ĐỀ XUẤT CẢI TIẾN HỆ MẬT MÃ AECC THÔNG QUA VỊ TRÍ ĐIỂM TRÊN ĐƯỜNG CONG ELLIPTIC

THE IMPROVEMENT OF AECC CRYTOSYSTEM BY POINTS ON THE ELLIPTIC CURVE: A PROPOSAL

Mai Mạnh Trừng[1*], Ngô Quang Trí[1], Lê Thị Thu Hiền[1], Lê Trung Thực[2]

[1]Khoa Công nghệ thông tin, Trường Đại học Kinh tế - Kỹ thuật Công nghiệp

[2]Khoa Công nghệ thông tin, Trường Đại học Công nghệ Đông Á

Đến Tòa soạn ngày 20/02/2023, chấp nhận đăng ngày 10/04/2023

Tóm tắt: Hệ mật mã AECC trên đường cong elliptic sử dụng thuật toán sinh chuỗi dữ liệu, sau đó sử dụng thuật toán để mã hóa và giải mã. Khi sử dụng thuật toán sinh chuỗi dữ liệu thì ưu điểm là tăng thêm độ phức tạp khi thám mã. Tuy nhiên, điều này sẽ dẫn đến tốn dung lượng và thời gian. Trong bài báo này nhóm nghiên cứu đề xuất cải tiến hệ mật mã này không cần sinh chuỗi dữ liệu để mã hóa mà chỉ cần lấy vị trí của điểm tương ứng ký tự để mã hóa. Với việc này thì bản mã ngắn gọn hơn khi gửi bản mã trên mạng sẽ chiếm ít băng thông trên quá trình truyền.

Từ khóa: Mật mã đường cong elliptic, bảo mật, chuỗi dữ liệu.

Abstract: The AECC cryptosystem on elliptic curves uses an algorithm to generate a data series before launches the process of encryption and decryption. The advantage of the data sequence generation algorithm is the increase of the cryptanalysis complexity but it has disadvantage which is the outrageous consumption of memory and processing time. In this paper, the research team proposed a solution to improve this cryptosystem by eliminating the generation of data series in purpose of encryption. Instead, the proposed algorithm uses index of point corresponding character for encrypting. The result is the size of the ciphertext is lower and the use of bandwidth for transmitting this ciphertext decreases.

Keywords: Elliptic curve cryptography, Security, Data sequence.

1. GIỚI THIỆU

Nghiên cứu về đường cong elliptic của các nhà đại số, các nhà lý thuyết số có từ giữa thế kỷ XIX. Mật mã đường cong Elliptic Curve Cryptography (ECC) được phát hiện vào năm 1985 bởi Neil Koblitz và Victor Miller [1, 2]. Chúng có thể được xem như các đường cong elliptic của các hệ mật mã logarit rời rạc.

Những năm gần đây, ở Việt Nam, đường cong elliptic có vai trò quan trọng, theo Thông tư số 39/2017/TT-BTTTT, ngày 15/12/2017 của Bộ Thông tin và Truyền thông về việc Ban hành Danh mục tiêu chuẩn kỹ thuật ứng dụng công nghệ thông tin trong cơ quan nhà nước đã khuyến nghị áp dụng giải thuật mã hóa trên đường cong Eelliptic của Tiêu chuẩn về an toàn thông tin.

Trên thế giới cũng có nhiều ứng dụng [3, 4, 5] sử dụng đường cong elliptic để đảm an toàn thông tin. Mật mã đường cong elliptic được sử dụng trong Chính phủ Hoa Kỳ để bảo vệ thông tin liên lạc nội bộ. Theo nghiên cứu [10, 11] chỉ mới dừng ở đưa ra ý tưởng về đường cong elliptic, với nghiên cứu [12] nghiên cứu rất sâu về lý thuyết đường cong elliptic nhưng cũng chưa vận dụng vào để bảo mật thông tin,

với nghiên cứu [6] sử dụng thuật toán sinh chuỗi sau đó sử dụng hàm mã hóa để mã hóa dữ liệu. Với nghiên cứu [7, 8] sử dụng thuật toán mã hóa mới được đề xuất và cũng sử dụng ý tưởng tạo chuỗi dữ liệu để mã hóa. Nghiên cứu này đã cải tiến so với nghiên cứu [8] là không sử dụng kỹ thuật sinh chuỗi dữ liệu mà lấy vị trí điểm của ký tự. Bởi vì nếu sinh chuỗi sẽ tạo ra không gian dữ liệu lớn làm ảnh hưởng băng thông trên quá trình truyền bản mã.

