Mục Lục

[**TÓM TẮT** 2](#_Toc167235149)

[**I. CƠ SỞ LÝ THUYẾT** 3](#_Toc167235150)

[**1. Hệ mật mã** 3](#_Toc167235151)

[**1.1 Hệ mật mã khóa đối xứng** 3](#_Toc167235152)

[**1.2 Hệ mật mã khóa công khai** 4](#_Toc167235153)

[**2. Thuật toán mã hóa AES** 5](#_Toc167235154)

[**2.1. Nguồn gốc ra đời của thuật toán AES** 5](#_Toc167235155)

[**2.2. Thuật toán AES** 6](#_Toc167235156)

[**3. Hệ mật trên đường cong Elliptic** 8](#_Toc167235157)

[**3.1. Cơ sở toán học** 8](#_Toc167235158)

[**3.2. Các phép toán trên đường cong elliptic** 9](#_Toc167235159)

[**4. Ứng dụng ECC trong thực tế** 11](#_Toc167235160)

[**II. Elliptic Curve Diffie-Hellman (ECDH)** 12](#_Toc167235161)

[**1. Các bước cơ bản của ECDH:** 12](#_Toc167235162)

[**2. Lợi ích của ECDH:** 13](#_Toc167235163)

[**3. Sử dụng ECDH trong thực tế** 13](#_Toc167235164)

[**III. Ứng dụng ECC trong việc xây dựng kênh chat chung** 14](#_Toc167235165)

[**1. Đặt vấn đề** 14](#_Toc167235166)

[**2. Các bước thực hiện** 15](#_Toc167235167)

[**3. Kết quả demo** 18](#_Toc167235168)

[**Tài liệu tham khảo** 21](#_Toc167235169)

**Hệ mật trên đường cong Elliptic và ứng dụng**

# **TÓM TẮT**

*Trong những năm gần đây, sự phát triển bùng nổ của công nghệ thông tin đã giúp cho việc trao đổi thông tin trở nên nhanh chóng, dễ dàng. Tuy nhiên, điều đó cũng dẫn đến việc nhiều cuộc tấn công quy mô lớn đã diễn ra với mục đích đánh cắp thông tin, giả mạo, phá hoại các cơ quan tổ chức. Điều này ảnh hưởng nặng nề đến các cá nhân, tổ chức hay thậm chí cả một quốc gia.*

*Để giải quyết tình hình trên, các vấn đề liên quan đến an toàn và bảo mật trở nên quan trọng hơn bao giờ hết. Các ý tưởng về an toàn, bảo mật dẫn đến sự phát triển của ngành mật mã học. Kỹ thuật này đã có từ xa xưa nhưng thời điểm đó nó còn rất thô sơ, ngày nay khi mạng máy tính trở nên phổ biến, người ta sử dụng mật mã học hiện đại. Các chuyên gia trong lĩnh vực này đã nghiên cứu, phát minh ra nhiều hệ mật mã khác nhau với mục đích che dấu thông tin để tránh việc kẻ xấu cố tình đánh cắp, phá hoại, ví dụ như RSA, Elgamal,…. Các phương pháp này rất an toàn, tuy nhiên độ dài khóa lớn nên trong một vài trường hợp nhất định sẽ không thể đáp ứng được.*

*Hiện nay, hệt mật trên đường cong Elliptic (Elliptic Curve Cryptography, viết tắt là ECC) là một trong những loại mật mã mạnh nhất hiện nay. Phương pháp này được sử dụng rộng rãi trong nhiều công ty tại Mỹ, như CloudFlare, đã sử dụng rộng rãi phương pháp đường cong Elliptic để bảo mật trên nhiều phương diện, từ các kết nối Https đến cách thức chuyển dữ liệu giữa các trung tâm dữ liệu.*

# **I. CƠ SỞ LÝ THUYẾT**

## **1. Hệ mật mã**

Mật mã là một lĩnh vực khoa học chuyên nghiên cứu về các phương pháp và kỹ thuật đảm bảo an toàn và bảo mật trong truyền tin liên lạc với mục đích đảm bảo các thế lực thù địch không thể ăn cắp thông tin để lợi dụng, phá hoại.

