BAN CƠ YẾU CHÍNH PHỦ

**HỌC VIỆN KỸ THUẬT MẬT MÃ**



**MÔN HỌC: MẬT MÃ ỨNG DỤNG**

**Đề tài: Tìm hiểu chuẩn mật mã khoá công khai trên đường cong Elliptic**

|  |  |  |
| --- | --- | --- |
|  | Giảng viên: | TS. Hoàng Đức Thọ |
|  | Học viên thực hiện: | Vi Thị Thoa  Khổng Văn Hiếu  Phạm Quang Long |

Hà Nội, 2024

# Mục lục

[Mục lục 2](#_Toc163770251)

[Lời nói đầu 3](#_Toc163770252)

[Chương 1. Tổng quan về hệ mật đường cong Elliptic 4](#_Toc163770253)

[1.1. Giới thiệu hệ mật đường cong Elliptic 4](#_Toc163770254)

[1.2. Các thuật toán liên quan 4](#_Toc163770255)

[1.2.1. Phương trình Weierstrass của đường cong Elliptic 4](#_Toc163770256)

[1.2.2. Đường cong Elliptic trên trường số thực 5](#_Toc163770257)

[1.2.2.1. Phép cộng trên đường cong Elliptic trường số thực 6](#_Toc163770258)

[1.2.2.2. Phép nhân vô hướng các điểm trên đường cong Elliptic trường số thực 10](#_Toc163770259)

[1.2.3. Đường cong Elliptic trên trường hữu hạn 11](#_Toc163770260)

[1.2.3.1. Đường cong Elliptic trên trường Zp 11](#_Toc163770261)

[1.2.3.2. Đường cong Elliptic trên trường GF(2m) 14](#_Toc163770262)

[1.3. Cơ chế hoạt động 15](#_Toc163770263)

[1.3.1. Phương pháp Elgamal 16](#_Toc163770264)

[1.3.2. Phương pháp Menezes – Vanstone: 16](#_Toc163770265)

[Chương 2. Giao thức trao đổi khoá Diffie Hellman trên đường cong Elliptic 18](#_Toc163770266)

[2.1. Tổng quan về giao thức trao đổi khoá 18](#_Toc163770267)

[2.1.1. Khái niệm 18](#_Toc163770268)

[2.1.2. Một số giao thức trao đổi khoá phổ biến 18](#_Toc163770269)

[2.2. Giới thiệu giao thức trao đổi khoá Diffie Hellman trên đường cong Elliptic 19](#_Toc163770270)

[2.2.1. Ứng dụng của giao thức trao đổi khoá ECDH 19](#_Toc163770271)

[2.2.2. Hoạt động của giao thức trao đổi khoá Diffie Hellman trên đường cong Elliptic 20](#_Toc163770272)

[Chương 3. Ứng dụng sử dụng giao thức trao đổi khoá trên đường cong Elliptic 21](#_Toc163770273)

[3.1. Mô hình thực nghiệm 21](#_Toc163770274)

[3.2. Hoạt động của ứng dụng 21](#_Toc163770275)

[3.2.1. Giai đoạn xác thực 22](#_Toc163770276)

[3.2.2. Giai đoạn trao đổi khoá 22](#_Toc163770277)

[3.2.3. Giai đoạn mã hoá, giải mã 23](#_Toc163770278)

[Kết luận 24](#_Toc163770279)

[Tài liệu tham khảo 25](#_Toc163770280)

[Phụ lục 26](#_Toc163770281)

# Lời nói đầu

Trong thế kỷ 21 hiện nay, việc bảo vệ thông tin và dữ liệu cá nhân trở thành một trong những ưu tiên hàng đầu của xã hội thông tin. Đồng thời, sự phát triển vượt bậc của công nghệ đòi hỏi sự sáng tạo và tiên tiến trong lĩnh vực bảo mật. Trong bối cảnh đó, mật mã trên đường cong Elliptic (ECC) nổi lên như một công nghệ bảo mật tiên tiến và mạnh mẽ, được áp dụng rộng rãi trong nhiều lĩnh vực, từ thanh toán điện tử đến IoT (Internet of Things).

Đường cong Elliptic không chỉ là một công cụ hiệu quả trong việc mã hóa dữ liệu mà còn mang lại những ưu điểm vượt trội so với các phương pháp truyền thống khác như RSA. Điều này bao gồm kích thước khóa nhỏ hơn, hiệu suất tính toán cao hơn và khả năng chống lại các cuộc tấn công phân tích theo dạng quantum.

Nghiên cứu này sẽ tìm hiểu sâu về mật mã trên đường cong Elliptic, từ nguyên lý cơ bản đến các ứng dụng thực tiễn. Nghiên cứu cũng tìm hiểu cấu trúc và tính toán trên đường cong Elliptic, đồng thời xem xét về bảo mật và hiệu suất của nó so với các phương pháp khác.

Với sự phát triển không ngừng của công nghệ và sự tăng cường về nhu cầu bảo mật, việc hiểu và áp dụng mật mã trên đường cong Elliptic trở thành một phần không thể thiếu trong hệ thống bảo mật thông tin hiện đại. Bằng cách khám phá và nghiên cứu sâu sắc về chủ đề này, chúng ta có thể nắm bắt được những tiềm năng và ứng dụng tuyệt vời mà ECC mang lại, đồng thời đóng góp vào việc tạo ra một môi trường truyền thông an toàn và bảo mật hơn cho thế giới số ngày nay.

# Tổng quan về hệ mật đường cong Elliptic

## Giới thiệu hệ mật đường cong Elliptic

Hệ mật trên đường cong Elliptic (Elliptic Curve Cryptography - ECC) là một trong những phương pháp mã hóa đối xứng tiên tiến và được sử dụng rộng rãi trong bảo mật thông tin ngày nay. Nó dựa trên tính toán các điểm trên đường cong elliptic trong một không gian affine hoặc projective để thực hiện các phép toán mã hóa và giải mã.

