 372, 701857423514298794065989459648164364342 9828 92341) we obtain:

C_2 = [(1 + 15) mod 14520461213667259339916753534962608405830622623 03]q = 16(56955719677637946675476226818129428125626717537 2, 70185742351429879406598945964816436434298289234 1) = (8010513491066603856754381417402633378047761148 88, 75384442254144670135392269591488990863827940273 5) which corresponds to the character "p". (iii) Similarly, for the remaining characters of the plaintext, the resulting ciphertext characters are "y" and "j". After encryption, the final ciphertext obtained is "gpyj". This ciphertext is then transmitted to the receiver over the communication channel. Upon receiving the ciphertext, the receiver performs decryption. The decryption process using Algorithm 2 is described as follows:

- Use the elliptic curve parameters with a = 3 and b = 17, and the prime number p = 14520461213667259339916736881686801143773968465 91. Use the generator point: q = (5695571967763794667547622681812942812562671753 72, 70185742351 42987940659894596481643643429828923 41).

- Convert the ciphertext characters into their corresponding points. The results are shown in Table 4.

Table 4. Ciphertext characters corresponding to points on the elliptic curve (2)

Ciphertext	Position of the point on E
g	(125187144354402479083403597858724230127164447495 1, 876763685484231533876198395698096024595554811942)
p	(8010513491066603856754381417402633378047761148 88, 75384442254144670135392269591488990863827940273 5)
y	(6588902844258731390830235753881159922382620026 7, 960812499594301701680941607208533656326896936020)
j	(5256228267497723227495990954062218547011271181 45, 1175244605739478531538583986166132080133772246293)

- Input the decryption key K= "done", which is the same as the encryption key.

- Apply the decryption function: Pi = [(c_i – k_j) mod (n)]q. To recover the plaintext.

(i) Consider the ciphertext character "g", which corresponds to the point (125187144354402479083403597858724230127164447495 1, 876763685484231533876198395698096024595554811942) which corresponds to (7q). We have: P_1 = [(7 - 4) mod 14520461213667259339916753534962608405830622623 03]q = 3q = 3(56955719677637946675476226818129428125626717537 2, 70185742351429879406598945964816436434298289234 1) = (1592760367622599208107355543171300984333206957 91, 20828970100242447946194083481795932031930715518 4) which corresponds to the character "c".

(ii) Consider the ciphertext character "p", which corresponds to the point điểm (8010513491066603856754381417402633378047761148 88, 75384442254144670135392269591488990863827940273 5) which corresponds to 16q. We have P_2 = [(16 - 15) mod 14520461213667259339916753534962608405830622623 03]q = 1q = (56955719677637946675476226818129428125626717537 2, 70185742351429879406598945964816436434298289234 1)

which corresponds to the character "a". Similarly, for the remaining ciphertext characters, we obtain the corresponding plaintext characters "k" and "e". Therefore, the original plaintext is recovered as "cake".

By using an elliptic curve equation with a large prime number, the proposed approach complies with the recommendations of NIST as well as the Vietnamese national standards on elliptic curve cryptography techniques. Experimental Setup: Operating system: Ubuntu 22.04 LTS; Programming languages: C++ and Python; CPU: Intel Core i5, 8 GB RAM; Elliptic curve: prime field 160-bit; Test message lengths: 32, 64, 128, 256 characters

5. EVALUATION

The proposed encryption method based on elliptic curves with a 160 bit large prime demonstrates several advantages in terms of both security and performance. First, the use of a large prime significantly increases the difficulty of the elliptic curve discrete logarithm problem (ECDLP), thereby enhancing resistance against cryptanalytic attacks, especially brute-force and approximation methods. This aligns with recommendations from standards such as NIST and recent studies on ECC [8, 9].

In addition, experimental results show that the VECC algorithm, when applied to the proposed curve, still maintains good computational performance despite the larger field size compared to previous studies. Optimization of point operations and appropriate parameter selection help reduce computational cost, enabling the system to be deployed in resource-constrained environments such as IoT devices. Compared to ECC models using small prime numbers, this approach achieves a better balance between performance and security.

However, the use of large prime numbers also increases implementation complexity and requires higher computational resources. Therefore, further

optimization techniques at both the hardware and algorithmic levels are necessary to ensure broader practical applicability.

Experimental results indicate that the point generation stage incurs a one-time initialization cost, while encryption and decryption times increase approximately linearly with message length. The proposed method maintains acceptable processing performance despite operating over a large prime field.