Một hệ mật mã bao gồm 4 thành phần cơ bản:

* Bản rõ (hay plaintext, là thông tin ở dạng đọc được)
* Thuật toán mã hóa (là một phép biến đổi với bản rõ là đầu vào, sử dụng một khóa để cho ra một bản mã)
* Bản mã (hay ciphertext, là thông tin ở dạng không thể đọc được)
* Thuật toán giải mã (là phép biến đổi cho phép người nhận thu được bản rõ từ bản mã bằng cách sử dụng một khóa)

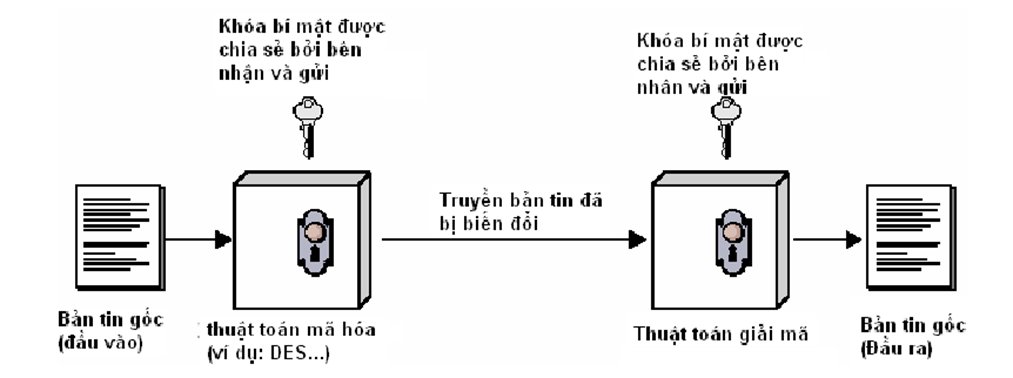
Mục đích của mật mã là làm cho kẻ thù không thể tái tạo lại bản rõ từ bản mã.

Hệ mật mã có khóa dùng chung cho cả thuật toán mã hóa và giải mã gọi là hệ mật mã đối xứng, còn hệ mật mã có khóa dùng cho thuật toán mã hóa khác với khóa dùng cho thuật toán giải mã gọi là hệ mật mã khóa phi đối xứng, hay còn gọi là hệ mật mã khóa công khai

### **1.1 Hệ mật mã khóa đối xứng**

Hệ mật má khóa đối xứng hay hệ mật mã khóa bí mật, là mô hình mà trong đó, khóa của hai thuật toán sinh mã và giải mã là giống nhau và ví mật đối với tất cả người khác, nói cách khác, hai bên gửi và nhận chia sẻ chung một khóa bí mật duy nhất. Vai trò của hai bên gửi và nhận giống nhau và có thể hoán đổi vai trò cho nhau.

Các thuật toán thông dụng được dùng trong hệ mật mã khóa đối xứng có thể kể đến như DES, AES,…



Ưu điểm:

* Mô hình khá đơn giản
* Dễ dàng tạo ra thuật toán mã hóa đối xứng cho cá nhân
* Dễ cài đặt, hoạt động hiệu quả
* Hoạt động nhanh do tốc độ mã hóa, giải mã cao

Tuy nhiên, hệ mật mã đối xứng cũng có những nhược điểm lớn như:

* Quản lý khóa: Số lượng khóa bí mật mà mỗi công ty hay cá nhân cần thiết lập với các đối tác khác có thể khá lớn, do đó việc quản lý lưu trữ gặp nhiều khó khăn
* Phân phối khóa: nếu hai bên gửi và nhận ở xa nhau và chỉ có thể liên lạc qua một kênh truyền tin thông thường, việc trao đổi khóa có thể bị nghe trộm

### **1.2 Hệ mật mã khóa công khai**

Vấn đề phát sinh trong các hệ thống mã hóa đối xứng là việc quy ước chung mã khóa k giữa người gửi A (Alice) và người nhận B (Bob). Trên thực tế, nhu cầu thay đổi nội dung của mã khóa k là cần thiết, do đó, cần có sự trao đổi thông tin về mã khóa k giữa A và B. Để bảo mật mã khóa k, A và B phải trao đổi với nhau trên một kênh liên lạc thật sự an toàn và bí mật. Tuy nhiên, rất khó có thể bảo đảm được sự an toàn của kênh liên lạc nên mã khóa k vẫn có thể bị phát hiện bởi người C.

Một hệ thống khóa công khai sử dụng hai loại khóa trong cùng một cặp khóa: khóa công khai (public key) được công bố rộng rãi và được sử dụng trong mã hóa thông tin, khóa riêng (private key) chỉ do một người nắm giữ và được sử dụng để giải mã thông tin đã được mã hóa bằng khóa công cộng. Các phương pháp mã hóa này khai thác những ánh xạ f mà việc thực hiện ánh xạ ngược f -1 rất khó so với việc thực hiện ánh xạ f. Chỉ khi biết được mã khóa riêng thì mới có thể thực hiện được ánh xạ ngược f -1.