Đặc điểm chính của ECC bao gồm:

* Độ dài khóa ngắn: ECC cung cấp mức độ bảo mật tương đương với RSA nhưng với các khóa ngắn hơn nhiều. Điều này làm giảm tải cho việc truyền thông mạng và tăng hiệu suất tính toán.
* Hiệu suất cao: ECC yêu cầu ít tài nguyên tính toán hơn so với các phương pháp mã hóa truyền thống khác như RSA, đặc biệt là trong các thiết bị có tài nguyên hạn chế như điện thoại di động hoặc thiết bị IoT.
* Bảo mật cao: ECC đang được sử dụng rộng rãi trong các ứng dụng an ninh như thanh toán điện tử, truyền thông không dây và các ứng dụng mạng khác vì tính bảo mật của nó.
* Quy trình hoạt động cơ bản của ECC bao gồm tạo ra một cặp khóa công khai và khóa bí mật từ đường cong elliptic. Bằng cách sử dụng khóa này, thông điệp có thể được mã hóa bằng khóa công khai và chỉ có thể được giải mã bằng khóa bí mật tương ứng. Điều này đảm bảo tính toàn vẹn và bảo mật của dữ liệu được truyền.

Tính toán trên đường cong elliptic liên quan đến các phép cộng, nhân và định lý của nhóm đường cong elliptic. Đối với một số lựa chọn thích hợp của các tham số, tính toán các phép toán này trở nên khó khăn đối với kẻ tấn công mà không cần sử dụng các khóa rất lớn như trong các hệ thống khác như RSA

## Các thuật toán liên quan

### Phương trình Weierstrass của đường cong Elliptic

Với *K* là một trường có *a,b K*, khi đó ta có thể nói đường cong Elliptic được định nghĩa trên trường *K*. Với mọi điểm x,y trên đường cong Elliptic và *(x,y) K* hữu tỷ,dạng tổng quát phương trình Weierstrass của đường cong Elliptic được biểu diễn là:

*y2 + xy + y = x3 + x2 + x + (1.1)*

Trong đó a1, a2, …, a6 là các hằng số. Dạng tổng quát trên thường được sử dụng trong các trường hợp K có *chap(K)* bằng 2 hoặc 3. Trường hợp *chap(K)* khác 2, ta có thể biến đổi như sau:

,

Ta có thể viết lại như sau: =

Với  và với các hằng số . Khi *K* có *chap(K)* khác 3, ta có thể dùng phép thế  từ đó ta có:

Trong đó *A,B* là các hằng số. Từ đó ta có phương trình tổng quát của đường cong Elliptic có dạng:

***y2 = x3 + Ax + B*** *(1.2)*

Với *A,B* là các hằng số và giá trị của *x, y, A, B* thường là các giá trị trên cùng một trường nào đó, ví dụ như **R** (số thực), **Q** (số hữu tỷ), **C** (số phức), hoặc trường hữu hạn **Fq**,với *q = pn*trong đó *p* là số nguyên tố với *n 1*. Đường cong *y2 = x3 + Ax + B* có định thức = -16(4A3 + 27B2). Khi thì đường cong này suy biến và không có đủ 3 nghiệm phân biệt, hay đường cong sẽ bị gãy hoặc gấp khúc tại một điểm nào đó. Do đó, ta chỉ xét tới trường hợp ≠ 0, nghĩa là -16(4A3 + 27B2) ≠ 0 hay (4A3 + 27B2) ≠ 0.

### Đường cong Elliptic trên trường số thực

Xét đường cong Elliptic có dạng ***y2 = x3 + ax + b*** trên trường số thực ***R***, với *x, y, a, b* ***R***. Với điều kiện *(4a3 + 27b2) ≠ 0*.

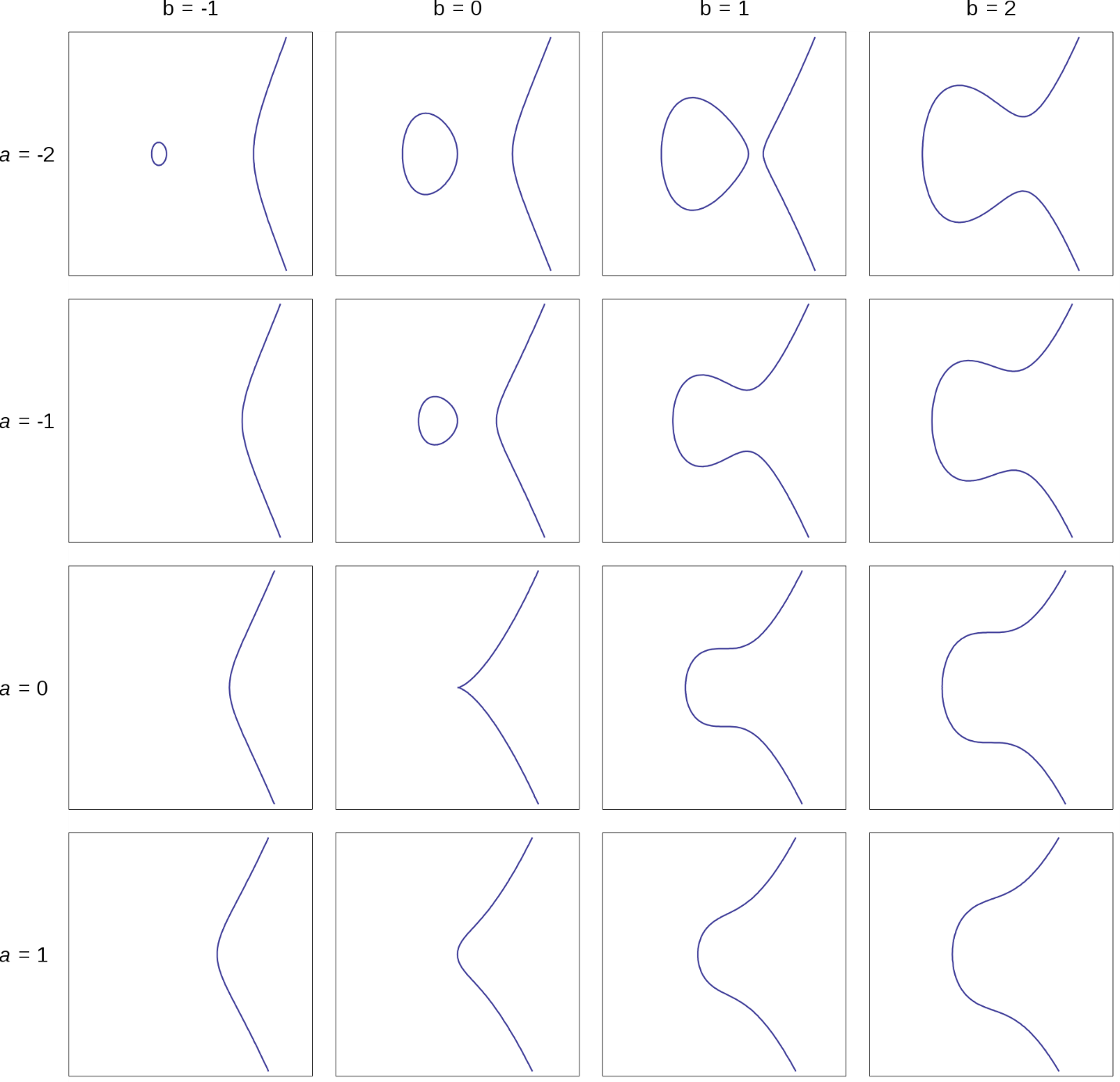
Đồ thị của đường cong sẽ được biểu diễn như sau:

