ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN CƠ TIN HỌC

Đàm Hải Đăng

GIAO THỨC XÁC THỰC DỰA TRÊN THUẬT TOÁN ELGAMAL TRÊN ĐƯỜNG CONG ELLIPTIC

Khóa luận tốt nghiệp đại học hệ chính quy
Ngành Toán tin
(Chương trình đào tạo chuẩn)

Hà Nội - 2025

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN CƠ TIN HỌC

Đàm Hải Đăng

GIAO THỨC XÁC THỰC DỰA TRÊN THUẬT TOÁN ELGAMAL TRÊN ĐƯỜNG CONG ELLIPTIC

Khóa luận tốt nghiệp đại học hệ chính quy
Ngành Toán tin
(Chương trình đào tạo chuẩn)

Cán bộ hướng dẫn: TS Nguyễn Hải Vinh

Hà Nội - 2025

Lời cảm ơn

Tôi muốn bày tỏ lòng biết ơn đến người hướng dẫn của tôi là Tiến sĩ Nguyễn Hải Vinh. Thầy đã đưa ra những lời khuyên quý báu và những gợi ý sâu sắc giúp tôi hoàn thành mục tiêu của mình trong khóa luận này.

Tôi cũng muốn cảm ơn gia đình tôi, đã luôn khuyến khích tôi tìm kiếm nguồn cảm hứng trong Toán học và Mật mã.

Do những hạn chế về thời gian và nguồn lực, khóa luận này không thể tránh khỏi những thiếu sót. Tôi rất mong nhận được những ý kiến đóng góp và gợi ý mang tính xây dựng từ bất kỳ ai quan tâm đến khóa luận của em.

Hà Nội, Tháng Năm 2025

Đàm Hải Đăng

Mục lục

Mở đầu

Mật mã học là lĩnh vực khoa học nhằm bảo vệ thông tin thông qua việc chuyển đổi dữ liệu gốc thành dạng không thể hiểu được bằng quá trình mật mã hóa (hay còn gọi là mã hóa). Quá trình giải mã ngược lại, khôi phục bản rõ từ bản mã, đòi hỏi kiến thức về một khóa bí mật.

Hiện nay, mật mã học được phân thành hai nhánh chính: mật mã hóa khóa đối xứng và mật mã hóa khóa công khai. Các hệ thống khóa đối xứng yêu cầu các bên tham gia giao tiếp phải chia sẻ trước một khóa bí mật duy nhất, được sử dụng cho cả quá trình mã hóa và giải mã. Ngược lại, các hệ thống khóa công khai sử dụng một cặp khóa có liên hệ toán học: một khóa công khai và một khóa riêng tư. Khóa công khai có thể được phân phối rộng rãi và thường dùng để mã hóa thông điệp hoặc xác thực chữ ký số, trong khi khóa riêng tư phải được giữ bí mật tuyệt đối bởi chủ sở hữu và được dùng để giải mã hoặc tạo chữ ký số. Thuật toán ElGamal là một ví dụ tiêu biểu trong số các hệ mật mã hóa khóa công khai.

Vào năm 1985, Koblitz [4] và Miller [7] đã độc lập đề xuất việc triển khai hệ mật mã khóa công khai sử dụng đường cong elliptic trên trường hữu hạn, mà ngày nay được biết đến với tên gọi Mật mã Đường cong Elliptic. Hệ mật mã ElGamal trên đường cong elliptic là một phiên bản tương tự trong lĩnh vực mật mã hóa khóa công khai của các lược đồ mã hóa ElGamal truyền thống, nhưng nó dựa trên Bài toán Logarithm Rời rạc trên Đường cong Elliptic. Bài toán logarithm rời rạc trên đường cong elliptic được xem là khó hơn đáng kể so với bài toán logarithm rời rạc trong các nhóm được sử dụng bởi các thuật toán khác (ví dụ như nhóm Z_p^* dùng trong DSA - Digital Signature Algorithm) và bài toán phân tích thừa số nguyên tố (nền tảng của RSA). Do đó, hệ

mật mã đường cong elliptic có thể đạt được mức độ bảo mật tương đương với các hệ mật mã khác trong khi sử dụng khóa có kích thước nhỏ hơn. Ví dụ, một hệ mật mã đường cong elliptic với khóa có kích thước 160 bit được xem là có độ an toàn tương đương với các hệ mật mã RSA và DSA với khóa có kích thước 1024 bit [5].

Trong một số văn bản tiếng Việt, encode và decode thường được dịch là biên mã và giải biên mã. Biên mã và giải biên mã là các bước thiết yếu trong quá trình mã hóa và giải mã của mật mã hóa đường cong elliptic. Phương pháp ánh xạ này chuyển đổi một thông điệp bản rõ thành tọa độ của một điểm trên đường cong elliptic. Baccouri và cộng sự [2] đã đề xuất một phương pháp biên mã mới dựa trên các tham số tạm thời. Việc sử dụng các tham số biên mã tạm thời cho mỗi phiên liên lạc giúp tăng cường đáng kể tính bảo mật bằng cách ngăn chặn các rủi ro tiềm ẩn liên quan đến việc sử dụng các giá trị tĩnh hoặc tái sử dụng. Tuy nhiên, tồn tại một vấn đề: khi Alice cần trao đổi các tham số tạm thời này với Bob, thông điệp trao đổi có thể bị xâm phạm, dẫn đến nguy cơ an toàn cho phiên liên lạc. Vì vậy, việc đảm bảo tính bảo mật và toàn vẹn của các tham số này, đồng thời xác thực cả hai bên giao tiếp là rất quan trọng.

Khóa luận này nghiên cứu, tìm hiểu một phương pháp giao thức xác thực mới dựa trên thuật toán ElGamal trên đường cong elliptic được đề xuất bởi Baccouri và cộng sự [1] nhằm giải quyết vấn đề trên, đạt được cả tính bảo mật cho các tham số biên mã được trao đổi và cơ chế xác thực, thông qua việc tận dụng chính các tham số tạm thời. Phương pháp này cung cấp một giải pháp toàn diện, không chỉ đảm bảo tính riêng tư của các tham số biên mã mà còn xác minh hiệu quả danh tính của các bên tham gia giao tiếp, đồng thời tôi phân tích bảo mật của phương pháp, tiến hành hiện thực hóa giao thức thông qua mô phỏng trên môi trường SageMath và mô phỏng một số kịch bản tấn công cơ bản.

Khóa luận bao gồm ba chương: chương một trình bày nền tảng toán học của ECC và ElGamal. Chương hai mô tả phương pháp đề xuất của Baccouri cho xác thực và trao đổi tham số biên mã. Chương ba bao gồm phân tích bảo mật và thực nghiệm. Cuối cùng, tôi đưa ra kết luận về khóa luận này.

Chương 1

Cơ sở lý thuyết

1.1 Số học mô đun

Số học mô-đun với mô-đun (modulus) p bao gồm các phép toán số học (cộng, trừ, nhân, chia, nhân nghịch đảo) thực hiện trên tập hợp các số nguyên 0, 1, 2, ..., p-1. Nếu kết quả của bất kỳ phép toán nào nằm ngoài phạm vi này, nó sẽ được đưa về lại phạm vi [0, p - 1].