A diagram of a key to a document

Description automatically generated

Khi áp dụng hệ thống mã hóa khóa công cộng, người A sử dụng mã khóa công cộng để mã hóa thông điệp và gửi cho người B. Do biết được mã khóa riêng nên B mới có thể giải mã thông điệp mà A đã mã hóa. Người C nếu phát hiện được thông điệp mà A gửi cho B, kết hợp với thông tin về mã khóa công cộng đã được công bố, cũng rất khó có khả năng giải mã được thông điệp này do không nắm được mã khóa riêng của B.

## **2. Thuật toán mã hóa AES**

### **2.1. Nguồn gốc ra đời của thuật toán AES**

Năm 1997 nhận thấy nguy cơ mã hóa DES là kích thước khóa ngắn có thể bị phá vỡ, do đó Viện chuẩn quốc gia Hoa Kỳ US NIST đã kêu gọi xây dựng một phương pháp mã hóa mới nhằm thay thế thuật toán này. Đến tháng 10/2000 thuật toán có tên là Rijndael được chọn làm mật mã nâng cao và xuất bản là chuẩn FIPS PUB 197. Vào 11/2001 và được đổi tên thành Andvanced Encryption Standard (AES). Cơ chế mã hóa của AES sử dụng mạng thay thế hoàn vị (SPN - Substitution-Permutation Network). Đầu vào và đầu ra là khối dữ liệu 128 bit. Khóa dùng để mã hóa và giải mã của giải thuật AES là 128 ,192 hoặc 256 bit. Kích thước khóa quyết định số vòng lặp trong mã hóa và giải mã.

### **2.2. Thuật toán AES**

Thuật toán AES khá phức tạp, được mô tả khái quát gồm 3 bước như sau:

* Vòng khởi tạo chỉ gồm phép *AddRoundKey*
* Vòng lặp gồm 4 phép biển đổi lần lượt: *SubBytes, ShiftRows, MixColumns, AddRoundKey.*
* Vòng cuối gồm các phép biến đổi giống vòng lặp và không có phép *MixColumns*.

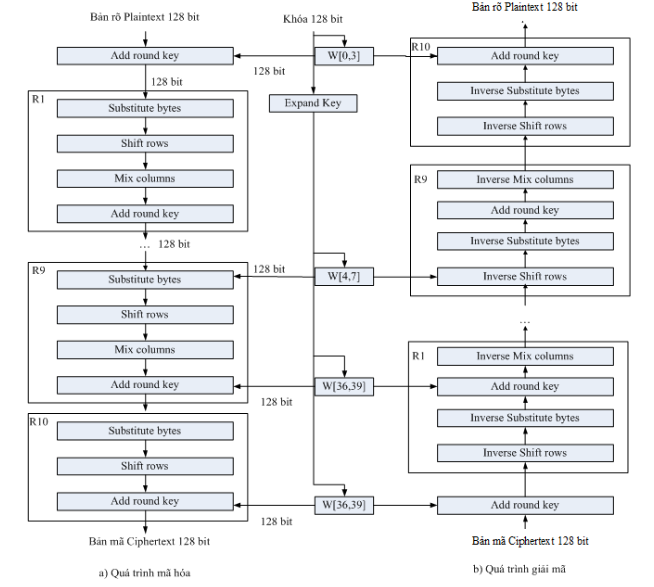
**Đặc tả chi tiết thuật toán mã hóa AES**

Cấu trúc sơ đồ mã hóa gồm: vòng khởi tạo, Nr-1 vòng lặp và vòng kết thúc. Trong đó vòng khởi tạo chỉ có phép biến đổi *AddRounKey*, vòng lặp gồm 4 phép biến đổi chính: *SubBytes*, *ShiftRows*, *MixColumns*, *AddRounKey*; vòng kết thúc khác với vòng lặp chính ở chỗ không có phép *MixColumns*.

Riêng đối với cấu trúc giải mã trong AES gồm 2 chế độ giải mã:

* Ở cấu trúc giải mã ngược, gồm vòng khởi tạo, Nr-1 vòng lặp và vòng kết thúc. Trong đó vòng khởi tạo chỉ có phép biến đổi *AddRounKey*, vòng lặp gồm lần lượt 4 phép biến đổi chính: *InvShiftRows*, *InvSubBytes*, *AddRounKey*, *InvMixColumns*; vòng kết thúc khác với vòng lặp chính ở chỗ không có phép *InvMixColumns*.
* Ngược lại với cấu trúc giải mã ngược là cấu trúc giải mã xuôi, việc ngược lại thể hiện ở điểm: trong cấu trúc giải mã xuôi việc sắp xếp các phép biến đổi ngược giống hệt với cấu trúc mã hóa, cụ thể bao gồm: vòng khởi tạo, Nr-1 vòng lặp và vòng kết thúc. Trong đó vòng khởi là phép *AddRounKey*; ở vòng lặp thứ tự các phép biến đổi ngược lần lượt là: *InvSubBytes*, *InvShiftRows*, *InvMixColumns*, *AddRounKey*; vòng kết thúc giống vòng lặp nhưng được lược bỏ phép *InvMixColumns*.