A graph of a function

Description automatically generated

*Hình 1. Đồ thị của đường cong Elliptic*

Hình dạng của đường cong phụ thuộc vào các giá trị *a, b*, sau đây là một vài hình dạng cơ bản của đường cong Elliptic trên trường số thực:



*Hình 2: Một số hình dạng của đường cong Elliptic*

Ta có thể thấy, *khi a = 0* và *b = 0* (tức *(4a3 + 27b2 = 0)* thì đường cong bị gãy khúc tại 1 điểm, khi đó đường cong không có đủ 3 nghiệm phân biệt.

#### Phép cộng trên đường cong Elliptic trường số thực

Đường cong Elliptic có một điểm *O* gọi là điểm vô cực. Gọi *E(a, b)* là tập các điểm thuộc đường cong *y2= x3 + ax + b* cùng với điểm *O*. Ta định nghĩa phép cộng trên tập các điểm thuộc *E(a, b)* như sau:

1. Điểm *O* là phần tử đơn vị của phép cộng. Như vậy, với *P E(a, b) và P 0* thì *P + O = O + P=P*.

2. Phần tử nghịch đảo của điểm *P* trong phép cộng, ký hiệu *– P*, là điểm đối xứng với *P* qua trục hoành.

3. Với 2 điểm *P, Q* bất kỳ, ta giả định *P 0 và Q 0,* kẻ một đường thẳng đi qua *P* và *Q* thì sẽ cắt đường cong Elliptic tại một điểm thứ 3 là điểm *S*. Phép cộng của *P* và *Q* sẽ là: *R= P + Q = -S*.

A diagram of a mathematical equation

Description automatically generated

*Hình 3: Phép cộng hai phần tử P và Q của đường cong Elliptic trên trường R*

Trong trường hợp *P* và *Q* đối xứng qua trục hoành, hay nói cách khác *Q = - P* theo định nghĩa 2 thì đường thẳng nối *P, Q* sẽ cắt đường cong Elliptic tại vô cực, hay *P + (- P ) = 0.*

4. Để tính *P + P*, ta vẽ đường thẳng tiếp tuyến với đường cong Elliptic tại *P*, đường thẳng này cắt đường cong tại điểm S, lúc đó: *R = P + P =* .

A graph of a function

Description automatically generated

*Hình 4: Phép cộng P và P trên đường cong Elliptic*

Tập *E(a, b)* cùng với phép cộng định nghĩa như trên tạo thành một nhóm Abel.

Tính giá trị của phép cộng:

Gọi tọa độ của điểm *P* là (, tọa độ của điểm *Q* là . Ta tính tọa độ của điểm *R = P + Q = -S* như sau:

*R ( = P ( + Q*

Trong đó, đường thẳng nối *P* với *Q* cắt đường cong Elliptic tại điểm *S ()* và *R = -S.* Đường thẳng đi qua 2 điểm P và Q có phương trình là:

*y = λx + µ (1.3)*

Đối với *λ* là hệ số góc của đường thẳng đi qua *P* và *Q*, ta có:

*= λ + µ (1.4)*

*= λ + µ (1.5)*

*= λ + µ (1.6)*

**Trường hợp *P ≠ Q:***

Từ *(1.4)* và *(1.5)* ta có thể suy ra = *λ( ).* Khi *P ≠ Q,* nghĩa là *≠ ,* ta có công thức sau:

***λ =***  *(1.7)*

*µ =*

*(1.8)*

Thay *y* ở *(1.3)* vào phương trình *(1.1),* ta có:

*y2 + xy + y = x3 + x2 + x +*

*⬄ x3 + x2 + x + (1.9)*

Từ đó dẫn đến phương trình *r(x) = 0* với:

*r(x) = x3 + ()* *x2 + ()x + ) (1.10)*

Do *r(x)* có 3 nghiệm phân biệt nên ta có thể viết:

*r (x) = (x – )( x – )( x – )*

*= [**x2 – ()x + ](x – )*

*= x3 ()* *x2+ ()*

*= x3* *() x2  (*+ *)* *(1.11)*

Đồng nhất hệ số *x2* của *r(x)* ở hai phương trình *(1.10)* và *(1.11)* ta có:

*()*

Từ đó ta có thể tính được theo công thức sau:

*(1.12)*

Tại đây, ta tính tiếp giá trị , do đã tính xong nên ta có thể coi là hằng số, thay vào (1.2) ta có thể viết lại như sau:

*(1.13)*

Phương trình bậc 2 này có 2 nghiệm là:

Cộng hai điểm này lại ta sẽ có , mặt khác do nằm trên đường thẳng *PQ* nên *λ +* *µ.* Từ đây ta có thể tính được như sau:

*λ µ (1.14)*

Thay *µ* từ *(1.8)* ta có thể tính như sau:

*λ()* *(1.15)*

**Trường hợp P và Q trùng nhau:**

Khi P và Q trùng nhau thì và do đó, công thức tính *λ* ở *(1.7)* không thể sử dụng được do Trong trường hợp này *λ* chính là hệ số góc của đường thẳng tiếp tuyến với đường cong tại *P* hay *Q.* Hệ số góc tại tiếp tuyến của đường cong chính là đạo hàm , sử dụng các quy tắc lấy đạo hàm của tích, đạo hàm của hàm số hợp và lấy đạo hàm 2 vế của phương trình *(1.1)*, ta có:

*⬄*

*⬄*

*⬄ 2y*

*⬄ 2y + +   
⬄*

Như vậy với điểm *P ()* ta có:

*λ = (1.16)*

Trong tất cả các trường hợp thì điểm *R* là tổng của hai điểm *P* và *Q* có tọa độ là:

*R ( = () (1.17)*

Với đường cong Elliptic dạng ***y2 = x3 + ax + b****,* khi đó Điểm *R* có tọa độ sẽ được tính theo công thức:

*R ( = () (1.18)*

Trường hợp *P = Q (P trùng với Q)* thì *(1.19)*

#### Phép nhân vô hướng các điểm trên đường cong Elliptic trường số thực

Với *n ∈ N \ {0}* định nghĩa phép nhân vô hướng của điểm *P* nằm trên đường

cong Elliptic là phép cộng *n* lần chính bản thân điểm *P*:

*P → nP =*

*n lần*

Ví dụ: Trong phép toán tính *3P*, đầu tiên ta sẽ tính *2P* bằng cách tính *P+P*.