1.1.1 Phép cộng

Với p = 19, a = 30, và b = 35:

$$a + b \pmod{p} = 30 + 35 \pmod{19} = 65 \pmod{19} = 8$$

Vì 65 nằm ngoài phạm vi [0, 18], ta trừ đi 19 liên tiếp cho đến khi đạt được kết quả 8, thuộc phạm vi này.

1.1.2 Phép trừ

Với p = 19, a = 30, b = 35,

$$a - b \pmod{p} = 30 - 35 \pmod{19} = -5 \pmod{19} = 14.$$

Trong số học mô-đun, ta có thể tính toán các giá trị âm bằng cách "lùi lại" trong tập hợp. Ví dụ, nếu kết quả của phép toán a - b là một số âm và nằm ngoài phạm vi [0, 18], ta sẽ "gói" nó lại trong phạm vi này bằng cách liên tục cộng -5 với 19 cho đến khi thu được một giá trị nằm trong khoảng [0, 18].

1.1.3 Phép nhân

Ví dụ, với p = 19, a = 30, và b = 35, ta có:

$$a * b \pmod{p} = 30 * 35 \pmod{19} = 1050 \pmod{19} = 5.$$

Do 1050 vượt quá phạm vi [0, 18], ta liên tục trừ đi 19 cho đến khi thu được kết quả (là 5) nằm trong khoảng này.

1.1.4 Phép chia

Trong trường F_p, phép chia $a \div b$ được định nghĩa là $a * b^{-1}$, nói cách khác, a chia mod cho b tương đương với a nhân với nghịch đảo nhân mod của b. Quy tắc này cho phép chúng ta thực hiện phép chia bằng cách tìm nghịch đảo nhân của số chia, sau đó thực hiện một phép nhân duy nhất.

1.1.5 Phép nhân nghịch đảo

Giá trị nghịch đảo của một số p là một số mà khi nhân với p sẽ bằng giá trị 1 trong một phép mô đun nào đó. Ví dụ, 5 là giá trị nghịch đảo của 4 (và ngược lại) trong mod 19, vì 5 * 4 = 20 = 1 (mod 19). Nó chỉ có thể tính được với các modulo nguyên tố. Một số thuật toán như thuật toán Euclid mở rộng có thể được sử dụng để tìm nghịch đảo trong trường hợp này. Có thể nói, việc tìm phần tử nghịch đảo là một thao tác tốn kém.

1.2 Mật mã đường cong elliptic

Trong mật mã khóa công khai, mỗi người dùng hoặc thiết bị tham gia vào giao tiếp sở hữu một cặp khóa: một khóa công khai và một khóa riêng tư, cùng với một tập hợp các phép toán liên quan đến các khóa này để thực hiện các thao tác mật mã hóa. Khóa riêng tư chỉ được biết bởi người sở hữu nó, trong khi khóa công khai được phân phối cho tất cả những người dùng tham gia giao tiếp.

Phương trình bậc ba tổng quát cho đường cong elliptic là

$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

Tuy nhiên, ta có thể đưa về dạng thu gọn: $y^2 = x^3 + ax + b$ với điều kiện $4a^3 + 27b^2 \neq 0$. Mỗi cặp giá trị của các tham số đường cong a và b cùng với định nghĩa trường cơ sở sẽ xác định một đường cong elliptic.

Gọi $E_p(a, b)$ là tập hợp bao gồm tất cả các điểm (x, y) thỏa mãn phương trình đường cong elliptic cùng với phần tử tại vô cùng O. Một nhóm giao hoán có thể định nghĩa trên tập hợp $E_p(a, b)$ thông qua một phép toán cộng điểm với hai giá trị a, b cụ thể. Nếu P, Q và R là các điểm trên $E_p(a, b)$, thì các tính chất giao hoán (P + Q = Q + P), kết hợp ((P + Q) + R = P + (Q + R)), tồn tại phần tử đơn vị (là phần tử tại vô cùng O, sao cho P + O = O + P = P), và tồn tại phần tử nghịch đảo (với mỗi điểm P tồn tại -P sao cho P + (-P) = O) đều được thỏa mãn. Một nhóm như vậy có thể được sử dụng để xây dựng một bài toán tương tự của bài toán Logarithm rời rạc, vốn là nền tảng cho hệ mật mã khóa công khai ElGamal. Bài toán này trên đường cong elliptic được gọi là bài toán Logarithm rời rạc trên đường cong Elliptic (Elliptic Curve Discrete Logarithm Problem - ECDLP) và độ khó tính toán của nó chính là cơ sở bảo mật cho ECC.

Trong ECC, khóa công khai là một điểm trên đường cong và khóa riêng tư là một số nguyên ngẫu nhiên d trong khoảng [1, n - 1], với 'n' là bậc của nhóm con cyclic được sinh bởi điểm G. Khóa công khai Q được tính bằng cách thực hiện phép nhân vô hướng khóa riêng tư d với điểm cơ sở G trên đường cong: $Q = d * G$ (nghĩa là cộng điểm G với chính nó d lần). Điểm cơ sở G là một điểm được chọn trước, cố định trên đường cong, dùng để sinh ra nhóm con được sử dụng cho các phép toán mật mã. Điểm cơ sở G, các tham số đường cong 'a' và 'b', cùng với các hằng số khác như đặc tả trường hữu hạn , bậc n của nhóm con sinh bởi G, và đồng yếu tố h cấu thành nên tham số miền của ECC. Tất cả các bên tham gia giao tiếp phải thống nhất sử dụng cùng một bộ tham số miền.

1.3 Bài toán Logarithm rời rạc trên đường cong elliptic

Bài toán Logarithm rời rạc trên Đường cong Elliptic (Elliptic Curve Discrete Logarithm Problem - ECDLP) là bài toán về việc tìm giá trị k thỏa mãn $P = k * G$, trong đó P là một điểm trên đường cong elliptic và G là điểm cơ sở. Độ khó của việc giải ECDLP xuất phát từ việc thiếu các thuật toán hiệu quả có thể giải quyết nó trong một khoảng thời gian hợp lý. Không giống như bài toán phân tích số nguyên (Integer Factorization Problem - IFP), vốn có thể bị tấn công bởi các thuật toán dưới hàm mũ như phương pháp tính chỉ số (index calculus method) hoặc sàng trường số chung (General number field sieve - GNFS), ECDLP hiện không có các thuật toán tấn công hiệu quả tương tự. Các thuật toán tốt nhất hiện được biết để giải ECDLP là các phương pháp chung (generic algorithms), chẳng hạn như các thuật toán có độ phức tạp $O(\sqrt{n})$ như Pollard's Rho hay Baby-step Giant-step, hoặc trong trường hợp xấu nhất là phương pháp vét cạn, bao gồm việc thử mọi giá trị k có thể cho đến khi tìm được giá trị thỏa mãn phương trình. Tuy nhiên, các cách tiếp cận này đều là bất khả thi về mặt tính toán đối với các đường cong elliptic với tham số đủ lớn, vì số lượng các giá trị có thể của k là cực kỳ lớn trong các ứng dụng thực tế. Vì thế, tính bảo mật của ECC phụ thuộc vào độ khó của bài toán Logarithm rời rạc trên đường cong Elliptic.