Một điểm khác biệt nữa trong hai cấu trúc giải mã ngược và giải mã xuôi đó là: trong giải mã ngược khóa vòng giải mã chính là khóa vòng mã hóa với thứ tự đảo ngược; riêng trong giải mã xuôi thì khóa giải mã ngoài việc đảo ngược thứ tự khóa vòng mã hóa còn phải thực hiện phép *InvMixColumns* đối với các khóa vòng của vòng lặp giải mã



## **3. Hệ mật trên đường cong Elliptic**

Hệ mật mã khóa công khai dựa trên việc sử dụng các bài toán khó giải quyết, các bài toàn này yêu cầu số lượng phép tính cần thiết để tìm ra lời giải là rất lớn. Trong lịch sử 20 năm của ngành mã hóa bất đối xứng đã có nhiều đề xuất khác nhau cho dạng bài toán như vậy, tuy nhiên chỉ có hai trong số các đề xuất đó còn tồn tại đến ngày nay. Hai bài toán đó bao gồm: bài toán logarit rời rạc và bài toán phân tích thừa số nguyên tố.

Năm 1985, Neal Koblitz và Victor S.Miller đã nghiên cứu và đưa ra đề xuất ứng dụng đường cong elliptic trên trường hữu hạn.

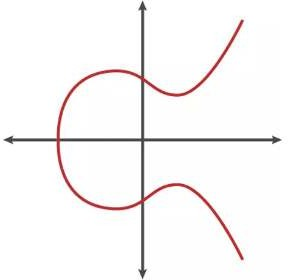
### **3.1. Cơ sở toán học**

Đường cong elip là tập hợp các điểm thỏa mãn một phương trình toán học cụ thể. Phương trình cho một đường cong elip:

y2 = x3 + Ax + B

Với điều kiện 4A3 + 27B2 # 0.

Đồ thị được biểu diễn như sau:



**Tính chất:**

* Đồ thị đối xứng qua trục hoành, vì vậy các điểm nằm trên đường cong khi lấy đối xứng qua Ox vẫn sẽ nằm trên đường cong
* Một đường thẳng bất kỳ có thể cắt đường cong tại tối đa 3 điểm

Dựa trên những tính chất nói trên, người ta đã nghiên cứu và phát hiện ra một phương pháp mới cho kỹ thuật mã hóa nói chung và chứng thực nói riêng, kỹ thuật mã hóa dựa trên đường cong elliptic.

### **3.2. Các phép toán trên đường cong elliptic**

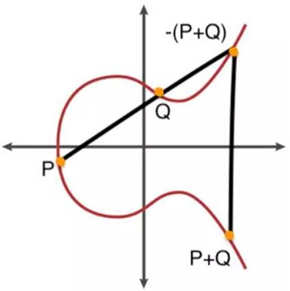
**Phép cộng**

Đường cong Elliptic có một tính chất: "Nếu hai điểm P và Q nằm trên đường cong, thì điểm P+Q cũng sẽ nằm trên đường cong".

Điểm này được xác định như sau:

* Vẽ đường thẳng nối 2 điểm P và Q, đường thẳng này sẽ cắt đường cong tại một điểm -R = -(P+Q)
* Lấy đối xứng của điểm này qua trục hoành, ta sẽ có được điểm R = P + Q

Chú ý: Nếu 3 điểm trên đường cong Elliptic là thẳng hàng, thì tổng của chúng bằng 0.

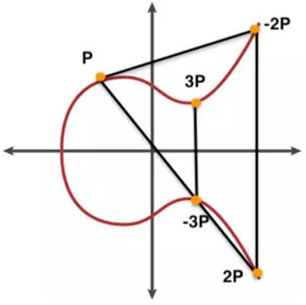


Trong trường hợp 2 điểm P và Q trùng nhau, đường thẳng A là tiếp tuyến với đường cong tại điểm P. Thực hiện tương tự ta được điểm 2P.

**Phép nhân**

Phép nhân trên đường con Elliptic tương đương với việc thực hiện phép cộng nhiều lần

Ví dụ, trong phép toán tính 3P, đầu tiên ta tính 2P bằng cách tính P+P. Lúc này, đường thẳng nối P với P sẽ là tiếp tuyến của đường cong và nó cắt đường cong tại điểm -2P, lấy đối xứng qua trục hoành ta được điểm 2P. Tiếp tục ta vẽ đường thẳng nối giữa P và 2P, thực hiện tương tự ta thu được điểm 3P.