Theo cách cộng ở bên trên, ta vẽ đường thẳng nối *P* và *P*, ở đây chính là tiếp tuyến của đường cong, nó cắt đường cong tại điểm -*2P*, lấy đối xứng qua trục hoành ta có *2P*.

Tiếp tục vẽ đường thẳng nối giữa *2P* và *P*, cắt đường cong tại -*3P*, lấy đối xứng ta có *3P*.

Hình dưới mô tả phép nhân được tiến hành trong đường cong Elliptic thế nào:

A diagram of a graph

Description automatically generated

*Hình 5: Phép nhân 3P trên đường cong Elliptic*

Do cách tính toán trên, ta có thể dễ dàng tính toán được phép nhân k\*P khi biết k và P, nhưng hoàn toàn không thể tính toán được theo chiều ngược lại, tức phép chia. Đó cũng chính là tính chất đặc trưng thú vị của mã hoá bất đối xứng sử dụng đường cong Elliptic.

Để tối ưu phép nhân vô hướng, có thể sử dụng phương pháp *Nhân đôi - và - cộng*, đầu tiên biểu diễn số *n* dưới dạng: n = với [], sau đó sử dụng thuật toán sau:

Thuật toán *Nhân đôi - và - cộng:*

1: *Q*

2: **for** *i = 0 to m* **do**

3: **if**  **then**

4:  *Q CộngĐiểm(Q,P)*

5: **then if**

6:  *P ← NhânĐôi(P)*

7: **end for**

8: **return** *Q*

Ngoài phương pháp *Nhân đôi – và - cộng*, có thể sử dụng phương pháp *Trượt-cửa sổ*. Các phương pháp này cho phép nhân vô hướng một cách tối ưu. Lưu ý, đường cong Elliptic không tồn tại:

* Phép nhân 2 điểm *P \* Q* trên đường cong Elliptic
* Phép chia vô hướng *Q : n (với Q = nP)*

Việc tìm số *n* là bài toán Logarit rời rạc - một bài toán khó có thể giải được trong thời gian đa thức

### Đường cong Elliptic trên trường hữu hạn

Đường cong Elliptic được xây dựng trên các trường hữu hạn. Có hai trường hữu hạn thường được sử dụng đó là trường hữu hạn F\_q với q là số nguyên tố hoặc q là 2m (với m là số nguyên).

#### Đường cong Elliptic trên trường Zp

Đối với trường hữu hạn *q* là số nguyên tố, ta xét đường cong trên trường hữu hạn này, sau đây gọi là trường . Đường cong Elliptic trên trường là đường cong có các hệ số thuộc trường , đường cong này có dạng:

*mod p = (* *+ ax + b) mod p*

Ví dụ trên trường  chọn a = 1, b = 1, x = 9, y = 7, ta có:

*23 = () mod 23*

*hay 49 mod 23 = 739 mod 23 = 3*

Khác với đường cong Elliptic trên trường số thực, chúng ta không thể biểu diễn đường cong Elliptic  bằng đồ thị hàm số liên tục. Cũng tương tự như khái niệm đối xứng qua trục hoành của đường cong Elliptic số thực, đường cong Elliptic cũng đối xứng theo nghĩa đối xứng modulo. Giả sử điểm *(x, y)* thuộc đường cong Elliptic trên thì điểm *(x, p - y)* cũng thuộc đường cong trên do: . Ví dụ: điểm *(1,7)* đối xứng với điểm *(1,16)* vì *7 + 16 = 0 mod 23.*

Hình dưới thể hiện các điểm thuộc đường cong trong trường với *a = 1* và *b = 1.* Các điểm đối xứng với nhau qua đường *y = 11.5*. Riêng điểm *(4, 0)* xem như là đối xứng với chính nó.

A grid of lines with red dots

Description automatically generated

*Hình 6: Minh họa các điểm thuộc đường cong Elliptic*

Tương tự như nhóm Abel *E(a,b)* định nghĩa trên đường cong Elliptic số thực, chúng ta cũng định nghĩa một nhóm Abel *(a,b)* gồm các điểm của đường cong Elliptic cùng với điểm vô cực *O*.

1. Điểm *O* là phần tử đơn vị của phép cộng. Như vậy, với *P (a,b)* *và P 0* thì *P + O = O + P = P*

1. Phần tử nghịch đảo của điểm *P* trong phép cộng, ký hiệu *-P* là điểm đối xứng với *P*. Như vậy *P + (-P) = O.*
2. Với 2 điểm *P, Q* bất kỳ, phép cộng *R = P + Q = -S* được xác định bằng công thức:

Giả sử *P* có tọa độ *(),* *Q* có tọa độ *()*, điểm R có tọa độ *()*, khi đó:

*mod p (1.20)*

*mod p (1.21)*

Trong đó:

Sau đây ta sẽ chứng minh công thức này:

Trong khái niệm đường cong Elliptic trên trường không có khái niệm đường thẳng. Do đó, ta cần chứng minh tính đóng, tức là *R ()* là điểm thuộc đường cong, điểm *S* () hay *S ()* là điểm đối xứng với R trên đường cong Elliptic*.* Xét trường hợp *P ≠ Q,*  ta có:

*(1.22)*

*(1.23)*

*(1.24)*

Ta cần chứng minh *()* thuộc đường cong, nghĩa là:

⬄

⬄ *(1.25)*

Vì *p* là số nguyên tố nên tồn tại phần tử nghịch đảo của *()* trong phép modulo *p.* Do đó phương trình *(1.25)* tương đương với:

*a mod p (1.26)*

Để chứng minh *(1.26),* lấy *(1.23)* trừ cho *(1.22),* ta được:

⬄  *a mod p*

⬄ 2 *mod p (1.27)*

Từ *(1.20)* ta có:

*mod p*

⬄ *mod p*

⬄ *mod p (1.28)*

Thay *(1.27)* vào phương trình *(1.28),* ta có:

2*mod p*

⬄ + 2 *mod p*

Vậy *(1.26)* đúng, nghĩa là điểm *S* () thuộc đường cong, mà *S* điểm đối xứng của *R* trên đường cong nên *R ()* cũng thuộc đường cong.