Hiện nay, hệ mật RSA là giải thuật khoá công khai được sử dụng nhiều, nhưng hệ mật dựa trên đường cong elliptic (ECC) có thể thay thế cho RSA bởi mức an toàn và tốc độ xử lý cao hơn. Ưu điểm của ECC là hệ mật mã này sử dụng khóa có độ dài nhỏ hơn so với RSA nhưng độ bảo mật là như nhau như bảng 1.

Bảng 1. Mật mã khóa đối xứng và khóa công khai [9]

Symmetric-key	ECC	RSA/DLP
64 bit	128 bit	700 bit
80 bit	160 bit	1024 bit
128 bit	256 bit	2048-3072 bit

2. CƠ SỞ TOÁN HỌC CỦA ĐƯỜNG CONG ELLIPTIC

Gọi K là một trường hữu hạn hoặc vô hạn. Một đường cong Elliptic được định nghĩa trên trường K bằng công thức Weierstrass:

$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$, trong đó $a_1, a_2, a_3, a_4, a_6 \in K$. Đường cong elliptic trên trường K được ký hiệu E(K). Số lượng các điểm nguyên trên E ký hiệu là #E(K), có khi chỉ đơn giản là #E. Đối với từng trường khác nhau, công thức Weierstrass có thể được biến đổi và đơn giản hóa thành các dạng khác nhau.

Đường cong elliptic trên trường số thực R là tập hợp các điểm (x, y) thoả mãn công thức:

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

trong đó a, b là số nguyên modulo p, thỏa mãn: $4a^3 + 27b^2 \neq 0$ đảm bảo rằng là đường cong elliptic. Tức là, không có điểm nào đó của đường cong có hai hoặc nhiều đường tiếp tuyến khác biệt. Cùng với một điểm đặc biệt ∞ được gọi là điểm vô cực. Cặp giá trị (x, y) đại diện cho một điểm trên đường cong elliptic và tạo nên mặt phẳng tọa độ hai chiều $R \times R$. Đường cong elliptic trên R^2 được gọi là định nghĩa trên R, ký hiệu là E(R). Đường cong elliptic trên R sử dụng hai toán là phép cộng điểm và phép nhân điểm.

3. HỆ MẬT MÃ AECC

3.1. Thuật toán sinh chuỗi

Thuật toán 3.1. Sinh chuỗi [6]

Input: Tham số đường cong elliptic

Output: Chuỗi các bit

Bước 1:

- Tính tổng số điểm (n) trên đường cong elliptic
- Xác định điểm q là điểm sinh của phương trình đã cho
- Đưa ra tập các điểm trên đường cong elliptic từ điểm sinh q

Bước 2:

- Chuyển đổi tổng số điểm (n) trong cơ số 3
- Lấy m sẽ là số chữ số chuyển tổng số điểm sang cơ số 3

Bước 3:

- Lập ma trận M có kích thước (n + 1) * m. Ở đây (n + 1) là số hàng, m là số cột cũng chính là số chữ số trong một hàng.

$$M = \begin{pmatrix} a_{0,0}a_{0,1}\dots \dots a_{0,m} \\ a_{1,0}a_{1,1}\dots \dots a_{1,m} \\ a_{1,0}a_{1,1}\dots \dots a_{1,m} \\ \dots \dots \\ a_{n,0}a_{n,1}\dots \dots a_{n,m} \end{pmatrix}$$

Bước 4:

- Dịch chuyển theo vòng hàng của ma trận ở bước 3 theo một phần tử sang bên phải: $[a_{i,0}\ a_{i,1}\ a_{i,2}\dots.a_{i,m-1}] \rightarrow [a_{i,m-1}\ a_{i,0}\ a_{i,1}\ a_{i,2}\ \dots.a_{i,m-2}]$

Bước 5:
Chuỗi được hình thành là: 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. Ý tưởng thuật toán AECC

Thuật toán AECC dựa vào phép toán trên đường cong elliptic kết hợp với hệ mật mã Affine cải tiến. Thuật toán này dựa vào thuật toán tạo chuỗi dữ liệu [6], rồi sử dụng ý tưởng mật mã Affine làm cơ sở để xây dựng thuật toán mã hóa bằng cách sử dụng đường cong elliptic trên trường hữu hạn có khóa đối xứng.