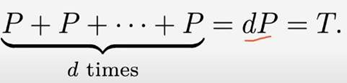


Do cách tính toán trên, ta có thể dễ dàng tính toán được phép nhân k\*P khi biết k và P, nhưng hoàn toàn không thể tính toán được theo chiều ngược lại, tức phép chia. Đó cũng chính là tính chất đặc trưng của mã hoá bất đối xứng.

Đường cong elliptic trên trường hữu hạn có thể được dùng để thực hiện lược đồ trao đổi khóa Diffie-Hellman, ElGamal và chữ ký NIST. Những hệ thống này có khả năng cung cấp tính an toàn tương đương với các hệ mật mã khóa công khai hiện tại nhưng với kích thước khóa nhỏ hơn (Ví dụ như một khóa trên ECC có độ dài 160 bit thì tương ứng với khóa có độ dài 1024 bit trong RSA). Việc dùng khóa có kích thước nhỏ hơn sẽ giúp giảm băng thông và bộ nhớ yêu cầu.

Độ khó của bài toán logarit rời rác trên đường cong Elliptic (ECDLP) là yếu tố quyết định cho mức độ an toàn của tất cả các lược đồ hệ mật đường cong Elliptic.

Định nghĩa: Cho đường cong elliptic E. Ta xét một điểm P và một điểm khác T. Bài toán logarigh rời rạc nhằm tìm số nguyên d thỏa mãn:



Hay nói cách khác, d = logPT.

**Số điểm của EC:**

Theo định lý Hass, cho đường cong E modun p, số điểm trên đường cong ký hiệu bởi #E và bị chặn bởi:



Khi đó, #E ≈ p. Nếu ta cần một đường cong với số điểm 2160, ta phải sử dụng số nguyên tố cỡ 160 bit

Tính an toàn:

Mọi giao thức EC dựa trên tính khó giải của bài toán ECDLP

Nếu EC được chọn cẩn thận, thuật toán tốt nhất để tính ECDLP cần xấp xỉ √p bước. Ví dụ, với p ≈ 2160, kẻ tấn công cần thực hiện ≈ 280 bước.

## **4. Ứng dụng ECC trong thực tế**

ECC hiện đang được sử dụng trong một loạt các ứng dụng: chính phủ Mỹ sử dụng để bảo vệ thông tin liên lạc nội bộ, các dự án Tor sử dụng để giúp đảm bảo ẩn danh, đây cũng là cơ chế được sử dụng để chứng minh quyền sở hữu trong Bitcoins, cung cấp chữ ký số trong dịch vụ iMessage của Apple, để mã hóa thông tin DNS với DNSCurve, và là phương pháp tốt để xác thực cho các trình duyệt web an toàn qua SSL/TLS. Thế hệ đầu tiên của thuật toán mã hóa khóa công khai như RSA và Diffie-Hellman vẫn được duy trì trong hầu hết các lĩnh vực, nhưng ECC đang nhanh chóng trở thành giải pháp thay thế cho RSA.

Cụ thể hơn nếu truy cập vào phiên bản HTTPS của các trang web phổ biến, như Google.com, Amazon.com, Facebook.com… từ một trình duyệt như Chrome hoặc Firefox, trình duyệt sẽ sử dụng ECC – như là sử dụng ECDHE\_ECDSA. ECDHE là viết tắt của Elliptic Curve Diffie Hellman Ephemeral và là một cơ chế trao đổi khóa dựa trên đường cong Elliptic. ECDSA là Elliptic Curve Digital Signature Algorithm là cơ chế tạo chữ ký số để xác thực kết nối với máy chủ.

Sự cải thiện hiệu suất của ECDSA hơn RSA là rất rõ ràng. Ngay cả với một phiên bản cũ của OpenSSL không có tối ưu cho ECC, tạo một chữ ký ECDSA với khóa 256-bit là nhanh hơn 20 lần so với một chữ ký RSA với khóa 2048-bit.

Trong kỷ nguyên công nghệ thông tin và truyền thông hiện nay, nhu cầu đảm bảo an toàn thông tin là không thể thiếu. Với việc khóa mã hóa có độ dài ngày tăng dần theo thời gian, ECC đang là ứng viên phù hợp để thay thế RSA trong việc tạo ra các khóa mã ngắn hơn mà vẫn đảm bảo an toàn, từ đó có thể triển khai trên nhiều nền tảng thiết bị từ các mạch điện tử đơn giản đến máy tính lớn, dễ dàng tạo ra hệ thống mạng đáng tin cậy phục vụ tốt hơn cho xã hội.

