ĐẠI NAM
UNIVERSITY

Bài 8

THUẬT TOÁN MÃ HÓA ELGAMAL

Giảng viên: TS.
(namtq@dainam.edu.vn)

1. Giới thiệu chung và ứng dụng Elgamal

2. Mô tả thuật toán Elgamal

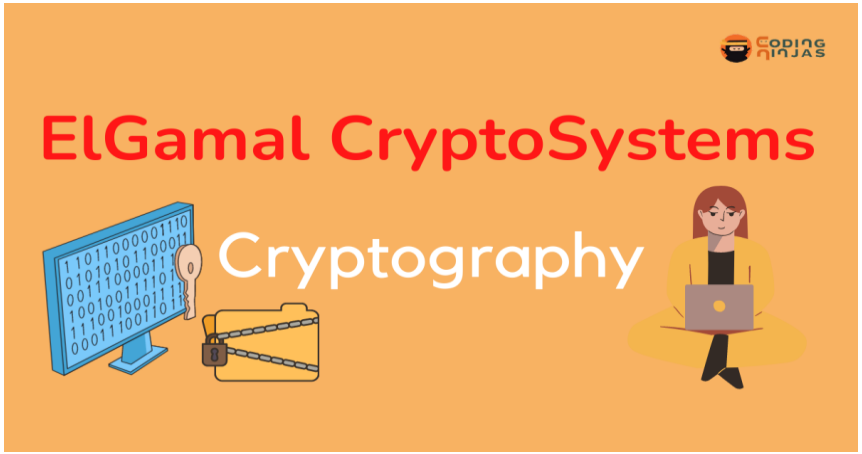
3. Ví dụ minh họa với các con số cụ thể

4. Luyện tập Lab 8 với lập trình

- Hệ mã hóa với khoá công khai ElGamal được đề xuất năm 1985, dựa vào độ phức tạp của bài toán lôgarit rời rạc.
- Thuật toán mã hóa ElGamal đảm bảo tính bảo mật và được sử dụng rộng rãi trong mật mã hiện đại.

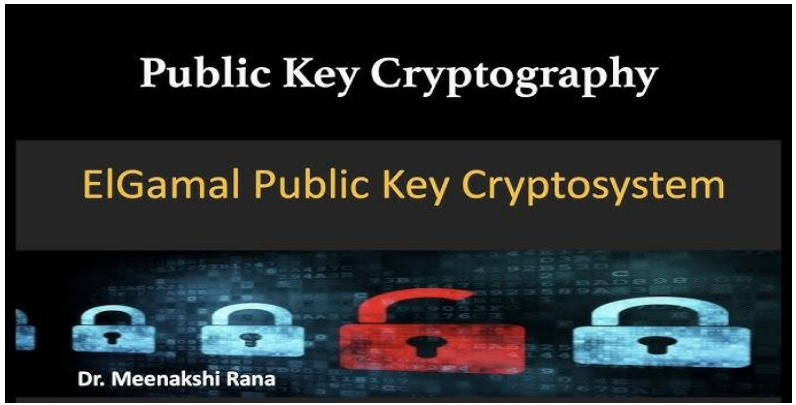
- Dùng ChatGPT hoặc Google, DeepSeek,... hỏi: "Mã hóa Elgamal hoạt động như thế nào" ➔ Em có thể tóm tắt mã hóa Elgamal ?
- "Mã hóa Elgamal dùng thực tế ở đâu" ➔ Em giải thích vai trò ý nghĩa của Elgamal trong thực tế ?
- Đặc điểm nổi bật của mã hóa Elgamal là gì?
- Tại sao Elgamal được dùng phổ biến trong thực tế?
- Elgamal khác RSA ở đặc điểm nào?

- Mã ElGamal được dùng trong số tiêu chuẩn như: Digital Signature Standard (DSS) và S/MIME e-mail standard
- An ninh của ElGamal dựa trên độ khó của việc tính logarit rời rạc



- ElGamal là thuật toán quan trọng trong mật mã công khai, được sử dụng rộng rãi trong chữ ký số và các hệ thống bảo mật như PGP (Pretty Good Privacy)

- ElGamal là ElGamal là một thuật toán mạnh mẽ, được ứng dụng rộng rãi trong nhiều lĩnh vực từ bảo mật thông tin đến tiền mã hóa và xác thực danh tính.