3.3. Thuật toán mã hóa AECC [8]

Thuật toán 3.2. Mã hóa AECC

```
BEGIN
Input: P = {p_i} i= 1..l;
   Do
       Begin
           Input (a, b, p);
       End;
   While ((4a^3 + 27b^2) mod p = 0)
   n = Tổng số điểm trên đường cong elliptic;
   q = Điểm sinh trên đường cong elliptic;
Gán n điểm với n ký tự tương ứng, tính từ điểm sinh q;
Gán các ký tự bản rõ với điểm trên đường cong elliptic;
   Do
       Begin
           Input K(u, v);
       End;
   While (UCLN(u, n) ≠ 1)
   i = 1;
   While (i<=l) do
      Begin
         Xác định vị trí p_i trong bảng có n điểm, n ký tự;
         C_i = [(u*P_i + v) mod (n)]q;
         Sinh chuỗi bit từ C_i
         i = i + 1;
      End;
Output: Xuất ra bản mã C;
END.
```

3.4. Thuật toán giải mã hóa AECC [8]

Thuật toán 3.3. Giải mã AECC

```
BEGIN
Input: C = {C_i} i= 1..l; Khóa K;
    a, b, p tham số đường cong elliptic;
Tính n = tổng số điểm trên đường cong elliptic;
Xác định q = điểm sinh của đường cong elliptic;
   while (i<=l) do
      Begin
         Xét đoạn gồm m chữ số của bản mã;
         Sử dụng phép dịch trái một phần tử;
         Chuyển đổi sang thập phân;
         Hiển thị mã điểm trên đường cong elliptic;
         Xác định vị trí C_i trong bảng điểm trên đường cong elliptic;
         P_i = [u^-1 (C_i – v) mod (n)]q;
         Hiển thị rõ điểm trên đường cong elliptic;
         Hiển thị ký tự bản rõ ứng với điểm trên đường cong elliptic;
         i = i + 1;
      End;
 Output: Xuất ra bản rõ P;
END.
```

4. CẢI TIẾN HỆ MẬT MÃ AECC

Nhóm nghiên cứu đề xuất không dựa vào thuật toán tạo chuỗi dữ liệu như [6], quá trình tạo chuỗi dữ liệu này thì độ mật sẽ tốt hơn. Tuy nhiên, mặt hạn chế là sẽ tốn nhiều thời gian và dung lượng bộ nhớ. Đặt tên hệ mật mã AECC cải tiến là AECC*.

4.1. Thuật toán mã hóa AECC cải tiến

Thuật toán 4.1. Thuật toán mã hóa AECC*

```
BEGIN
    Input: P = {p_i} i= 1..l;
   Do
      Begin
         Input (a, b, p);
      End;
   While ((4a^3 + 27b^2) mod p = 0)
   n = Tổng số điểm trên đường cong elliptic;
   q = Điểm sinh trên đường cong elliptic;
   Gán n điểm với n ký tự tương ứng, tính từ điểm sinh q;
   Gán các ký tự bản rõ với điểm trên đường cong elliptic;
   Do
       Begin
           Input K(u, v);
       End;
   While (UCLN(u, n) ≠ 1)
```

```
   i = 1;
   While (i<=l) do
      Begin
         Xác định vị trí p_i trong bảng có n
      điểm, n ký tự;
         C_i = [(u*P_i + v) mod (n)]q;
         Tìm được mã điểm trên đường cong
      elliptic;
      Tìm được mã ký tự theo mã điểm;
         i = i + 1;
      End;
 Output: Xuất ra bản mã C;
END.
```

4.2. Thuật toán giải mã AECC cải tiến

Thuật toán 4.2. Thuật toán giải mã AECC*

```
BEGIN
Input: C = {C_i} i= 1..l; Khóa K;
    a, b, p tham số đường cong elliptic;
Tính n = tổng số điểm trên đường cong
elliptic;
Xác định q = điểm sinh của đường cong
elliptic;
while (i<=l)do
  Begin
      Xác định vị trí C_i trong bảng của điểm
      trên đường cong elliptic;
      P_i = [u^-1 (C_i – v) mod (n)]q;
      Hiển thị rõ điểm trên đường cong
   elliptic;
      Hiển thị ký tự bản rõ ứng với điểm trên
   đường cong elliptic;
      i = i + 1;
   End;
Output: Xuất ra bản rõ P;
END.
```