1.4 Trường hữu hạn

Thông thường trong toán học, các phép toán trên đường cong elliptic được định nghĩa trên trường số thực. Tuy nhiên, các phép toán được thực hiện trên máy tính thường không chính xác và chậm do lỗi làm tròn, trong khi các phép toán mã hóa đòi hỏi phải chính xác và nhanh chóng. Để làm cho các phép toán trên đường cong elliptic trở nên chính xác và hiệu quả hơn, mật mã đường cong elliptic thường được định nghĩa trên hai loại trường hữu hạn: trường nguyên tố F_p và trường nhị phân F_{2^m}.

Trường được chọn cần có một số lượng điểm hữu hạn đủ lớn, phù hợp cho các

hoạt động mật mã. Trong khóa luận này, tôi sẽ triển khai đường cong elliptic trên trường hữu hạn nguyên tố (F_p) cho hệ mật mã ElGamal trên đường cong elliptic. Do đó, các vấn đề được trình bày sau đây sẽ chỉ giới hạn trong phạm vi đường cong elliptic trên trường hữu hạn nguyên tố.

1.4.1 Đường cong Elliptic trên trường nguyên tố

Phương trình của đường cong elliptic dạng Weierstrass rút gọn trên một trường nguyên tố F_p với $p > 3$ là:

$$y^2 \pmod p = x^3 + ax + b \pmod p$$

trong đó a và b cũng là các phần tử của F_p và thỏa mãn điều kiện không kỳ dị:

$$4a^3 + 27b^2 \neq 0 \pmod p$$

Ở đây, p là một số nguyên tố. Các phần tử của trường hữu hạn F_p là các số nguyên trong tập hợp 0, 1, 2, ..., p - 1. Tất cả các phép toán như cộng, trừ, nhân, và chia (ngoại trừ chia cho 0) đều được thực hiện trên các số nguyên này và kết quả được lấy modulo p. Đây chính là số học mô-đun. Số nguyên tố p cần được chọn sao cho có một số lượng điểm đủ lớn hữu hạn trên đường cong elliptic để làm cho hệ mật mã trở nên an toàn. Tiêu chuẩn SEC (Standards for Efficient Cryptography) chỉ định các đường cong với p cần có độ dài nằm trong khoảng từ 160 đến 521 bit khi được sử dụng trong các ứng dụng thực tế [8].

1.4.2 Phép nhân vô hướng

Phép nhân điểm vô hướng là phép toán cốt lõi của tất cả các hệ mật mã đường cong elliptic. Đó là một phép toán có dạng $k * P$ với P là một điểm trên đường cong elliptic và k là một số nguyên dương. Về cơ bản, $k * P$ có nghĩa là cộng điểm P vào chính nó $k - 1$ lần, kết quả thu được là một điểm khác Q cũng nằm trên cùng đường cong elliptic đó.

Phép nhân điểm sử dụng hai phép toán đường cong elliptic cơ bản:

- Phép cộng điểm: Cộng hai điểm khác nhau $P_1 + P_2$.
- Phép nhân đôi điểm: Cộng điểm P với chính nó $P + P$

Ví dụ, để tính $k * P = Q$ với k là 23 thì $k * P = 23 * P$, ta có thể tính như sau:

$$k * P = 2 * (2 * (2 * (2 * P) + P) + P) + P$$

Phương pháp nhân vô hướng mô tả ở trên, sử dụng lặp đi lặp lại phép cộng điểm và nhân đôi điểm để tìm kết quả, được gọi là phương pháp "Nhân đôi và Cộng". Có nhiều phương pháp hiệu quả hơn khác cho phép nhân vô hướng, chẳng hạn như NAF (Non-Adjacent Form) và phương pháp Binary NAF cho nhân vô hướng.

1.4.3 Phép cộng

Cho P và Q là hai điểm phân biệt trên đường cong elliptic và $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$. Khi đó, ta có $R = P + Q$ với $R = (x_R, y_R)$ thì

$$x_R = \lambda^2 - x_P - x_Q \mod p$$

$$y_R = \lambda(x_P - x_R) - y_P \mod p$$

với $\lambda = (y_Q - y_P)/(x_Q - x_P) \mod p$

Nếu $Q = -P$ hay $Q = (x_P, -y_P \mod p)$ thì $P + Q = O$, O là điểm tại vô cùng.

Nếu $Q = P$ thì $P + Q = 2P$ và ta sử dụng phép nhân đôi để tính.

1.4.4 Phép nhân đôi

Cho P là một điểm thuộc đường cong elliptic, $P = (x_P, y_P)$.

Khi đó, nếu $R = 2P$ với $R = (x_R, y_R)$ thì

$$x_R = \lambda^2 - 2x_P \mod p$$

$$y_R = \lambda(x_P - x_R) - y_P \mod p$$

với $\lambda = (3x_P^2 + a)/(2y_P) \mod p$

Nếu $y_P = 0$ thì $2P = O$ với O là điểm tại vô cùng.

1.4.5 Phép trừ

Cho P và Q là hai điểm phân biệt trên đường cong elliptic và $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$. Khi đó, ta có $R = P - Q = P + (-Q)$ với $-Q = (x_Q, y_Q \mod p)$.

1.5 Thuật toán ElGamal trên đường cong elliptic

Hệ mã ElGamal, do Taher ElGamal phát triển vào năm 1985, là một thuật toán mã hóa mang tính ngẫu nhiên. Đặc tính này xuất phát từ việc sử dụng yếu tố ngẫu nhiên trong quá trình mã hóa. Chính yếu tố ngẫu nhiên này mang lại nhiều ưu điểm quan trọng cho ElGamal. Đáng chú ý, nó khiến cho ánh xạ mã hóa không phải là đơn ánh; tức là cùng một thông điệp có thể được mã hóa thành nhiều bản mã khác nhau. Do đó, các kiểu tấn công dựa trên phân tích thống kê bản mã trở nên kém hiệu quả hơn. Cơ chế hoạt động cơ bản của ElGamal như sau:
Giả sử Alice muốn gửi một tin nhắn m cho Bob.

1. Bob chọn một số nguyên tố lớn p, đường cong elliptic E, điểm sinh $G \in E(F_p)$ có bậc q và khóa bí mật $1 \leq n_b < q$.

2. Bob tính $Q_B = n_a * P$.

3. Bob công khai E, p, G, Q_B.

4. Alice ánh xạ m thành một điểm $M \in E(F_p)$ và chọn một số k ngẫu nhiên trong khoảng $[1, q-1]$.

5. Mã hóa: Alice tính $C_1 = k * G$, $C_2 = k * Q_B + M$

6. Alice gửi $C = (C_1, C_2)$ cho Bob.

7. Giải mã: Bob tính $C_2 - n_b * C_1 = M$.

Chương 2

Giao thức xác thực Baccouri

Giả sử Alice đóng vai trò là bên mã hóa và Bob là bên giải mã. Khi Alice muốn gửi một thông điệp mật đến Bob, cô ấy thực hiện phép nhân vô hướng trên đường cong elliptic sử dụng điểm cơ sở G và một số nguyên ngẫu nhiên k_A, đóng vai trò là khóa bí mật của Alice. Thông qua phép toán k_A * G, Alice thu được khóa công khai Q_A của mình và chia sẻ với mọi người. Bob cũng thực hiện các bước tương tự để tạo ra khóa công khai Q_B, sử dụng một số nguyên ngẫu nhiên k_B do anh ấy chọn và cùng điểm cơ sở G trên đường cong E. Các tham số của đường cong elliptic trên trường hữu hạn số nguyên tố F_p được công khai cho tất cả các bên tham gia. Ngoài ra, tất cả các bên phải thống nhất về các phần tử xác định đường cong elliptic. Khi đó, các tham số của lược đồ có thể được mô tả bằng một bộ sáu T = (p, a, b, G, n, h).