- Chữ ký số (ElGamal Signature): ElGamal được sử dụng trong chữ ký số, giúp xác minh tính xác thực và toàn vẹn của dữ liệu.
  - Được sử dụng trong các hệ thống xác thực tài liệu điện tử, hợp đồng số, giao dịch ngân hàng.
  - Cơ chế chữ ký số ElGamal là nền tảng cho nhiều thuật toán khác như DSA (Digital Signature Algorithm).

E-MAIL APPLICATION : In this project, we used ElGamal which is one of the encryption algorithms designed for distributed systems and used to sign e-mails, to encrypt all of the message. Further when the program is used on any local operating system to provide high-level security in login processes and to prevent from accessing of unauthorized users to the program, we used a hash algorithm during saving password to the database.

- Mã hóa thông tin và bảo mật dữ liệu:
  - ElGamal giúp mã hóa tin nhắn, email, file để bảo vệ thông tin khỏi bị đánh cắp hoặc chỉnh sửa.
  - Được sử dụng trong các hệ thống bảo mật dữ liệu, đặc biệt là trong truyền thông an toàn.
  - Là một lựa chọn thay thế cho RSA trong một số trường hợp yêu cầu bảo mật cao hơn.

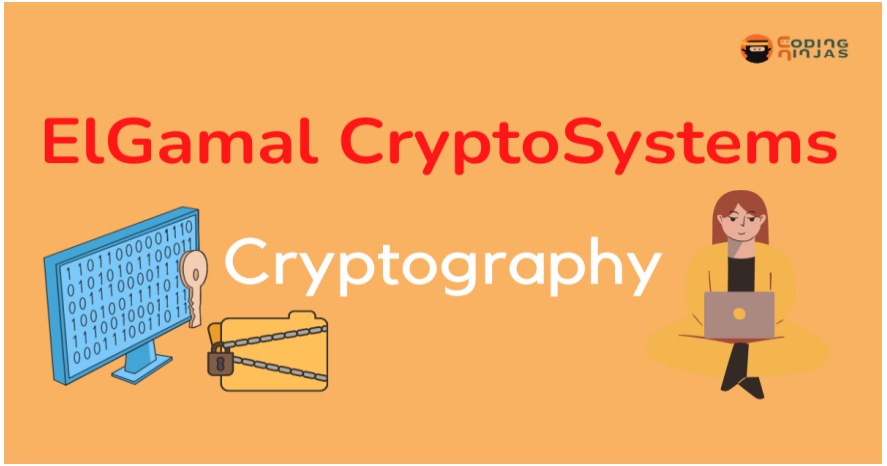
- Hệ thống bảo mật trong Blockchain & Tiền mã hóa
  - Một số hệ thống tiền mã hóa và blockchain sử dụng ElGamal hoặc các biến thể của nó để bảo vệ giao dịch.
  - Ví dụ: Một số giao thức Zero-Knowledge Proof (Bằng chứng không tiết lộ) sử dụng ElGamal để xác minh danh tính mà không tiết lộ thông tin thực tế.

- Hệ thống bỏ phiếu điện tử (E-Voting)
  - ElGamal được dùng trong hệ thống bỏ phiếu điện tử để đảm bảo tính bí mật và xác thực của phiếu bầu.
  - Giúp người tham gia bỏ phiếu ẩn danh mà vẫn có thể kiểm tra kết quả một cách minh bạch.

- Xác thực và quản lý danh tính
  - Được dùng trong hệ thống xác thực người dùng như đăng nhập an toàn, quản lý danh tính số.
  - Ứng dụng trong các giao thức bảo mật như SSL/TLS, SSH để mã hóa và bảo vệ kết nối mạng.

Logarit rời rạc là gì?

- Dùng ChatGPT hay Google hỏi và chúng ta cùng thảo luận: "Tại sao nói an ninh của ElGamal dựa trên độ khó của việc tính logarit rời rạc?"