5. ÁP DỤNG THUẬT TOÁN AECC*

Phương trình đường cong Elliptic đề xuất là:

$$y^2 = x^3 - 2x + 3 \bmod 137 \qquad (2)$$

Với phương trình (2) thì a = –2, b = 3, ta có $4\times(-2)^3 + 27\times(3)^2 = 211 \neq 0$. Do vậy, phương trình (2) là phương trình đường cong elliptic. Bên A gửi cho bên B một bản rõ (văn bản đầu vào) là SECURITY. Để đảm bảo bí mật trên quá trình truyền, bên A sẽ mã hóa bản rõ trên trước khi gửi trên kênh truyền. Quá trình mã hóa được thể hiện như sau:

Theo thuật toán 4.1 với đường cong elliptic ở (2) ta có 131 điểm trên đường cong tính cả điểm vô cực. Ta tìm được điểm sinh q = (23, 43). Sử dụng phép toán cộng điểm, nhân điểm trên đường cong Elliptic ta được tập hợp các điểm tính từ điểm sinh. Tiếp theo ta gán các ký tự ứng với các điểm ta được kết quả như bảng 2.

Bảng 2. Tập hợp ký tự ứng với tất cả các điểm trên đường cong xét từ điểm q

(23, 43) **a**	(63, 131) **à**	(12, 36) **ã**	(39, 34) **ả**
(2, 125) **á**	(53, 35) **ạ**	(111, 17) **ă**	(20, 90) **ằ**
(35, 8) **ẵ**	(117, 117) **ẳ**	(5, 114) **ắ**	(92, 115) **ặ**
(71, 32) **â**	(82, 59) **ầ**	(17, 19) **ẫ**	(113, 8) **ẩ**
(65, 19) **ấ**	(136, 2) **ậ**	(131, 115) **b**	(90, 95) **c**
(8, 122) **d**	(1, 106) **đ**	(36, 125) **e**	(134, 121) **è**
(126, 129) **ẽ**	(93, 126) **ẻ**	(84, 98) **é**	(99, 12) **ẹ**
(40, 9) **ê**	(78, 67) **ề**	(51, 22) **ễ**	(55, 118) **ể**
(135, 37) **ế**	(88, 51) **ệ**	(48, 112) **f**	(62, 85) **g**
(41, 43) **h**	(73, 94) **i**	(49, 62) **ì**	(101, 37) **ĩ**
(43, 85) **ỉ**	(85, 46) **í**	(13, 121) **ị**	(118, 13) **k**
(72, 59) **l**	(76, 4) **m**	(66, 30) **n**	(121, 130) **o**
(15, 34) **ò**	(81, 63) **õ**	(132, 132) **ỏ**	(80, 55) **ó**
(83, 103) **ọ**	(32, 85) **ô**	(119, 57) **ồ**	(4, 14) **ỗ**
(74, 45) **ổ**	(38, 37) **ố**	(120, 78) **ộ**	(44, 61) **ơ**
(26, 111) **ờ**	(69, 56) **ỡ**	(59, 60) **ở**	(127, 121) **ớ**
(116, 127) **ợ**	(116, 10) **p**	(127, 16) **q**	(59, 77) **r**
(69, 81) **s**	(26, 26) **t**	(44, 76) **u**	(120, 59) **ù**
(38, 100) **ũ**	(74, 92) **ủ**	(4, 123) **ú**	(119, 80) **ụ**