Trường hợp *R = P+P* ta chứng minh tương tự.

Ví dụ: Trong *(1, 1),* chọn *P = (3, 10)* và *Q = (9,7)*, từ đó:

*=> = 17*

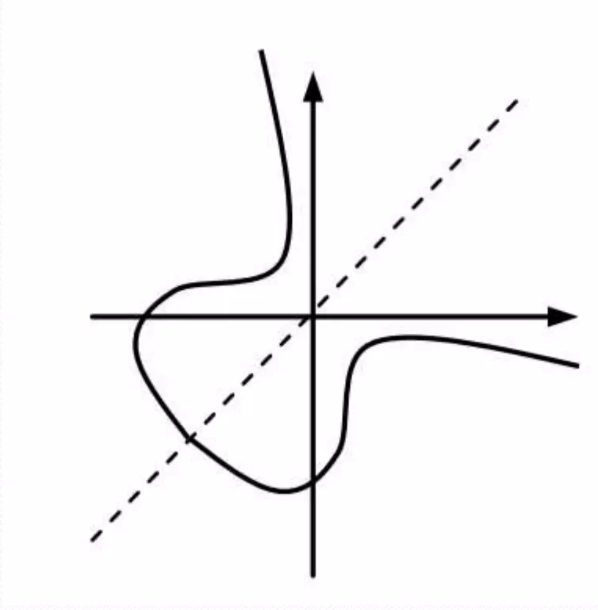
*=> 11(3-17) -10 mod 23 =20*

Do đó điểm *R* *(17, 20)* thuộc đường cong *(1, 1)*

#### Đường cong Elliptic trên trường GF(2m)

Đường cong Elliptic trên trường *GF(2m)* là đường cong có các hệ số thuộc trường *GF(2m),* đường cong này có dạng hơi khác so với đường cong trên :

*,với x, y, a, b GF(2m) (1.29)*

**

*Hình 7: Đường cong trên trường số thực*

Ta xét tập gồm các điểm thuộc đường cong Elliptic này cùng với điểm vô cực *O.*

Ví dụ: Xét trường *GF(2m)* với đa thức tối giản là *m(x) =* *x4 + x + 1.* Phần tử sinh *g* của trường này có điều kiện *g4 = g + 1.* Ta có bảng các lũy thừa của *g* là:

|  |  |  |
| --- | --- | --- |
| ***Biểu diễn lũy thừa*** | ***Đa thức trong GF(23)*** | ***Số nhị phân*** |
| *0* | *0* | *0000* |
| *g0* | *1* | *0001* |
| *g1* | *g* | *0010* |
| *g2* | *g2* | *0100* |
| *g3* | *g3* | *1000* |
| *g4* | *g + 1* | *1011* |
| *g5* | *g2 + g* | *0110* |
| *g6* | *g3 + g2* | *1100* |
| *g7* | *g3 + g + 1* | *1011* |
| *g8* | *g2 + 1* | *0101* |
| *g9* | *g3 + g* | *1010* |
| *g10* | *g2 + g + 1* | *0111* |
| *g11* | *g3 + g2 + g* | *1110* |
| *g12* | *g3 + g2 + g + 1* | *1111* |
| *g13* | *g3 + g2 + 1* | *1101* |
| *g14* | *g3 + 1* | *1001* |

Xét ví dụ về đường cong Elliptic trên trường *GF(24):*

*y2 + xy = x3 + gx + 1 , (a = g4, b = 1)*

Bảng liệt kê bên dưới liệt kê các điểm thuộc đường cong này:

|  |  |  |
| --- | --- | --- |
| *(0, 1)* | *(g5, g3)* | *(g9, g13)* |
| *(1, g6)* | *(g5, g11)* | *(g10, g)* |
| *(1, g13)* | *(g6, g8)* | *(g10, g8)* |
| *(g3, g8)* | *(g6, g14)* | *(g12, 0)* |
| *(g3, g13)* | *(g9, g10)* | *(g12, g12)* |

Tương tự như nhóm Abel *(a, b),* ta cũng xây dựng một nhóm Abel gồm các điểm thuộc đường cong Elliptic trên trường *GF(2m)* cùng với điểm vô cực *O.* Ta có:

1. Điểm *O* là phần tử đơn vị của phép cộng, Như vậy, với *P* là một điểm thuộc đường cong và *P ≠ 0* thì: *P + O = O + P = P*.
2. Phần tử nghịch đảo của điểm *P* trong phép cộng, ký hiệu *-P* là điểm đối xứng với *P.* Ta có: giả sử *P ()* thì *-P ().*
3. Với hai điểm *P, Q* bất kỳ *(P ≠ Q)* thì phép cộng *R = P + Q* được xác định bằng công thức sau, giả sử điểm *Q* có tọa độ *Q ()* và *R* có tọa độ *R ()*:

*( )*

*Trong đó:*

## Cơ chế hoạt động

Trong vấn đề mã hóa/giải mã trên đường cong Elliptic, ta chọn các tham số để tạo một nhóm Abel *Eq(a,b)* và chọn một điểm cơ sở G có hạng *n* lớn.

Các thành phần khóa khóa riêng và công khai trong mã hóa EC được định nghĩa như sau:

*KR = (d, G, q, a, b)*

*KU = (E, G, q, a, b*)

Trong đó *d < n* và *E = dG* với *d* là một số bí mật do người sinh khóa chọn. Do tính chất của hàm một chiều từ *E* và *G*, không thể suy ra được *d*.