Căn nguyên thủy/ phần tử sinh

Giả sử p là một **số nguyên tố**. Một số nguyên g là **căn nguyên thủy modulo p** nếu:

$g^k \bmod p$ (với k = 1, 2, ..., p-1) tạo ra **toàn bộ** các số từ 1 đến p-1 không trùng nhau.

Với p = 7, ta xét các số từ 1 đến 6 để tìm căn nguyên thủy (còn gọi là phần tử sinh):

Thử g = 3:

$$3^1 \equiv 3 \mod 7$$
$$3^2 \equiv 2 \mod 7$$
$$3^3 \equiv 6 \mod 7$$
$$3^4 \equiv 4 \mod 7$$
$$3^5 \equiv 5 \mod 7$$
$$3^6 \equiv 1 \mod 7$$

Tập kết quả: {3, 2, 6, 4, 5, 1} là đủ cả 1 đến 6

⇒ **3 là căn nguyên thủy modulo 7.**

Logarit rời rạc là gì?

Trong toán học, **logarit thông thường** có dạng:

$$b = a^x \Rightarrow x = \log_a b$$

Ví dụ: $2^3 = 8$ thì $\log_2 8 = 3$.

Nhưng trong số học modulo, logarit không dễ tính như vậy.

Định nghĩa logarit rời rạc

Cho một số nguyên tố p, một số nguyên g (được gọi là **căn nguyên thủy** của p), và một số y, bài toán logarit rời rạc là tìm x sao cho:

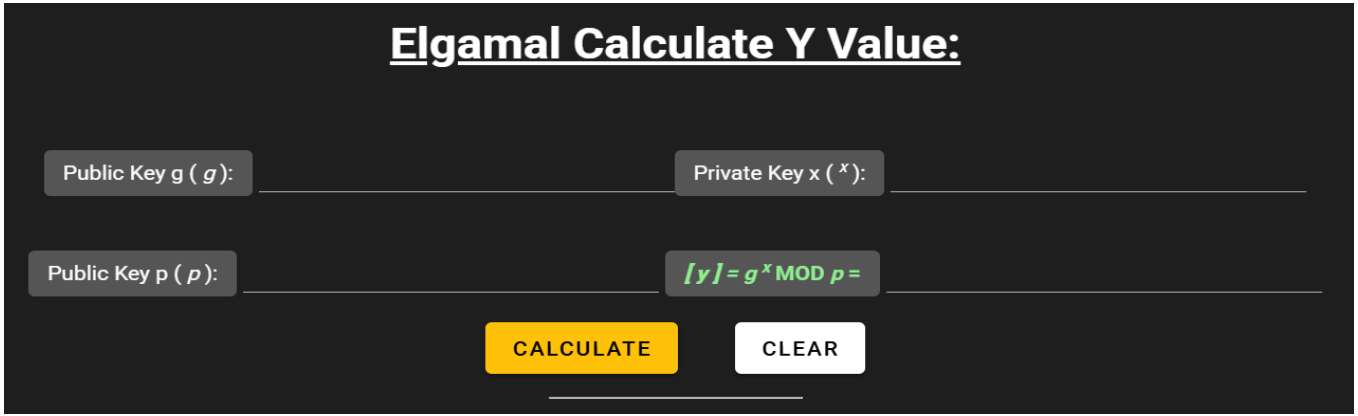
$$g^x \equiv y \pmod{p}$$

Ví dụ: Với $p = 23$, $g = 5$, nếu biết $5^x \mod 23 = 8$, thì tìm x là rất khó (ở đây $x = 6$).

Không có công thức đơn giản nào để giải bài toán này một cách nhanh chóng, đặc biệt khi p rất lớn.

- Website Elgamal online:

  https://cryptocalc.com.au/elgamal-crypto-calc/



Người sử dụng tạo cặp khóa công khai/cá nhân:

Chọn số nguyên tố lớn p, và cơ số α là phần tử sinh (primitive root) theo modulo p *(tức là với k = [1..p-1], α^k mod p sẽ cho các giá trị [1..p-1]).* Các giá trị này sẽ được công bố công khai.