(32, 52) **ư**	(83, 34) **ừ**	(80, 82) **ữ**	(132, 5)20 **ử**
(81, 74) **ứ**	(15, 103) **ự**	(121, 7) **v**	(66, 107) **x**
(76, 133) **y**	(72, 78) **ỳ**	(118, 124) **ỹ**	(13, 16) **ỷ**
(85, 91) **ý**	(43, 52) **ỵ**	(101, 100) **z**	(49, 75) **0**
(73, 43) **1**	(41, 94) **2**	(62, 52) **3**	(48, 25) **4**
(88, 86) **5**	(135, 100) **6**	(55, 19) **7**	(51, 115) **8**
(78, 70) **9**	(40, 128) **dấu cách**	(99, 125) **_**	(84, 39) **=**
(93, 11) **[**	(126, 8) **]**	(134, 16) **;**	(36, 12) **‘**
(1, 31) **,**	(8, 15) **.**	(90, 42) **!**	(131, 22) **?**
(136, 135) **@**	(65, 118) **$**	(113, 129) **%**	(17, 118) ^
(82, 78) **|**	(71, 105) **&**	(92, 22) **#**	(5, 23) +
(117, 20) **-**	(35, 129) *****	(20, 47) **:**	(111, 120) **/**
(53, 102) **(**	(2, 12) **)**	(39, 103) **{**	(12, 101) **}**
(63, 6) <	(23, 94) >	∞	

- Chọn khóa ngẫu nhiên là K = (7, 23).
- Rõ điểm: Theo bảng 2 ta có được các ký tự bản rõ tương ứng với số điểm cho kết quả ở bảng 3.

Bảng 3. Ký tự ứng với điểm trên đường cong

S	E	C	U	R	I	T	Y
(69, 81)	(36, 125)	(90, 95)	(44, 76)	(59, 77)	(73, 94)	(26, 26)	(76, 133)

- Áp dụng hàm mã hóa: $C_i = [(u \times P_i + v) \bmod (n)]q$.

Xét ký tự ‘S’: Ta được P_1 của ‘S’ là 69q ứng với điểm (69, 81).

Ta có C_1 = [7*69+23) mod 131]q = 113q = 113(23, 43) = (136, 135), điểm này tương ứng với ký tự ‘@’.

Ta xét ký tự tiếp theo, ta có C_2 = [(7*23+23) mod 131]q = 53q = 53(23, 43) = (83, 103), điểm này tương ứng với ký tự ‘Ọ’. Tương tự dùng hàm mã hóa ta xác định được các ký tự mã hóa còn lại, ta được kết quả như bảng 4.

Bảng 4. Bảng các ký tự sau khi mã hóa

Ký tự	Rõ điểm	Mã điểm	Bản mã
S	(69, 81)	(136, 135)	@
E	(36, 125)	(83, 103)	Ọ
C	(90, 95)	(55, 118)	Ế
U	(44, 76)	(39, 103)	{
R	(59, 77)	(126, 8)	]
I	(73, 94)	(84, 98)	É
T	(26, 26)	(5, 23)	+
Y	(76, 133)	(41, 94)	2

Ta được chuỗi mã hóa là: “@ỌẾ{]É+2”. Bản mã này được gửi trên kênh truyền cho bên B.

Giải mã:

Khi bên B nhận được bản mã và tiến hành giải mã như sau:

Sử dụng tham số đường cong elliptic, điểm sinh và khóa như phần mã hóa. Tiếp theo, sử dụng hàm giải mã: $P_i = [u^{-1} (C_i - v) \bmod (n)]q$;

Xét ký tự đầu của chuỗi mã hóa là ký tự ‘@’, ký tự này ứng với điểm (136, 135) có vị trí là 113q trên đường cong elliptic đã cho. Lúc đó ta có:

P_1 = [(7^{-1}*(113 - 23) mod 131]q =69q= 69(23, 43) = (69, 81) ứng với ký tự ‘S’.

Tương tự xét ký tự tiếp theo của chuỗi mã hóa, ký tự này ứng với điểm (83, 103) có vị trí 53q trên đường cong elliptic đã cho, ta có:

P_2 =[(7^{-1}*(53 – 23) mod 131]q =23q= 23(23, 43) = (36, 125) ứng với ký tự ‘E’.