Từ đó chúng ta có hai cách thức thực hiện mã hóa, giải mã như sau:

### Phương pháp Elgamal

* Giả sử Alice muốn gửi một thông điệp *M* cho Bob, trước tiên Alice chuyển *M* từ dạng dãy bít sang dạng điểm *PM =(x, y).* Bản mã *CM* (dùng khóa công khai của Bob) được tính là một cặp điểm như sau:

*CM = {kG, PM + kE}*với *k* là một số ngẫu nhiên do Alice chọn

* Để giải mã dùng khóa riêng, Bob sẽ nhân điểm thứ nhất trong *CM* với *d*, sau đó lấy điểm thứ hai trừ cho kết quả:

*PM + kE – dkG = PM + kdG – kdG = PM*

* Trong phương thức mã hóa, Alice đã che giấu PM bằng cách cộng *PM* với *kE*. Để giải mã, Bob cần trừ ra lại *kE*. Thay vì gửi trực tiếp *k* cho Bob để Bob tính *kE* (Trudy có thể chặn được), Alice gửi một dấu hiệu là *kG*. Dựa vào *kG* và *d*, Bob có thể tính *kE*. Còn Trudy, dù biết *G* và *kG*, tuy nhiên vẫn không thể tính được *k* do tính chất của hàm một chiều.
* Ví dụ: chọn *p = 751, a = 1, b = 188* ta có đường cong Elliptic trên Z751 như sau:

y2 mod 751 = (x3 + x + 188) mod 751 trong đó a,b,x,y ∈ Z751

Chọn điểm cơ sở là G = (0,376)

Giả sử Alice cần mã hóa bản rõ là điểm PM = (562, 201) dùng khóa công khai E = (201, 5). Alice chọn k = 386. Ta có

386(0, 36)= (676, 58)

(562,201)+368(201,5)=(385,328)

Vậy bản mã là cặp điểm {(676,58), (385,328)}

### Phương pháp Menezes – Vanstone:

* Thông điệp *M* của Alice được tách thành hai phần *M=(m1, m2)* sao cho *m1, m2 ∈ Zp*. Alice chọn một số ngẫu nhiên *k*, kết hợp với khóa công khai của Bob, Alice tính điểm P như sau:

*P(xp,yp) = kE*

* Bản mã CM gồm 3 thành phần:

CM ={ c0, c1, c2 } ={kG, xpm1 mod p, ypm2 mod p}

* Để giải mã dùng khóa riêng, từ dấu hiệu kG, Bod tính:

P(xp, yp) = kdG

* Và từ đó tính nghịch đảo của xp-1 và yp-1 trong phép modulo p. Cuối cùng bản giải mã là:

M ={m1, m2} = { xp-1 c1 mod p, yp-1 c2 mod p}

Tương tự như phương pháp Elgamal, dù biết G và kG, Trudy cũng không thể tính được k để tính P.

# Giao thức trao đổi khoá Diffie Hellman trên đường cong Elliptic

## Tổng quan về giao thức trao đổi khoá

### Khái niệm

Giao thức trao đổi khóa là một phương pháp mà các bên trong một hệ thống bảo mật sử dụng để tạo ra, trao đổi và thống nhất trên một khóa bí mật giữa họ. Mục tiêu của giao thức này là đảm bảo rằng hai hoặc nhiều bên có thể thiết lập một khóa chung mà chỉ có họ biết, và các bên khác không thể tiếp cận hoặc hiểu được nó.

Một số giao thức trao đổi khóa phổ biến bao gồm Diffie-Hellman, RSA, và các biến thể của chúng như ECDH (Elliptic Curve Diffie-Hellman) hoặc DHKE (Diffie-Hellman Key Exchange). Các giao thức này thường được sử dụng trong các ứng dụng mạng và bảo mật thông tin để đảm bảo tính toàn vẹn và bảo mật của dữ liệu được truyền qua mạng.

### Một số giao thức trao đổi khoá phổ biến

Hiện nay có nhiều giao thức trao đổi khóa được sử dụng trong các hệ mật. Một số giao thức trao đổi khóa tiêu biểu như:

* Diffie-Hellman (DH): Giao thức Diffie-Hellman cho phép hai bên (ví dụ: máy khách và máy chủ) thiết lập một khóa chia sẻ mà không cần truyền dữ liệu trực tiếp thông qua kênh không an toàn. Cả hai bên sử dụng số nguyên tố và phép toán modulo để tạo ra khóa chia sẻ.
* RSA (Rivest-Shamir-Adleman): Giao thức RSA kết hợp hai quá trình quan trọng: mã hóa và ký số. Nó sử dụng một cặp khóa công khai và khóa bí mật để trao đổi thông tin mật và xác thực. RSA thường được sử dụng để trao đổi khóa đối xứng cho các giao thức như TLS.
* Elliptic Curve Diffie-Hellman (ECDH): ECDH sử dụng các điểm trên đường cong elip để tạo ra khóa chia sẻ giữa hai bên mà không cần truyền số nguyên tố hay các phần tử khác. Nó được biết đến vì đòi hỏi ít tài nguyên tính toán hơn so với Diffie-Hellman cổ điển.
* Secure Remote Password (SRP): SRP là một giao thức trao đổi khóa được thiết kế đặc biệt cho việc xác thực mật khẩu của người dùng mà không cần truyền mật khẩu đó qua mạng. Thay vào đó, các bên tạo ra một khóa chia sẻ từ mật khẩu và sử dụng nó để xác thực lẫn nhau.
* Secure Remote Password (SRP): SRP là một giao thức trao đổi khóa được thiết kế đặc biệt cho việc xác thực mật khẩu của người dùng mà không cần truyền mật khẩu đó qua mạng. Thay vào đó, các bên tạo ra một khóa chia sẻ từ mật khẩu và sử dụng nó để xác thực lẫn nhau.

Các giao thức này thường được sử dụng trong các ứng dụng bảo mật mạng và truyền thông để đảm bảo tính bảo mật của dữ liệu truyền qua kênh không an toàn.

## Giới thiệu giao thức trao đổi khoá Diffie Hellman trên đường cong Elliptic

### Ứng dụng của giao thức trao đổi khoá ECDH

Giao thức trao đổi khóa Elliptic Curve Diffie-Hellman (ECDH) có nhiều ứng dụng trong lĩnh vực bảo mật mạng và truyền thông như:

* Mã hóa và Giải mã Dữ liệu: ECDH có thể được sử dụng để mã hóa và giải mã dữ liệu giữa hai bên một cách an toàn. Các khóa chia sẻ được tạo ra từ ECDH được sử dụng để mã hóa thông điệp, đảm bảo rằng dữ liệu không thể bị đọc được nếu không có khóa chính xác.
* Xác thực và Chứng thực: ECDH cũng có thể được sử dụng để xác thực và chứng thực các bên trong một hệ thống. Bằng cách sử dụng khóa chia sẻ được tạo ra từ quá trình trao đổi khóa, các bên có thể xác nhận danh tính của nhau một cách an toàn.
* Tạo chữ ký số: ECDH có thể được sử dụng để tạo chữ ký số cho các thông điệp hoặc tệp tin. Bằng cách sử dụng khóa chia sẻ được tạo ra từ ECDH, một bên có thể tạo ra một chữ ký số dựa trên dữ liệu và khóa của họ mà chỉ có thể được xác nhận bởi bên khác bằng cách sử dụng khóa công khai tương ứng.
* Bảo mật Giao dịch Thẻ Tín Dụng (Credit Card): ECDH được sử dụng rộng rãi trong các giao thức bảo mật như TLS để bảo vệ giao dịch thẻ tín dụng trực tuyến. Các khóa ECDH được sử dụng để thiết lập kết nối an toàn giữa máy khách và máy chủ, đảm bảo rằng thông tin thẻ tín dụng không bị đánh cắp trong quá trình truyền dữ liệu.
* Ứng dụng IoT (Internet of Things): Trong các ứng dụng IoT, ECDH có thể được sử dụng để đảm bảo tính toàn vẹn và bảo mật của dữ liệu truyền qua mạng giữa các thiết bị kết nối. Các khóa chia sẻ được tạo ra từ ECDH có thể được sử dụng để mã hóa và giải mã dữ liệu IoT một cách an toàn.

### Hoạt động của giao thức trao đổi khoá Diffie Hellman trên đường cong Elliptic

Năm 1998, Laurie và cộng sự đề xuất giao thức trao đổi khóa dựa trên ECC. Sau đó giao thức này đã được đưa vào các tiêu chuẩn ANSI X9.42, ANSI X9.63 và IEEE P1363.

Trước tiên ta chọn một số nguyên *q* lớn, với *q* là số nguyên tố (nếu sử dụng đường cong Elliptic ) hoặc *q* có dạng *2m* (nếu chọn đường cong *GF(2m)*), và chọn 2 tham số *a, b* tương ứng để tạo thành nhóm *(a,b).* Ta gọi *G* là điểm cơ sở của nhóm nếu tồn tại một số nguyên n sao cho *nG = 0*. Số nguyên *n* nhỏ nhất như vậy được gọi là hạng của *G*.

Trong trao đổi khóa EC Diffie-Hellman, hai bên *A* (Alice) và *B* (Bob) cần tạo khóa phiên bí mật trao đổi trong một kênh truyền công khai, hai bên cùng thỏa thuận điểm cơ sở *P* trên *E* (đường cong Elliptic), ta chọn một điểm G có hạng *n* lớn, và giao thức trao đổi khóa giữa Alice và Bob tiến hành như sau:

1) Alice chọn một số *nA < n* và giữ bí mật số *nA* này. Sau đó trong (a,b) Alice tính *PA= nAG* và gửi cho Bob.

2) Tương tự Bob chọn một số bí mật *nB*, tính *PB* và gửi PB cho Alice.

3) Alice tạo khóa phiên bí mật là *K= nA PB= nA nBG*

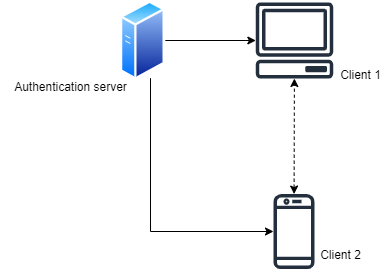
4) Bob tạo khóa phiên bí mật là *K= nB PA= nA nBG* (nhóm Abel có tính giao hoán) giống với khóa của Alice.

Trudy có thể chặn được *PA* và *PB*, tuy nhiên chỉ có thể tính được điều này là bất khả thi như ta đã thấy ở phần trên.

Chú ý: khóa phiên *K* là một điểm trong đường cong Elliptic, để sử dụng khóa này cho mã hóa đối xứng như *DES* hay *AES*, ta cần chuyển *K* về dạng số thường.

# Ứng dụng sử dụng giao thức trao đổi khoá trên đường cong Elliptic

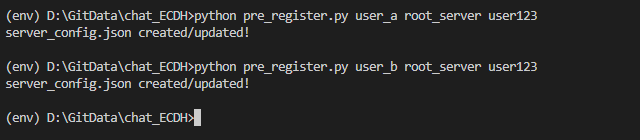
## Mô hình thực nghiệm

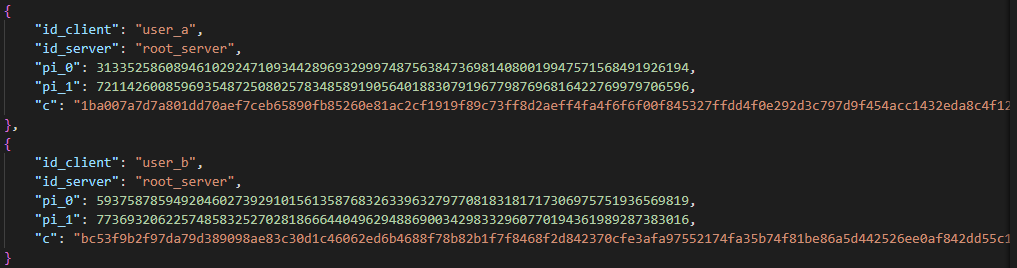


Đề tài thử nghiệm xây dựng ứng dụng cho phép các thiết bị client có thể trao đổi khoá và truyền thông trên môi trường không an toàn.

## Hoạt động của ứng dụng

Đầu tiên, client cần được thêm thông tin định danh vào cơ sở dữ liệu để phục vụ cho quá trình xác thực và kết nối với các client khác. Trong khuôn khổ đề tài này, client cần xác thực với server trước khi kết nối peer-to-peer để trao đổi thông tin.