Người dùng A chọn ngẫu nhiên số nguyên X_A với $0 < X_A < p-1$ Tính:

$$Y_A = \alpha^{X_A} \bmod p$$

Khoá cá nhân và khoá công khai của A là $\mathbf{X_A}$ và $\{\mathbf{p, \alpha, Y_A}\}$

Giả sử B có khóa công khai của A và B có thể mã hoá thông điệp để gửi cho A như sau:

- Giả sử thông điệp M là một số nguyên: $0 \leq M \leq p - 1$
- B chọn số nguyên ngẫu nhiên k với $1 \leq k \leq p - 1$
- B tính khoá một lần: $K = (Y_A)^k \bmod p$
- Sau đó B tạo bản mã gồm cặp hai giá trị $\{C_1, C_2\}$ với $C_1 = \alpha^k \bmod p$; $C_2 = (K \times M) \bmod p$

Vậy B đã mã hóa bản nguồn M thành bản mã {C1, C2} và gửi bản mã cho A

Số k chỉ dùng đúng một lần để tính {C1, C2}.

A giải mã {C1, C2} bằng khóa cá nhân của mình X_A như sau:

A tính:

$$K = C_1^{X_A} \bmod p$$

$$\left(C_1^{X_A} \bmod p = \alpha^{k.X_A} \bmod p = Y_A^{\,k} \bmod p = K \right)$$

A khôi phục lại bản nguồn $M = (C_2 \times K^{-1}) \bmod p$

An toàn của thuật toán ElGamal phụ thuộc vào độ khó của bài toán tính logarit rời rạc trên các số lớn

Ví dụ như tính từ Y_A trong công thức:

$$Y_A = \alpha^{X_A} \bmod p$$

Phải dùng phép tính logarit rời rạc. Hoặc để tính K trong công thức $K = (Y_A)^k \bmod p$ người thám mã phải tính được k từ công thức

$C_1 = \alpha^k \bmod p$ cũng phải dùng phép tính logarit rời rạc.

VÍ DỤ THUẬT TOÁN

Chọn số nguyên tố: p = 97, và giá trị sinh của nó: α = 5

- A chọn khoá cá nhân $X_A = 58$
- A tính:

$$Y_A = \alpha^{X_A} \bmod p = 5^{58} \bmod 97 = 44$$

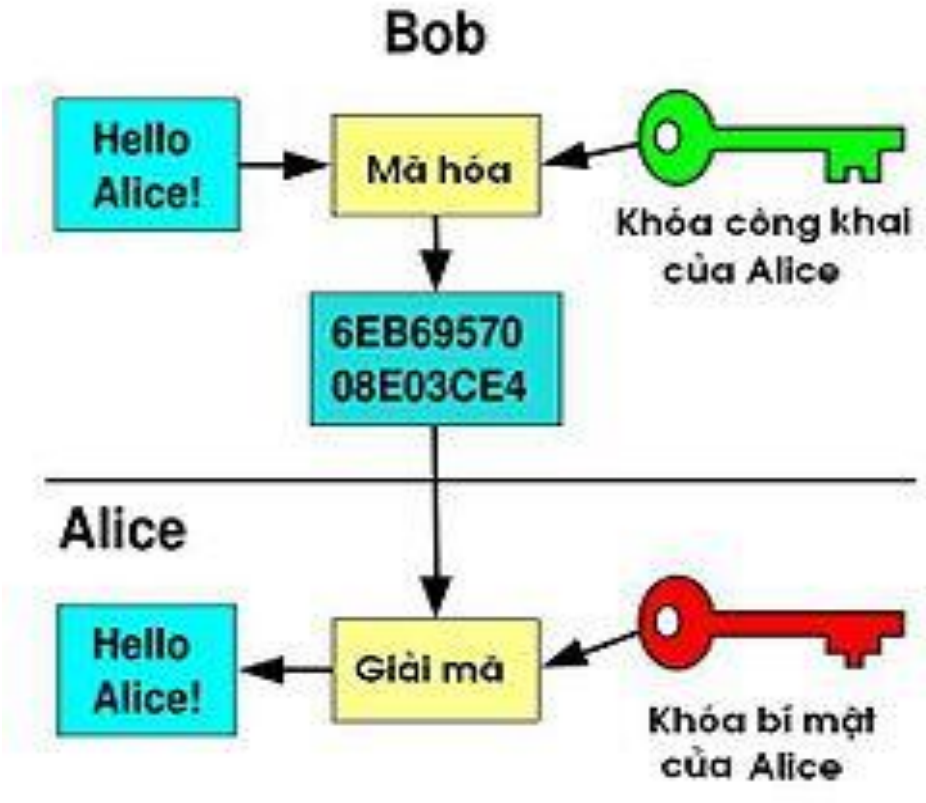
Khóa công khai của A là $\{p, \alpha, Y_A\} = \{97, 5, 44\}$ được gửi cho B B muốn gửi thông điệp M = 3 cho A