- p: Số nguyên tố đặc trưng cho trường hữu hạn F_p mà trên đó đường cong elliptic được định nghĩa. Nó quyết định số lượng phần tử của trường, vì vậy ảnh hưởng đến cấu trúc của nhóm các điểm trên đường cong.

- a và b: Các hệ số xác định phương trình đường cong elliptic, được lựa chọn dựa trên các yêu cầu bảo mật.

- G: Điểm sinh (hoặc điểm cơ sở), là một phần tử thuộc $E(F_p)$, có bậc n lớn nhất.

- n: là một số nguyên tố lớn, được gọi là bậc của điểm sinh G. Bậc n là số nguyên dương nhỏ nhất sao cho phép nhân vô hướng n lần điểm G (tức là G + G + ...

+ G, n lần) cho kết quả là điểm vô cực O, phần tử trung hòa của phép cộng trên đường cong.

- h: Hệ số phụ (cofactor), biểu diễn tỷ lệ giữa số lượng điểm trên đường cong elliptic E, ký hiệu là $\#E(F_p)$ và bậc n của điểm sinh G, được tính bằng công thức: $h = \frac{\#E(F_p)}{n}$

Trước khi mã hóa thông điệp bằng lược đồ ElGamal trên đường cong elliptic (ECEG), ta cần phải ánh xạ thông điệp đó thành một điểm trên đường cong. Phương pháp ánh xạ được sử dụng như sau:

$$E_i = (\alpha * i + \beta) * G$$

với

- G là điểm sinh của đường cong
- i là mã ASCII của kí tự
- α, β là hai số nguyên được biết bởi Alice và Bob

Mỗi ký tự ASCII được ánh xạ thành một điểm duy nhất trên đường cong elliptic. Quá trình ánh xạ này được tham số hóa bởi hai giá trị tạm thời α và β. Mục tiêu cốt lõi là tận dụng cấu trúc nhóm phi tuyến vốn có của đường cong elliptic E. Bằng cách sử dụng hàm ánh xạ phụ thuộc vào α và β, mối quan hệ trực tiếp và dễ đoán giữa giá trị số của ký tự ASCII và điểm tương ứng trên đường cong bị che khuất trong ngữ cảnh của phiên đó. Các tham số α và β được tạo ngẫu nhiên bởi bên gửi vào đầu mỗi phiên liên lạc mới. Khi Alice và Bob trao đổi các tham số biên mã tạm thời này, một bảng ánh xạ ASCII tạm thời được tạo cục bộ cho mỗi phiên. Để cả hai bên có thể sử dụng cùng một hàm ánh xạ, các tham số này cần được trao đổi một cách an toàn. Quá trình trao đổi này phải đảm bảo các yêu cầu bảo mật sau:

- Tính bảo mật (Confidentiality): Giá trị của α và β phải được bảo vệ khỏi việc bị lộ trong quá trình truyền.

- Tính toàn vẹn (Integrity) và Xác thực hai chiều (Mutual Authentication): Cả Alice và Bob cần xác minh rằng các tham số α, β nhận được không bị sửa đổi và xác thực định danh của đối phương.

Vì thế, ta sẽ mã hóa hai tham số biên mã α và β để đảm bảo tính bảo mật. Hơn nữa, chúng sẽ được sử dụng kết hợp với các hàm băm và phép toán XOR để đảm bảo tính xác thực và toàn vẹn.

2.1 Quá trình sinh α và β

Trong bước này, Alice sẽ sử dụng điểm sinh G và bậc n của đường cong đã chọn. Các bước của quá trình sinh như sau:

1. Chọn ngẫu nhiên một số nguyên s trong khoảng $[0, n-1]$.
2. Thực hiện phép nhân vô hướng: s * G = (x, y).
3. Alice kiểm tra x và y theo điều kiện sau:

$$((x * MaxASCIICode) + y) \in [0, n-1]$$

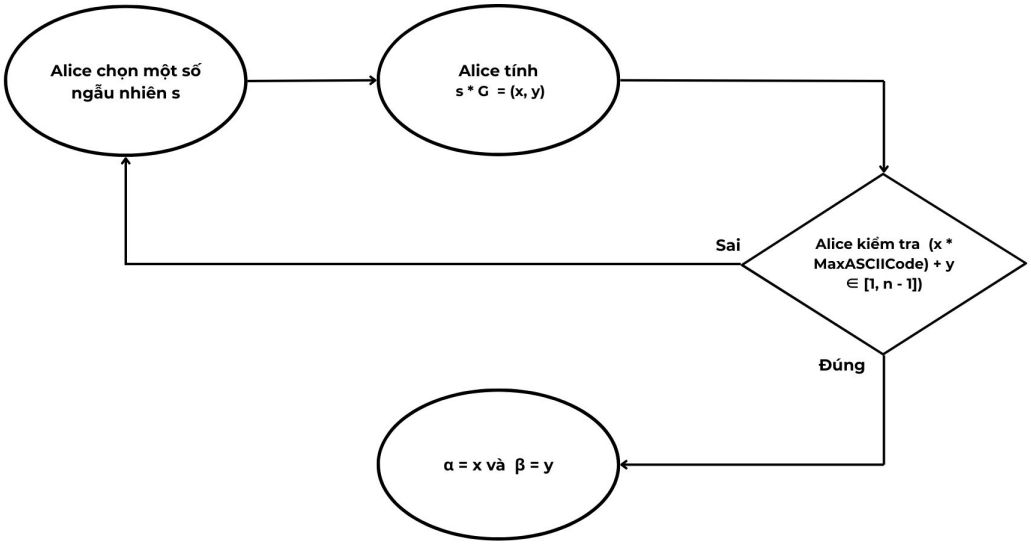
   với MaxASCIICode = 256.

4. Nếu x và y thỏa mãn điều kiện, thì α = x và β = y. Nếu không, quay lại bước 1 và lặp lại quá trình cho đến khi tìm được α và β thích hợp.

Điều kiện ở bước 3 được sử dụng để đảm bảo rằng, với mọi ký tự ASCII, khi thực hiện chuyển đổi, tất cả các kí tự đều được ánh xạ thành một điểm thuộc đường cong elliptic đã chọn trong quá trình biên mã.

2.2 Giao thức xác thực và trao đổi α, β

Để đảm bảo tính bí mật và tính toàn vẹn của α và β, Alice cần mã hóa chúng bằng lược đồ mật mã ElGamal trên đường cong elliptic. Ngoài ra, Alice và Bob phải



Hình 2.1: Quá trình sinh α và β

thực hiện xác thực hai chiều lẫn nhau, sử dụng các tham số mã hóa vừa tạo và đã được sử dụng trước đó, kết hợp với hàm băm và phép toán XOR.