Table 5. Compared to the old VECC

Method	Field size	Encryption time (128 chars)	Security level
Original VECC [7]	Small prime field	5.1ms	Moderate
Proposed VECC	160-bit prime field	7.2ms	High

Security Analysis Using Large Prime Numbers:

Brute-Force Attack

- The key space increases with the size of the large prime field: 2^{160}

- Therefore, exhaustive key search becomes computationally infeasible.

ECDLP Attack

- The computational complexity of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP) using the Pollard-rho algorithm is: $O(\sqrt{n})$.

- When a 160-bit prime field is used, the computational cost becomes extremely high.

Table 6. Comparison with NIST standard security levels

ECC Cryptosystem	Key Size	Security Level
ECC-160	160 bit	~1024 RSA
ECC-224	224 bit	~2048 RSA
ECC-256	256 bit	High

6. CONCLUSION

This paper has proposed an encryption method based on elliptic curves using a 160 bit large prime number to enhance information security. Through the establishment of the mathematical foundation, appropriate parameter selection, and the application of the VECC algorithm, the model has demonstrated its feasibility in both data encryption and decryption. Experimental results show that the proposed method not only meets high security requirements but also maintains acceptable computational performance. The proposed this method shows strong potential for practical deployment in modern secure communication systems.

The main contribution of this study is the proposal of an approach that utilizes large prime fields combined with algorithm optimization to strengthen resistance against attacks in the context of increasingly sophisticated security threats. In the future, the research can be extended toward hardware-level optimization, as well as applications in real-world systems such as IoT, blockchain, and secure communication systems, along with more in-depth evaluation against advanced attack models.

REFERENCES

[1]. Hamad Marzouqi, Mahmoud Al-Qutayri, Khaled Salah, "Review of Elliptic Curve Cryptography processor designs," *Microprocessors and Microsystems*, 39, 2, 97-112, 2015. https://doi.org/10.1016/j.micpro.2015.02.003

[2]. Yue Hao, Shun'an Zhong, Mingzhi Ma, Rongkun Jiang, Shihan Huang, Jingqi Zhang, Weijiang Wang, "Lightweight Architecture for Elliptic Curve Scalar Multiplication over Prime Field," *Electronics*, 11(14), 2234, 2022. https://doi.org/10.3390/electronics11142234

[3]. Adrian O'Gara, *Investigation into the Cryptographic Properties of Elliptic Curves Defined over a Prime Field.* Grin Verlag Publisher, 2015

[4]. Marwan Alshar'e, Sharf Alzu'bi, Ahed Al-Haraizah, Hamzah Ali Alkhazaleh, Malik Jawarneh, Mohammad Rustom Al Nasar, "Elliptic curve cryptography based light weight technique for information security," *Electrical Engineering and Informatics (BEEI),* 14, 3, 2300~2308, 2025. DOI: 10.11591/eei.v14i3.8587, 2025

[5]. Abidemi Emmanuel Adeniyi, Rasheed Gbenga Jimoh, Joseph Bamidele Awotunde, "A systematic review on elliptic curve cryptography algorithm for internet of things: Categorization, application areas, and security," *Computers and Electrical Engineering*, 118, Part A, 2024.

[6]. F.Amounas, E.H.El.Kinani, "ECC Encryption and Decryption with a Data Sequence, Applied Mathematical Sciences,", *Applied Mathematical Sciences,* 6, 101, 5039 - 5047, 2012.

[7]. Mai Manh Trung, Le Phe Do, Do Trung Tuan, Nguyen Van Tanh, Ngo Quang Tri, "Design a cryptosystem using elliptic curves cryptography and sysmetry key," *International Journal of Electrical and Computer Engineering (IJECE)*, 13, 2, 1734~1743, 2023. DOI: 10.11591/ijece.v13i2.pp1734-1743

[8]. Meltem Sonmez Turan, Kerry McKay, Donghoon Chang, Lawrence E. Bassham, Jinkeon Kang, Noah D. Waller, John M. Kelsey Deukjo Hong, "Status Report on the Final Round of the NIST Lightweight Cryptography Standardization Process," *NIST Internal Report NIST IR 8454*, National Institute of Standards and Technology, 2023. https://doi.org/10.6028/NIST.IR.8454.

[9]. TCVN 12852-5: 2020, ISO/IEC 15946-5: 2017, *Information technology - Security techniques - Cryptography based on elliptic curves - Part 5: Elliptic curve generation.* Hanoi, 2020.