- B chọn k = 36, tính $K = (Y_A)^k \bmod p = 44^{36} \bmod 97 = 75$
- B tính $C_1 = \alpha^k \bmod p = 5^{36} \bmod 97 = 50$

$C_2 = (K \times M) \bmod p = (75 \times 3) \bmod 97 = 31$ Nhận được thông điệp: $\{C_1, C_2\} = (50, 31)$, A giải mã:

$K = C_1^{X_A} \bmod p = 50^{58} \bmod 97 = 75 M = (C_2 \times K^{-1}) \bmod p = (31 \times K^{-1}) \bmod 97 = (31 \times 22) \bmod 97 = 3$

SƠ ĐỒ THUẬT TOÁN



VIDEO GIẢI THÍCH ELGAMAL

- Xem video giải thích Elgamal trên Youtube

https://www.youtube.com/watch?v=3ooB1-T2x_U

https://www.youtube.com/watch?v=UrzZDvb4cw4

Quá trình tạo khóa của A sử dụng hệ ElGamal gồm các bước chính sau:

- A, B thống nhất số nguyên tố q và phần tử sinh q: α
- Bên tạo khóa (A) chọn giá trị bí mật X_a ($X_a < q-1$) và tính giá trị $Y_a = \alpha^{X_A} \bmod q$. Khi đó, bộ khóa K = {PU, PR} của A, với khóa công khai PU = {q, α, Y_A} và khóa bí mật PR = {X_A}

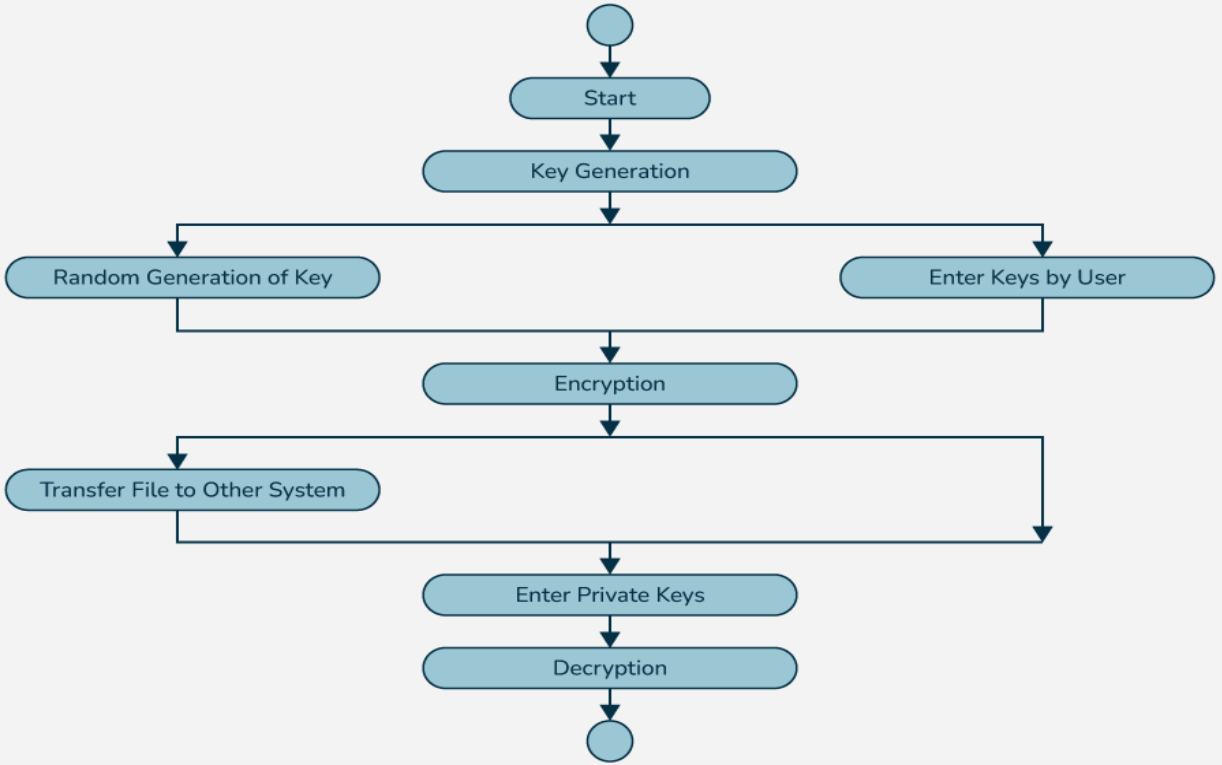
Quá trình B sử dụng bộ khóa của A trong việc truyền dữ liệu M (M<q):