Phía Alice:

1. Chọn một số nguyên ngẫu nhiên r, sau đó thực hiện tính $C_1 = r * G$ và $C_2 = r * Q_B + (\alpha, \beta)$ để nhận được $C = (C_1, C_2)$.

2. Thực hiện tính $h_1 = hash(C|(\alpha, \beta)lastsession)$, $(\alpha, \beta)lastsession$ là các tham số biên mã ở phiên làm việc trước.

3. Gửi C và h_1 cho Bob.

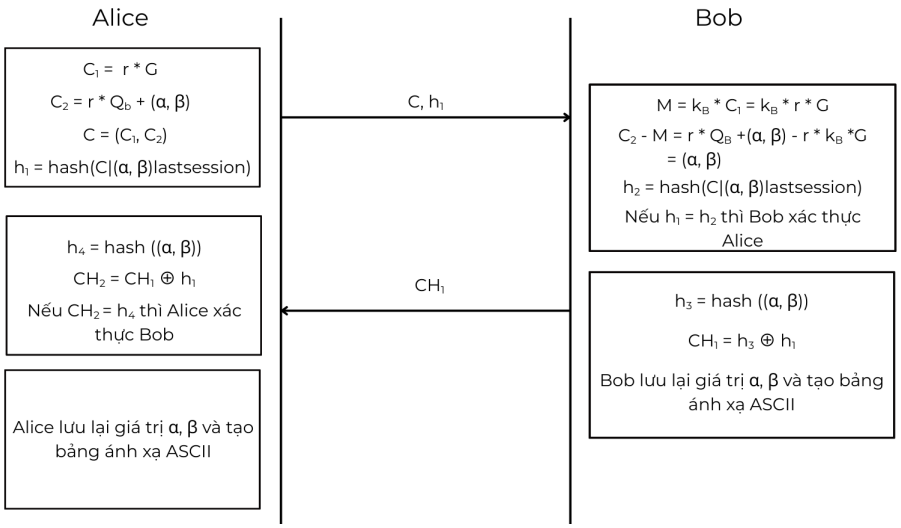
Phía Bob:

Khi Bob nhận được thông điệp

1. Bob giải mã C để thu được (α, β) bằng cách tính toán:

$$M = k_B * C_1 = r * k_b * G$$

$$C_2 - M = r * Q_B + (\alpha, \beta) - r * k_B * G = (\alpha, \beta)$$



Hình 2.2: Giao thức xác thực và trao đổi α, β

2. Bob tính $h_2 = hash(C|(\alpha, \beta)lastsession)$ và so sánh h_1 với h_2.

3. Nếu h_1 khớp với h_2, Bob xác thực Alice và xác nhận tính toàn vẹn của (α, β). Sau đó, Bob tạo bảng ánh xạ ASCII sử dụng α và β. Ngược lại, việc xác thực Alice thất bại và Bob sẽ từ chối phiên làm việc.

4. Bob tính:

$$h_3 = hash((\alpha, \beta))$$

rồi thực hiện phép XOR

$$h_3 \oplus h_1 = CH_1$$

5. Bob gửi CH_1 cho Alice.

Về phía Alice:

Khi Alice nhận được CH_1:

1. Thực hiện tính $h_4 = hash((\alpha, \beta))$.

2. Thực hiện XOR: $CH_2 = h_1 \oplus CH_1$. Nếu CH_2 khớp với h_4, Alice xác thực Bob và xác nhận rằng Bob đã nhận được (α, β) chính xác.

3. Alice dùng α và β để tạo bảng ánh xạ ASCII.

2.3 Cách thức triển khai ECEG cùng với giao thức xác thực đề xuất

Quá trình triển khai ECEG có thể được chia thành sáu giai đoạn:

a) Thiết lập hệ thống

Việc thiết lập một hệ mật mã đường cong Elliptic trên trường nguyên tố đòi hỏi các bên tham gia phải thống nhất về cùng một bộ tham số miền ECC:

1. Số nguyên tố p
2. Dạng phương trình của đường cong Elliptic E trên trường F_p
3. Các hệ số 'a' và 'b' của đường cong Elliptic
4. Điểm cơ sở G = (x_G, y_G) trên đường cong $E(F_p)$
5. Bậc (order) n của điểm cơ sở G
6. Các thuật toán thực hiện phép toán trên đường cong Elliptic: Hai bên cần dùng chung thuật toán cho các phép toán cơ bản như cộng điểm, nhân điểm (nhân vô hướng) trên đường cong trong trường F_p đã chọn

b) Tạo khóa

Alice và Bob thực hiện các bước sau để tạo cặp khóa riêng và công khai của mình:

- Chọn một số nguyên ngẫu nhiên k trong khoảng [1, n - 1].
- Tính điểm Q = k * G.

Số k được chọn chính là khóa bí mật, điểm Q vừa tính được là khóa công khai. Như vậy, cặp khóa riêng và khóa công khai của Alice là (k_A, Q_A) và của Bob là (k_B, Q_B)

c) Kiểm tra tính hợp lệ của khóa công khai

Khi nhận được khóa công khai Q (ví dụ, Alice nhận Q_B từ Bob, và ngược lại), các bên tham gia phải kiểm tra tính hợp lệ của khóa này. Việc này nhằm đảm bảo rằng Q là một điểm hợp lệ thuộc nhóm con có bậc n trên đường cong $E(F_p)$ (với n là bậc của điểm cơ sở G đã thống nhất). Việc kiểm tra bao gồm các bước sau:

- Kiểm tra điểm Q không phải là điểm vô cực (O).
- Kiểm tra các tọa độ x_Q, y_Q thuộc trường F_p hay không.
- Kiểm tra điểm Q nằm trên đường cong Elliptic hay không.
- Kiểm tra n * Q = O, O là điểm vô cực.

d) Thực hiện tạo, trao đổi hai tham số tạm thời, xác thực hai chiều và tạo bảng ánh xạ ASCII

Đầu tiên, Alice chọn ngẫu nhiên một số nguyên s nằm trong khoảng [1, n - 1] và tính toán điểm tạm thời bằng phép nhân vô hướng. Sau đó, Alice kiểm tra xem điểm thu được này có thỏa mãn điều kiện không. Nếu điểm đó hợp lệ, Alice thực hiện trao đổi và xác thực hai chiều với Bob, sau đó cả hai bên tạo bảng ánh xạ ASCII như đã trình bày ở 3.1 và 3.2.

e) Mã hóa

Để trao đổi tin nhắn một cách an toàn, Alice thực hiện mã hóa theo các bước sau:

- Sử dụng bảng ánh xạ ASCII để chuyển ký tự thành một điểm Pm.
- Chọn số ngẫu nhiên r thuộc khoảng [1, n - 1].
- Tính $E_1 = r * G$ với G là điểm cơ sở.
- Tính $E_2 = r * Q_B + Pm$.

$E = (E_1, E_2)$ là cặp điểm mã hóa của điểm Pm và được gửi cho Bob.

f) Giải mã

- Bob thực hiện tính $M = k_B * E_1$ với k_B là khóa bí mật của Bob.
- Tính $E_2 - M = E_2 - (k_B * E_1) = (r * Q_B + Pm) - (k_B * r * G) = (r * k_B * G + Pm) - (k_B * r * G) = Pm$ để thu được Pm.
- Sử dụng bảng ánh xạ ASCII để chuyển Pm về văn bản có thể đọc được.