Tương tự với các ký tự còn lại của bản mã ta được kết quả giải mã như bảng 5:

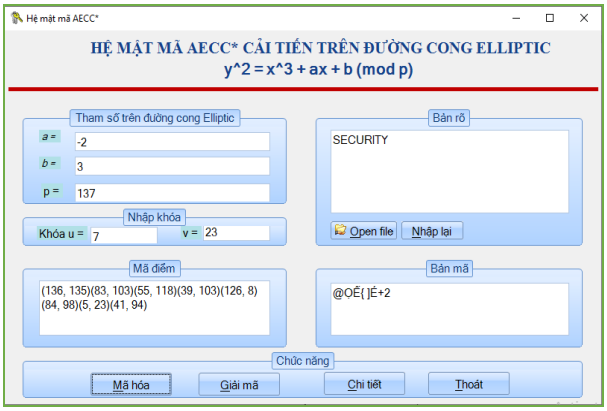
Bảng 5. Bảng kết quả giải mã

Bản mã	Mã điểm	Rõ điểm	Ký tự
@	(136, 135)	(69, 81)	S
Q	(83, 103)	(36, 125)	E
Ế	(55, 118)	(90, 95)	C
{	(39, 103)	(44, 76)	U
]	(126, 8)	(59, 77)	R
É	(84, 98)	(73, 94)	I
+	(5, 23)	(26, 26)	T
2	(41, 94)	(76, 133)	Y

Vậy ta được bản rõ ban đầu là: SECURITY

5. CÀI ĐẶT CHƯƠNG TRÌNH

Thuật toán được cài đặt trên thiết bị với cấu hình phần cứng là: CPU Intel(R) Core(TM) i5, 2.5 GHZ; RAM: 4GB; HDD: 500 GB; và phần mềm với Hệ điều hành Windows 10, môi trường lập trình Visual studio .NET-2021. Chương trình thực hiện cài đặt thuật toán mã hóa và giải mã trên đường cong Elliptic dùng ngôn ngữ lập trình C# của Visual studio .NET -2021 với giao diện như hình 1. Chương trình chạy cho kết quả đúng đắn với thuật toán đã trình bày ở trên.



Hình 1. Giao diện chương trình

6. ĐÁNH GIÁ ĐỘ AN TOÀN BẢO MẬT, ĐỘ PHỨC TẠP THUẬT TOÁN AECC*

Tính an toàn bảo mật của AECC* dựa trên độ phức tạp của bài toán Logarit rời rạc trên đường cong elliptic. Hiện chưa thuật toán nào có khả năng tính toán với thời gian nhỏ hơn cấp lũy thừa. Do không tồn tại phép chia trên đường cong elliptic, nên với P = nQ, khi cho chúng ta điểm P và một điểm khởi đầu Q, cách để tìm ra số n thường là thử lần lượt n = 1, 2, ... n-1 đến khi tìm được kết quả nQ = P. Về cơ bản, không thể tính được n trong thời gian đa thức. Vậy khi cho điểm P = nQ, chúng ta có thể biểu diễn hình học để xác định đường thẳng đi qua Q và P, từ đó tìm được điểm (n-1)Q. Nhưng như thế chúng ta mới chỉ biết được tọa độ của (n-1)Q, còn n bằng bao nhiêu thì chúng ta vẫn không biết, và phải đệ quy quá trình này nhiều lần mới xác định được giá trị n. Về cơ bản vẫn là liên tục thử nhiều lần.

Dựa vào thuật toán 4.1 và thuật toán 4.2 ta có độ phức tạp tính toán của thuật toán cải tiến AECC* là O(n*logn).

Bảng 6. Đánh giá thuật toán các hệ mật mã đường cong elliptic

Thuật toán ECC	Dung lượng bộ nhớ (ram)	Thời gian (giây)	Kích thước mã nguồn
CECC	30.47 MB	21	36.50 Byte
AECC	51.35 MB	22	35.98 Byte
AECC*	27.89 MB	20	34.40 Byte
ECC [6]	31.42 MB	35	34.58 Byte
ECC [13]	52.65 MB	60	106.94 Byte

Trong bảng 6 đã so sánh các thuật toán mật mã đường cong elliptic với các hệ mật mã khác. Mật mã AECC* cho kết quả chạy tối ưu nhất so với các thuật toán trên.