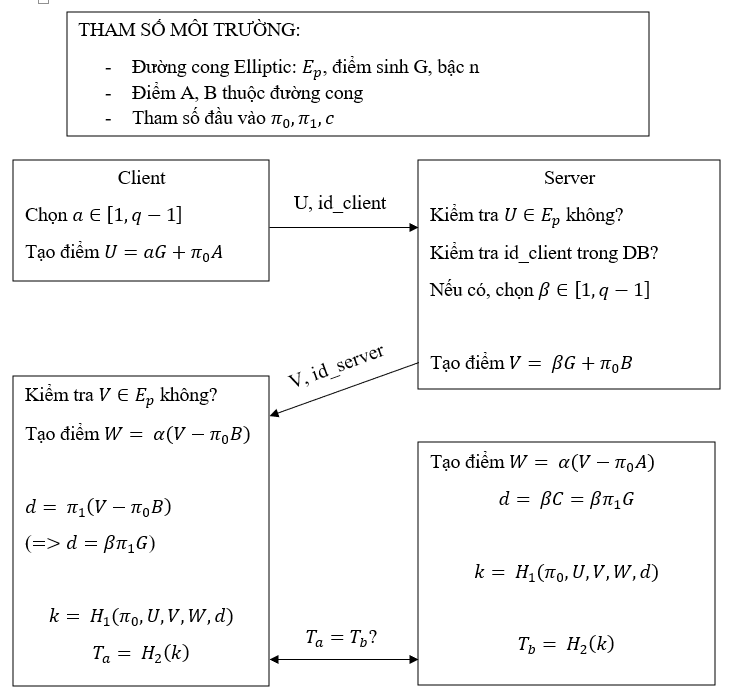


Hình .. Khởi tạo thông tin ban đầu

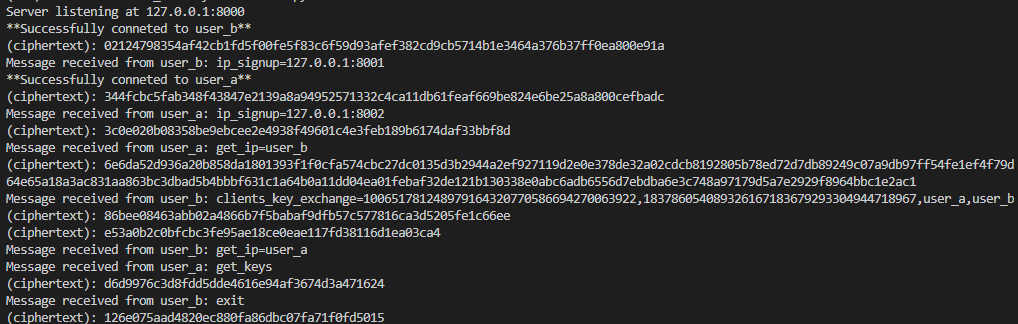
Quy trình thực hiện xác thực gồm hai giai đoạn: xác thực và trao đổi khoá.

### Giai đoạn xác thực

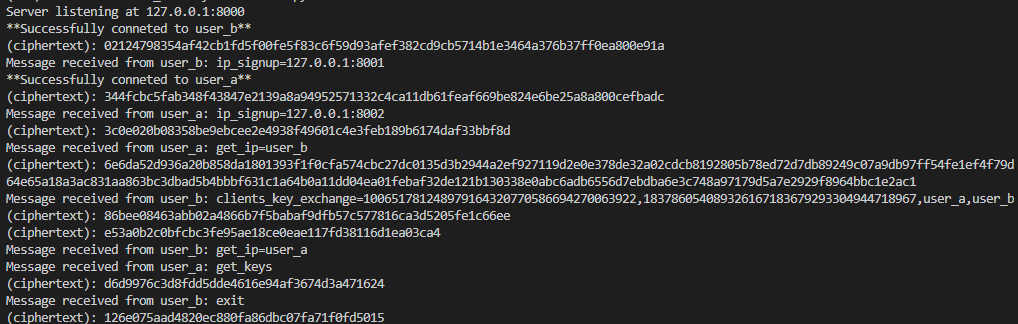
Trước khi các client giao tiếp với nhau cần xác thực với Verifier (server), server này sẽ đóng vai trò trung gian xác thực và phân phối khoá mã hoá. Quy trình xác thực như sau:



Hình .. Quy trình xác thực client-server



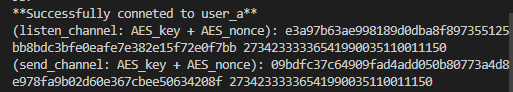
Hình .. Server xác thực client



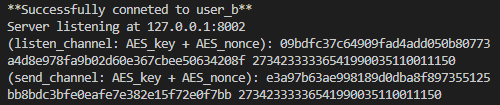
Hình .. Server tạo kết nối và phân phối khoá cho các client

### Giai đoạn trao đổi khoá

Trước khi truyền dữ liệu, các client sẽ được phân phối khoá từ server, khoá này sẽ sử dụng làm khoá mã hoá cho AES.



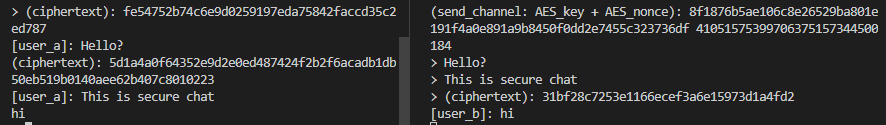
Hình .. Khoá mã hoá AES phân phối cho user\_b



Hình .. Khoá mã hoá AES phân phối cho user\_a

### Giai đoạn mã hoá, giải mã

Cuối cùng, client chia sẻ khoá mã hoá và trao đổi thông tin:



Hình .. Client trao đổi thông tin qua kênh truyền mã hoá

# Kết luận

Sau quá trình nghiên cứu, đề tài đã đạt được một số kết quả như sau:

* Nghiên cứu tổng quan về hệ mật đường cong Elliptic.
* Thuật toán, phương trình đường cong.
* Nghiên cứu các ứng dụng của đường cong Elliptic và chi tiết ứng dụng của đường cong Ellliptic trong giao thức trao đổi khoá.

Tuy nhiên, do vốn kiến thức có hạn nên đề tài còn chưa triển khai và so sánh kết quả ứng dụng của giao thức trong thực tế hiện nay cung như so sánh với các giao thức trao đổi khoá khác.

Trong tương lai, đề tài có thể phát triển hơn nữa, khắc phục các điểm hạn chế này và đánh giá kết quả một cách khách quan hơn.

# Tài liệu tham khảo

Sách tham khảo:

1. Bruce Schneier, ***“Applied Cryptography: Protocols, Algorithms and Source Code in C”***, 2017.
2. St Denis, Tom. ***“Cryptography for developers”***, 2006.

Tài liệu online:

1. <https://www.geeksforgeeks.org/blockchain-elliptic-curve-cryptography/>
2. <https://www.ibm.com/docs/en/linux-on-z?topic=keys-ec-diffie-hellman-csndedh>
3. <https://cryptobook.nakov.com/asymmetric-key-ciphers/ecdh-key-exchange>

# Phụ lục