Chương 3

Phân tích bảo mật, thiết kế thực nghiệm

3.1 Phân tích bảo mật

3.1.1 Đảm bảo tính bảo mật và tính toàn vẹn của tham số biên mã

Tính bảo mật

Giao thức đảm bảo tính bảo mật cho các tham số (α, β) bằng cách mã hóa chúng sử dụng ECEG. Cụ thể, bên gửi Alice thực hiện các phép tính sau:

- Chọn một số ngẫu nhiên r.
- Tính toán thành phần thứ nhất của bản mã: $C_1 = r * G$, trong đó G là điểm cơ sở của đường cong elliptic.
- Tính toán thành phần thứ hai của bản mã: $C_2 = M + r * Q_B$, trong đó M biểu diễn điểm trên đường cong elliptic tương ứng với (α, β) và Q_B là khóa công khai ECEG của bên nhận Bob.
- Bản mã hoàn chỉnh là $C = (C_1, C_2)$.

Để một đối thủ có thể phục hồi (α, β) từ C, đối thủ cần tính được giá trị $r * Q_B$. Với $C_1 = r * G$ và $Q_B = k_B * G$ (trong đó k_B là khóa bí mật của Bob) là các giá trị công khai hoặc có thể suy ra, việc tính toán $r * Q_B = r * k_B * G$ từ $r * G$ và $k_B * G$

tương đương với việc giải quyết bài toán Elliptic Curve Diffie-Hellman. Hơn nữa, việc xác định r từ $C_1 = r * G$ khi biết G là bài toán Logarithm rời rạc trên đường cong Elliptic (ECDLP). Dựa trên giả định về độ khó tính toán của các bài toán ECDLP [6] và ECDHP [3], việc đối thủ tính toán $r * Q_B$ từ các thông tin công khai là bất khả thi về mặt tính toán. Do đó, đối thủ không thể giải mã C_2 để thu được (α, β). Vì thế, tính bảo mật của (α, β) được đảm bảo.

Tính toàn vẹn

Tính toàn vẹn của bản mã C và nguồn gốc của nó được xác thực bằng cách sử dụng hàm băm. Alice tính toán một giá trị băm $h_1 = hash(C|(\alpha, \beta)lastsession)$, trong đó $(\alpha, \beta)lastsession$ là các tham số từ phiên giao tiếp trước đó. Giá trị h_1 được gửi kèm theo C. Khi nhận được C và h_1, Bob thực hiện các bước sau:

- Sử dụng khóa bí mật k_B của mình để giải mã C và thu được các tham số (α', β').
- Tính toán giá trị băm: $h_2 = hash(C|(\alpha, \beta)lastsession))$, sử dụng giá trị $(\alpha, \beta)lastsession$ đã được lưu trữ trước đó.
- So sánh h_1 và h_2. Nếu $h_1 = h_2$, Bob có thể kết luận với độ tin cậy cao rằng: C không bị sửa đổi trong quá trình truyền và C được tạo bởi một bên sở hữu kiến thức về $(\alpha, \beta)lastsession$.

Do (α', β') được giải mã trực tiếp từ C, việc xác minh tính toàn vẹn của C cũng sẽ đảm bảo rằng các tham số (α', β') thu được là kết quả giải mã của bản mã gốc chưa bị thay đổi. Nếu C bị thay đổi trong quá trình truyền phát, giá trị giải mã (α', β') cũng sẽ bị thay đổi và khi đó giá trị băm h_2 sẽ không khớp với h_1. Vì thế, giao thức đã đảm bảo tính toàn vẹn dữ liệu cho bản mã C và qua đó, đảm bảo tính toàn vẹn cho cặp tham số được giải mã từ C.

3.1.2 Xác thực lẫn nhau sử dụng tham số biên mã, hàm băm và phép toán XOR

Bên cạnh việc đảm bảo tính bảo mật và toàn vẹn, các tham số biên mã (α, β), hàm băm và phép toán XOR được tận dụng để thực hiện xác thực lẫn nhau giữa hai bên giao tiếp.

Xác thực Alice bởi Bob

Quá trình xác thực Alice bởi Bob được tích hợp trực tiếp vào bước kiểm tra tính toàn vẹn. Khi Bob nhận C và h_1, Bob thực hiện giải mã và tính toán $h_2 = hash(C|(\alpha, \beta)lastsession)$. Việc xác minh thành công đẳng thức $h_1 = h_2$ không chỉ đảm bảo tính toàn vẹn mà còn xác thực Alice bởi vì chỉ có thực thể sở hữu $(\alpha, \beta)lastsession)$ hợp lệ mới có thể tính toán đúng giá trị h_1 tương ứng với bản mã C đã tạo. Do đó, sự trùng khớp này chứng tỏ thông điệp bắt nguồn từ Alice. Vì thế, Bob đã xác thực Alice dựa trên khả năng của Alice trong việc cung cấp bằng chứng về kiến thức của $(\alpha, \beta)lastsession)$ thông qua giá trị băm h_1 đi kèm với bản mã C.

Xác thực Bob bởi Alice

Sau khi Bob đã xác thực Alice và giải mã thành công C để thu được (α, β), Bob tiến hành chứng minh danh tính của mình với Alice.

- Bob tính toán giá trị băm của các tham số vừa giải mã: $h_3 = hash((\alpha, \beta))$.
- Bob thực hiện phép toán XOR giữa h_3 và h_1: $CH_1 = h_3 \oplus h_1$. Bob gửi CH_1 cho Alice.

Alice, khi nhận được CH_1, thực hiện các bước sau:

- Alice đã biết h_1 và (α, β).
- Alice tính toán lại giá trị băm của tham số gốc: $h_4 = hash((\alpha, \beta))$.
- Alice phục hồi giá trị h_3 mà Bob đã tính bằng cách thực hiện phép XOR: $CH_2 = h_1 \oplus CH_1$.

- Alice so sánh giá trị CH_2 (là h_3 được phục hồi) với h_4 (giá trị băm cô tự tính).

Nếu CH_2 bằng với h_4, điều này chứng tỏ h_3 bằng với h_4. Sự trùng khớp này xác nhận rằng Bob đã sở hữu đúng khóa bí mật k_B. Bởi lẽ, chỉ khi giải mã thành công bản mã C và thu được (α, β) chính xác, Bob mới có thể tạo ra h_3 bằng với h_4. Do đó, việc Bob gửi CH_1 cho phép Alice khôi phục và xác minh h_3 đã xác thực Bob. Cơ chế xác thực này dựa trên sự kết hợp của các tham số (α, β), hàm băm và phép toán XOR.

3.2 Thực nghiệm

3.2.1 Mô phỏng giao thức

Ở phần này, tôi sẽ lập trình đầy đủ các bước của giao thức và tiến hành chạy với ví dụ cụ thể để khẳng định giao thức hoạt động đúng như mô tả theo lý thuyết sử dụng SageMath trên một hệ thống có cấu hình: CPU Intel(R) Core(TM) i7-12700H 2.30 GHz, RAM 16.00GB, chạy trên nền tảng Ubuntu.