7. KẾT LUẬN

Với hệ mật mã AECC* trên đường cong Elliptic mà nhóm nghiên cứu đề xuất cải tiến dựa trên hệ mật mã AECC. Sự cải tiến này không sử dụng thuật toán sinh chuỗi mà thông qua vị trí điểm trên đường cong elliptic. Việc cải tiến này giúp cho bản mã ngắn hơn sẽ tiết kiệm dung lượng và thời gian. Tính bảo mật

của hệ mật mã AECC* phụ thuộc vào độ khó của việc tìm giá trị của khóa k mà khóa k phụ thuộc vào cặp giá trị là u và v, với kq trong đó k là một số lớn ngẫu nhiên và q là một điểm sinh ngẫu nhiên trên đường cong elliptic. Các tham số đường cong elliptic cho các sơ đồ mã hóa nên được lựa chọn cẩn thận để chống lại tất cả các cuộc tấn công đã biết của bài toán logarit rời rạc đường cong elliptic (ECDLP). Do đó, phương pháp mã hóa được đề xuất cải tiến ở đây cung cấp bảo mật đầy đủ chống lại việc phá mã, chi phí tính toán tương đối thấp. Ngoài ra, nó là ánh xạ ký tự dạng 1 → 1 giữa bản rõ và bản mã khi gửi bản mã trên đường truyền sẽ không tốn băng thông so với các thuật toán trước. Thuật toán được cài đặt và thử nghiệm trên ngôn ngữ lập trình C# cho kết quả đúng đắn theo thuật toán đề xuất.

TÀI LIỆU THAM KHẢO

[1] N. Koblitz, "Elliptic curve cryptosystems, Mathematics", 203 – 209, 1987.

[2] V. Miller, *"Uses of Elliptic curves in Cryptography. In advances in Cryptography"* (CRYPTO 1985), Springer LNCS 218,417-4 26, 1985.

[3] Utku Gulen, Selcuk Baktir, *"Elliptic Curve Cryptography for Wireless Sensor Networks Using the Number Theoretic Transform"*, journal-sensors, Published: 9 March, 2020.

[4] Negin Dinarvand, Hamid Barati, "*An efficient and secure RFID authentication protocol using elliptic curve cryptography"*, Springer Science, LLC, 2017.

[5] F. Amounas and E.H. El Kinani, "*ECC Encryption and Decryption with a Data Sequence"*, Applied Mathematical Sciences, Vol. 6, no. 101, 5039 – 5047, 2012.

[6] Mai Mạnh Trừng, Lê Thị Thu Hiền, Trần Minh Đức, *"Đề xuất hệ mật đường cong elliptic với khóa đối xứng",* Tạp chí Khoa học Công nghệ, Trường Đại học Kinh tế - Kỹ thuật Công nghiệp, 2020.

[7] Mai Mạnh Trừng, Đỗ Trung Tuấn, Lê Phê Đô, Lê Trung Thực, Đào Thị Phương Anh, "*Xây dựng hệ mật mã đường cong elliptic với khóa đối xứng affine để mã hóa giải mã văn bản tiếng Việt*", Kỷ yếu Hội nghị KHCN Quốc gia lần thứ XIII về Nghiên cứu cơ bản và ứng dụng công nghệ thông tin (FAIR), 724-732, Nha Trang, ngày 8-9/10/2020.

[8] S. Sandeep, Kumar, *"Elliptic curve cryptography for constrained devices*", PhD thesis, Ruhr-University Bochum, June, 2006.

[9] Trần Duy Lai, *"Mật mã hạng nhẹ*", Viện Khoa học Công nghệ mật mã, Ban Cơ yếu Chính phủ, 2012, https://antoanthongtin.vn/gp-mat-ma/mat-ma-hang-nhe-100502, thời gian truy cập: 19/03/2023.

[10] Trần Văn Trường, Nguyễn Quốc Toàn, *"Mật mã đường cong elliptic*", Viện Khoa học Công nghệ mật mã, Ban Cơ yếu Chính phủ, 2015, https://antoanthongtin.vn/gp-mat-ma/mat-ma-duong-cong-elliptic-va-mat-ma-hang-nhe-101337, thời gian truy cập: 19/03/2023.

[11] Đặng Minh Tuấn, *"Chứng minh tính chất kết hợp của phép cộng trên đường cong elliptic bằng phương pháp đại số*", Học viện Công nghệ Bưu chính Viễn thông, 2020.

[12] D. Sravana Kumar, CH.Suneetha, A.ChandrasekhAR, *"Encryption of data using elliptic curve over finite fields"*, International Journal (IJDPS) Vol.3, No.1, January 2012.

Thông tin liên hệ: **Mai Mạnh Trừng**

Điện thoại: 09123.55.022 - Email: mmtrung@uneti.edu.vn

Khoa Công nghệ thông tin, Trường Đại học Kinh tế - Kỹ thuật Công nghiệp.