# **II. Elliptic Curve Diffie-Hellman (ECDH)**

Elliptic Curve Diffie-Hellman (ECDH) là một giao thức trao đổi khóa sử dụng toán học của đường cong elliptic để cho phép hai bên tạo ra một khóa bí mật chung, ngay cả khi họ đang trao đổi thông tin qua một kênh không an toàn. ECDH là phiên bản nâng cao của giao thức Diffie-Hellman, sử dụng đường cong elliptic thay vì số nguyên lớn để cung cấp mức độ bảo mật tương đương nhưng với khóa ngắn hơn, làm cho nó hiệu quả hơn về tính toán và băng thông.

## **1. Các bước cơ bản của ECDH:**

***B1: Khởi tạo khóa riêng và khóa công khai***

Mỗi bên (Alice và Bob) sẽ tạo một cặp khóa riêng và khóa công khai sử dụng đường cong elliptic đã chọn.

* Khóa riêng: Một số nguyên ngẫu nhiên *d*, có kích thước từ 256 đến 521 bit (tùy thuộc vào đường cong elliptic được sử dụng).
* Khóa công khai: Một điểm trên đường cong elliptic, được tính bằng cách nhân khóa riêng với điểm sinh *G* (generator point) trên đường cong. Công thức là *Q=d.G*

Ví dụ:

* Alice chọn khóa riêng *dA* và tính khóa công khai *QA=dA.G*
* Bob chọn khóa riêng *dB* và tính khóa công khai *QB=dB.G*

***B2: Trao đổi khóa công khai***

Alice và Bob trao đổi khóa công khai của mình qua một kênh không an toàn. Giả sử Alice gửi *QA cho Bob, và Bob gửi QB cho Alice.*

***B3: Tính khóa bí mật chung***

Mỗi bên sử dụng khóa riêng của mình và khóa công khai của bên kia để tính toán khóa bí mật chung:

* Alice tính *SA = dA. QB*
* Bob tính *SB = dB. QA*

Do tính chất của nhóm điểm trên đường cong elliptic, *SA* và *SB* sẽ giống nhau.

* *SA = dA.(dB.G) = dA.dB.G*
* *SB = dB.(dA.G) = dB.dA.G*

***B4: Sử dụng khóa bí mật chung***

Khóa bí mật chung này thường không được sử dụng trực tiếp. Thay vào đó, nó sẽ được đưa qua một hàm dẫn xuất khóa (Key Derivation Function - KDF) như HKDF (HMAC-based Extract-and-Expand Key Derivation Function) để tạo ra khóa đối xứng dùng trong các thuật toán mã hóa như AES.

## **2. Lợi ích của ECDH:**

* Bảo mật cao với khóa ngắn hơn: So với Diffie-Hellman truyền thống, ECDH sử dụng khóa ngắn hơn nhưng vẫn đảm bảo mức độ bảo mật tương đương. Ví dụ, bảo mật 256-bit của ECDH tương đương với bảo mật 3072-bit của Diffie-Hellman.
* Hiệu quả tính toán: Phép toán trên đường cong elliptic ít tốn kém hơn về mặt tính toán so với phép toán số nguyên lớn của Diffie-Hellman.
* Bảo mật mạnh: ECDH dựa trên bài toán logarithm rời rạc trên đường cong elliptic, một bài toán được coi là khó giải quyết bằng các phương pháp hiện tại, kể cả các máy tính lượng tử.

## **3. Sử dụng ECDH trong thực tế**

ECDH được sử dụng rộng rãi trong nhiều giao thức bảo mật hiện đại, bao gồm:

* TLS/SSL: Bảo mật các kết nối internet, đảm bảo rằng dữ liệu truyền giữa trình duyệt và máy chủ được mã hóa.
* SSH: Bảo mật các kết nối điều khiển từ xa, cho phép người dùng kết nối an toàn với máy chủ từ xa.
* PGP/GPG: Bảo mật email, đảm bảo rằng chỉ người nhận dự kiến mới có thể đọc được nội dung email.
* Wi-Fi: Trong các giao thức bảo mật mạng không dây như WPA3, ECDH giúp bảo vệ kết nối mạng không dây khỏi các cuộc tấn công.