Thiết lập hệ thống

Đầu tiên, tôi sẽ tiến hành thêm các thư viện cần thiết, lựa chọn các tham số của đường cong elliptic, điểm cơ sở g, p và n. Ở mô phỏng này, tôi sẽ chọn đường cong có các tham số là:

- p = 100000000000000000039
- a = 0
- b = 7
- g = (69496283786894253888, 58916120834459954240)
- n = 50000000008969935366

Đoạn code thực hiện:

```
import hashlib
from typing import Union
import random
import string
import time

p = 10000000000000000039
a = 0
b = 7

E = EllipticCurve(GF(p),[a, b])
G = E.gens()
g = E.point(G[0])
n = g.order();
```

Tạo khóa công khai/bí mật cho Alice và Bob

Tiếp theo, tôi tiến hành tạo cặp khóa công khai và bí mật cho Alice và Bob.

```
kA = randint(1, n)
QA = kA * g

kB = randint(1, n)
QB = kB * g
```

Ở lần chạy này, các giá trị nhận được là:

- $k_A = 17234869679141697959$
- $Q_A = (24148284530724724990 : 89237694802055227180 : 1)$
- $k_B = 32426097915697275469$
- $Q_B = (67119382834134792199 : 71328156358363944895 : 1)$

k_A, k_B là khóa bí mật và Q_A, Q_B là khóa công khai của Alice và Bob.

Quá trình sinh α và β

Kế tiếp, Alice tạo cặp tham số biên mã (α, β) phù hợp với điều kiện đã trình bày ở trên.

```
def genEphemeralParameter(n, g):
    s = randint(1, n)
    while True:
        parameter = s * g
        s = randint(1, n)
```

```
        if 0 <= int(parameter[0]) * 256 + int(parameter[1]) < n:
            return parameter

ephemeral = genEphemeralParameter(n, g)
```

Ở lần chạy này, cặp tham số nhận được là (125929319290300242, 7491963579492646982).

Giao thức xác thực và trao đổi α, β

Alice thực hiện:

```
r = randint(1, n)
C1 = r * g
C2 = r * QB + ephemeral
C = (C1, C2)
h1 = calculate_sha256(f"{C}|{ephemeralLastSession}")
```

với ephemeralLastSession là cặp tham số được sử dụng ở phiên trước. Ở lần chạy này, các giá trị nhận được là:

- $C_1 = (5405829232180582341 : 72620887608343965895 : 1)$
- $C_2 = (83191304044977708726 : 9188903516077432223 : 1)$
- $h_1 = 1bad007a3c02584516caa7d93a89bd0393147876500925e45810fa53f17226a7$

và gửi C, h_1 cho Bob.

Bob thực hiện:

```
M = kB * C[0]
parameterB = C[1] - M
h2 = calculate_sha256(f"{C}{ephemeralLastSession}")
if h1 == h2:
    tableB = create_ASCII_map(parameterB, g)
    h3 = calculate_sha256(f"{parameterB}")
    CH1 = xorSha256Digests(h1, h3)
```

Ở lần chạy này, các giá trị nhận được là:

- $M = (20167080408120019994 : 73521288540533939090 : 1)$
- $parameterB = (125929319290300242 : 7491963579492646982 : 1)$
- $h_2 = 1bad007a3c02584516caa7d93a89bd0393147876500925e45810fa53f17226a7$

$parameterB$ chính là cặp tham số mà Alice gửi cho Bob. $h_1 = h_2$ nên Bob xác thực Alice, tiếp tục tính được các giá trị sau:

- $h_3 = bc3a9368484267de0881392b4535cbb7f31246ba28010caa82f75696f1ececd1$
- $CH_1 = a797931274403f9b1e4b9ef27fbc76b460063ecc7808294edae7acc5009eca76$
- tableB là bảng ánh xạ ASCII được Bob tạo từ cặp tham số nhận được.

và gửi CH_1 cho Alice.

Alice thực hiện:

```
h4 = calculate_sha256(f"{ephemeral}")
CH2 = xor_sha256_digests(h1, CH1)
if CH2 == h4:
    tableA = create_ASCII_map(ephemeral, g)
```

Ở lần chạy này, các giá trị nhận được là:

- $h_4 = bc3a9368484267de0881392b4535cbb7f31246ba28010caa82f75696f1ececd1$
- $CH_2 = bc3a9368484267de0881392b4535cbb7f31246ba28010caa82f75696f1ececd1$

Vì $h_4 = CH_2$ nên Alice xác thực Bob và xác nhận rằng Bob đã nhận được (α, β) chính xác. Alice tiến hành tạo bảng ánh xạ ASCII tableA từ cặp tham số của cô.

Vậy giao thức đã hoạt động đúng như mô tả lý thuyết, xác thực và trao đổi tham số biên mã (α, β) hiệu quả.

3.2.2 Phân tích an toàn thực nghiệm

Em sẽ tiến hành mô phỏng một số kịch bản tấn công đơn giản để cho thấy giao thức có thể chống lại chúng như thế nào, dựa trên các cơ chế đã được thiết kế.

Tấn công phát lại (Replay Attack) đơn giản

Kịch bản tấn công: Kẻ tấn công Eve thu thập một cặp (C, h_1) hợp lệ từ một phiên trao đổi trước đó giữa Alice và Bob. Ở một phiên làm việc sau, Eve cố gắng gửi lại cặp (C, h_1) này cho Bob.

Phân tích khả năng phòng vệ: Tham số $(\alpha, \beta)lastsession$ được cập nhật sau mỗi phiên trao đổi thành công. Giả sử Bob đã cập nhật $(\alpha, \beta)lastsession$ của mình thành giá trị tương ứng với phiên hiện tại. Khi Bob nhận được cặp (C, h_1) cũ, Bob sẽ tính $h_2^{replay} = hash(C|(\alpha, \beta)lastsession\_current)$. Do $(\alpha, \beta)lastsession\_current$ khác với $(\alpha, \beta)lastsession\_old$ (tham số được sử dụng để tạo h_1 ban đầu), giá trị h_2^{replay} sẽ không khớp với h_1 nhận được. Kết quả là Bob sẽ không xác thực và từ chối thông điệp.

Minh họa thực nghiệm: Để minh họa, giả sử C, h_1 có giá trị như đã tính ở phần 3.3.1. Bob đã chuyển sang một phiên mới và chuyển cặp tham số $(\alpha, \beta) = (29293896899421307565, 47451308829324521335)$ đã nhận được từ Alice ở phần 3.3.1 thành $(\alpha, \beta)lastsession$ mới. Bob tính

$$\begin{aligned} h_2^{replay} &= hash(C|(\alpha, \beta)lastsession\_current) \\ &= 3af7e25e7557dbba91cfbfcc81cdd24f9019790066d610cd90224b4db12b7c5a \end{aligned}$$

Rõ ràng $h_2^{replay} \neq h_1$, do đó Bob phát hiện sự thay đổi.

Mô phỏng kẻ tấn công không có khóa bí mật của Bob cố gắng giả mạo Bob

Kịch bản tấn công: Eve chặn cặp (C, h_1) do Alice gửi. Eve không sở hữu khóa bí mật k_B của Bob, do đó không thể giải mã C để thu được cặp (α, β) chính xác mà Alice đã mã hóa. Để giả mạo Bob, Eve tạo ra một thông điệp CH_1' gửi lại cho Alice. Giả sử Eve tạo ra một cặp (α', β') ngẫu nhiên hoặc dựa trên một phỏng đoán nào đó và tính $h_3' = hash((\alpha', \beta'))$. Sau đó, Eve tính $CH_1' = h_3' \oplus h_1$ (trong đó h_1 là giá trị Eve chặn được) và gửi CH_1' cho Alice.

Phân tích khả năng phòng vệ: Khi Alice nhận được $CH1'$, cô ấy sẽ thực hiện phép toán $CH_2' = h_1 \oplus CH_1'$ để khôi phục giá trị h_3 mà Bob (theo giả định) đã gửi. Trong trường hợp này, $CH_2' = h_1 \oplus (h_3' \oplus h_1) = h_3'$. Đồng thời, Alice tự tính toán giá trị băm $h_4 = hash((\alpha, \beta))$, với (α, β) là cặp tham số gốc mà Alice đã khởi tạo và mã hóa trong C. Do Eve không có (α, β) chính xác, nên $(\alpha', \beta') \neq (\alpha, \beta)$, dẫn đến $h_3' \neq h_4$. Khi Alice so sánh $CH_2' = h_3'$ với h_4, cô ấy sẽ phát hiện sự không khớp. Điều này cho

thấy thông điệp không xuất phát từ Bob hợp lệ. Alice sẽ từ chối xác thực.

Minh họa thực nghiệm: Để minh họa, giả sử C, h_1, h_4 có giá trị như đã tính ở phần 3.3.1. Eve tạo ra một cặp $(\alpha', \beta') = (10933699806426560872, 75750767755776949727)$ và tính:

- $h'_3 = 42234c900c2b650b9504621e4fa3d85206ba49c3d93c2d46751f573589a5ad4d$
- $CH'_1 = 598e4cea30293d4e83cec5c7752a655195ae31b5893508a22d0fad6678d78bea$

Alice tính:

$$CH'_2 = 42234c900c2b650b9504621e4fa3d85206ba49c3d93c2d46751f573589a5ad4d$$

Rõ ràng $CH'_2 \neq h_4$, do đó Alice phát hiện sự thay đổi.

Kết luận

Trong khuôn khổ khóa luận này, tôi đã trình bày phương pháp được đề xuất bởi Baccouri và cộng sự để giải quyết một vấn đề thực tiễn trong việc tăng cường bảo mật cho hệ mật mã ElGamal trên đường cong elliptic (ECEG). Cụ thể, bài toán đặt ra là đảm bảo tính bí mật, toàn vẹn và xác thực lẫn nhau khi trao đổi các tham số biên mã tạm thời (α, β).

Để giải quyết vấn đề này, Baccouri và cộng sự đã đề xuất và xây dựng phương pháp giao thức xác thực mới dựa trên thuật toán ElGamal trên đường cong elliptic. Phương pháp này không chỉ kế thừa tính hiệu quả của ECEG mà còn tích hợp thêm một lớp xác thực hai chiều. Bằng cách sử dụng chính cơ chế mã hóa ECEG để bảo vệ (α, β), kết hợp với hàm băm mật mã và phép toán XOR trên các tham số này, giao thức cho phép hai bên liên lạc vừa trao đổi an toàn các tham số cần thiết, vừa xác thực được danh tính của nhau trước khi bắt đầu phiên mã hóa dữ liệu chính. Trong khuôn khổ của khóa luận, tôi đã:

- Trình bày về cơ sở lý thuyết: các khái niệm nền tảng của số học mô-đun, mật mã đường cong elliptic, thuật toán ElGamal trên đường cong elliptic.
- Phân tích chi tiết giao thức: Làm rõ các bước trong quá trình sinh (α, β), cơ chế trao đổi và xác thực hai chiều mà Baccouri và cộng sự đề xuất.
- Phân tích bảo mật và thực nghiệm: Tiến hành phân tích bảo mật và hiện thực hóa giao thức thông qua mô phỏng trên môi trường SageMath. Quá trình mô phỏng đã bao gồm việc thiết lập hệ thống, tạo khóa, sinh và trao đổi (α, β), cũng

như các bước xác thực giữa Alice và Bob. Kết quả mô phỏng đã minh chứng giao thức hoạt động đúng đắn, tuân thủ chặt chẽ các bước được mô tả trên lý thuyết.

- Phân tích an toàn thực nghiệm: Mô phỏng một số kịch bản tấn công cơ bản như Tấn công Phát lại (Replay Attack) và Tấn công Giả mạo Bob. Kết quả phân tích thực nghiệm cho thấy các cơ chế bảo vệ của giao thức có khả năng phát hiện và chống lại các hình thức tấn công này, qua đó khẳng định tính hiệu quả của các biện pháp an toàn được tích hợp

Qua quá trình nghiên cứu, tôi thấy rằng giao thức của Baccouri và cộng sự là một đề xuất thú vị, kết hợp các nguyên lý mật mã đã được kiểm chứng để giải quyết vấn đề bảo mật trong trao đổi tham số. Việc tích hợp xác thực hai chiều vào quá trình trao đổi tham số biên mã là một điểm mạnh, giúp tăng cường đáng kể độ tin cậy của phiên giao tiếp so với việc chỉ đơn thuần mã hóa các tham số này. Quá trình thực hiện khóa luận này giúp tôi củng cố kiến thức chuyên ngành về mật mã học đường cong elliptic và thiết kế giao thức an toàn, mở ra những nhận thức về các thách thức và giải pháp trong việc bảo vệ thông tin trong thế giới số. Tôi mong rằng những kết quả tìm hiểu được sẽ là tài liệu tham khảo hữu ích cho những ai quan tâm đến lĩnh vực này.

Tài liệu tham khảo

[1] Baccouri S., H. Fatrhat, T. Azzabi and P. R. Attia (2023), "Lightweight authentication scheme based on Elliptic Curve El Gamal", *Journal of Information and Telecommunication*, 8(2), pp. 231–261.

[2] Baccouri S., H. Fatrhat, N. Gharbi, R. Tahar, T. Azzabi and P. R. Attia (2022), "Ephemeral Encoding Message in the Elliptic Curve Cryptography for IoT", *2022 5th International Conference on Advanced Systems and Emergent Technologies*, pp. 172-177.

[3] Haakegaard R. and J. Lang (2015), The Elliptic Curve Diffie-Hellman (ECDH), pp. 1-4.

[4] Koblitz N. (1987), “Elliptic Curve Cryptosystems,” *Mathematics of Computation*, 48, pp. 203-209.

[5] Mahto D. and D. K. Yadav (2017), RSA and ECC: A comparative analysis, *International Journal of Applied Engineering Research*, 12(19), pp. 9053–9061.

[6] Menezes A. (2001), "Evaluation of security level of cryptography: The elliptic curve discrete logarithm problem (ECDLP)", *University of Waterloo*, 14, pp. 1-24.

[7] Miller V. S. (1985), "Use of Elliptic Curves in Cryptography", *Advances in Cryptology — CRYPTO ’85 Proceedings*, pp. 417-426.

[8] Standards for Efficient Cryptography Group (SECG) (2009), "SEC 1: Elliptic Curve Cryptography (Version 2.0)".