# **III. Ứng dụng ECC trong việc xây dựng kênh chat chung**

## **1. Đặt vấn đề**

Khi xây dựng một ứng dụng như kênh chat, việc bảo vệ dữ liệu truyền qua mạng là rất quan trọng để đảm bảo tính riêng tư và an toàn của người dùng. Dưới đây là một số ưu điểm khi kết hợp ECC (Elliptic Curve Cryptography) hoặc các phương pháp mã hóa khác với kênh chat:

* **Bảo mật thông tin**: Khi người dùng gửi tin nhắn qua mạng, dữ liệu của họ có thể bị tấn công và đánh cắp bởi các bên thứ ba không mong muốn. Sử dụng ECC để mã hóa dữ liệu giúp bảo vệ thông tin truyền tải tránh khỏi các mối đe dọa này.
* **Chống lại tin tặc**: Các tin tặc có thể cố gắng theo dõi và nghe trộm thông tin giao tiếp giữa các người dùng trong kênh chat. Sử dụng ECC để tạo và trao đổi các khóa bí mật giữa các thiết bị kết nối giúp ngăn chặn các cuộc tấn công nghe trộm.
* **Đảm bảo tính toàn vẹn của dữ liệu**: ECC không chỉ được sử dụng để mã hóa dữ liệu, mà còn để xác thực tính toàn vẹn của dữ liệu. Điều này đảm bảo rằng tin nhắn gửi và nhận không bị sửa đổi trên đường truyền.
* **Tăng cường bảo mật toàn diện**: Bằng cách kết hợp ECC với các phương pháp mã hóa khác, admin có thể tăng cường bảo mật toàn diện của kênh chat của mình, cung cấp một môi trường truyền thông an toàn và bảo mật cho người dùng.

Do đó, trong bài tập lớn lần này nhóm em sẽ tiến hành demo việc ứng dụng ECC trong xây dựng kênh chat chung.

## **2. Các bước thực hiện**

Các bước thực hiện của bài code chủ yếu dựa trên quá trình trao đổi khóa và thiết lập kết nối an toàn giữa server và client bằng cách sử dụng thuật toán ECDH (Elliptic Curve Diffie-Hellman) và hàm băm HKDF (HMAC-based Extract-and-Expand Key Derivation Function). Các đoạn tin nhắn được mã hóa và giải mã bằng thuật toán AES-GCM (Advanced Encryption Standard - Galois/Counter Mode):

**Gửi và nhận khóa công khai:**

Server:

A computer code on a black background

Description automatically generated

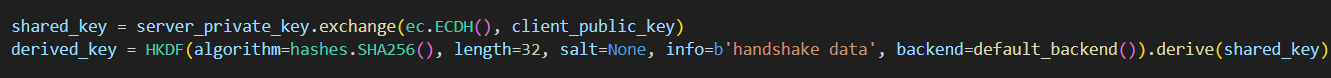
Client:

A computer code on a black background

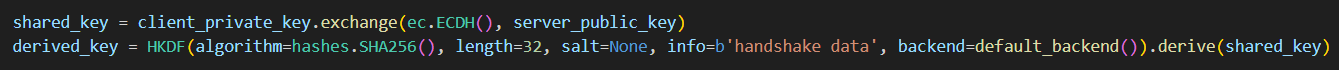
Description automatically generated

**Trao đổi khóa chung và tạo khóa dẫn xuất:** Sử dụng thuật toán ECDH, Client/Server tính toán khóa chia sẻ (shared key) từ khóa bí mật của mình và khóa công khai của người kia. Sau đó, sử dụng hàm băm HKDF để tạo khóa dẫn xuất từ khóa chia sẻ. Khóa dẫn xuất này được sử dụng để mã hóa và giải mã tin nhắn giữa máy chủ và client.

Server:



Client:



**Gửi tin nhắn:**

Server:

A screen shot of a computer code

Description automatically generated

Client:

A screen shot of a computer code

Description automatically generated

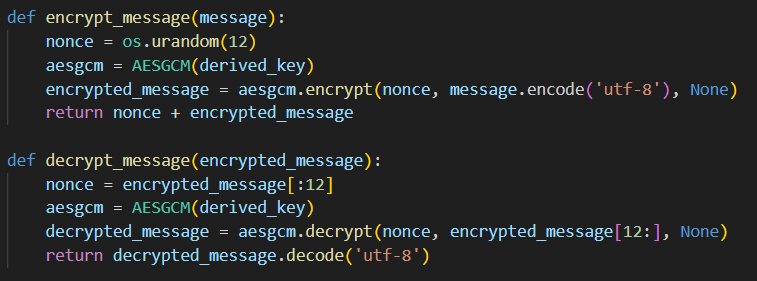
**Hàm sử dụng để mã hóa và giải mã tin nhắn**

Server:

A screen shot of a computer code

Description automatically generated

Client:



## **3. Kết quả demo**

Server gửi tin nhắn:

**A screenshot of a computer

Description automatically generated**

Client nhận được tin nhắn và gửi lại:

A screenshot of a computer

Description automatically generated

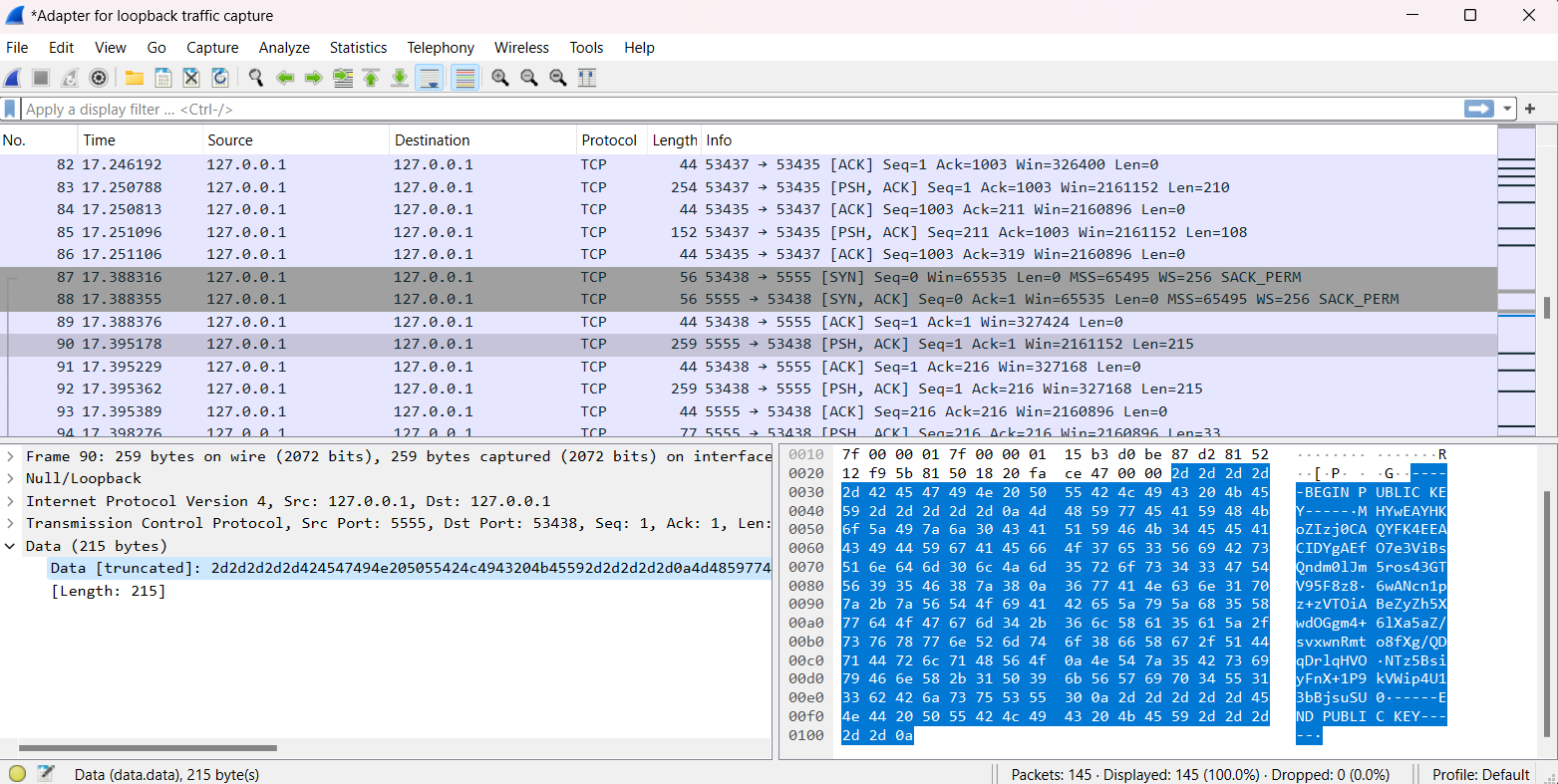
Server nhận tin nhắn thành công

A screenshot of a chat

Description automatically generated

**Tiến hành kiểm tra wireshark:**

Server gửi public key:



Client gửi public key:

A screenshot of a computer

Description automatically generated

Server gửi tin nhắn đã mã hóa:

A screenshot of a computer

Description automatically generated

Client gửi tin nhắn đã mã hóa:

A screenshot of a computer

Description automatically generated

# **Tài liệu tham khảo**

1. BÀI GIẢNG MẬT MÃ HỌC CƠ SỞ - Đỗ Xuân Chợ

2. The Elliptic Curve Diffie-Hellman (ECDH) - Rakel Haakegaard and Joanna Lang

<http://koclab.cs.ucsb.edu/teaching/ecc/project/2015Projects/Haakegaard+Lang.pdf>

3. Elliptic Curve Cryptography

<https://dl.acm.org/doi/pdf/10.1145/1386853.1378356>