- B chọn giá trị k (k<q) và tính toán khóa $K = (Y_A)^K \bmod q$, $C_1 = \alpha^k \bmod q$, $C_2 = K*M \bmod q$. Khi đó (C1, C2) là bản mã được truyền đi

Quá trình bên nhận (A) giải mã:

- Tính khóa $K = (C_1)^{X_A} \bmod q$
- Tìm bản gốc theo công thức: $M = (C_2 K^{-1}) \bmod q$

ElGamal Encryption Algorithm



Source: https://www.geeksforgeeks.org/elgamal-encryption-algorithm/

TẠO KHÓA

Bên nhận (người muốn nhận tin nhắn mã hóa) thực hiện các bước sau:

1. **Chọn tham số:**
   - Chọn một số nguyên tố lớn p.
   - Chọn một phần tử nguyên thủy g trong trường $\mathbb{Z}_p^*$ (tức là g là số có bậc tối đa trong nhóm phần dư modulo p).
2. **Tạo khóa bí mật và khóa công khai:**
   - Chọn một số ngẫu nhiên x sao cho $1 \leq x \leq p-2$ (đây là khóa bí mật).
   - Tính $y = g^x \mod p$ (đây là khóa công khai).

Khóa công khai: (p, g, y)

Khóa bí mật: x

Người gửi muốn mã hóa một bản tin M (được biểu diễn dưới dạng một số trong phạm vi $1 \leq M \leq p-1$), thực hiện các bước sau:

1. Chọn một số ngẫu nhiên k sao cho $1 \leq k \leq p-2$.
2. Tính $c_1 = g^k \mod p$.
3. Tính $c_2 = M \cdot y^k \mod p$.

Bản mã gửi đi là cặp (c_1, c_2).

Bên nhận (người có khóa bí mật x) nhận được bản mã (c_1, c_2), thực hiện giải mã như sau:

1. Tính $s = c_1^x \mod p$ (vì $s = g^{kx} \mod p$).
2. Tính nghịch đảo của s theo modulo p, tức là $s^{-1} \mod p$.
3. Khôi phục bản rõ:

$$M = c_2 \cdot s^{-1} \mod p$$

Do $c_2 = M \cdot y^k \mod p$ và $y^k = g^{kx}$, nên khi nhân với s^{-1} sẽ triệt tiêu g^{kx}, khôi phục được M.

VÍ DỤ MINH HỌA

Giả sử:

- $p = 23, g = 5$
- Khóa bí mật $x = 6$, khóa công khai $y = g^x \mod p = 5^6 \mod 23 = 8$

Người gửi muốn mã hóa $M = 10$ với $k = 15$:

1. $c_1 = 5^{15} \mod 23 = 19$
2. $c_2 = 10 \times 8^{15} \mod 23 = 5$

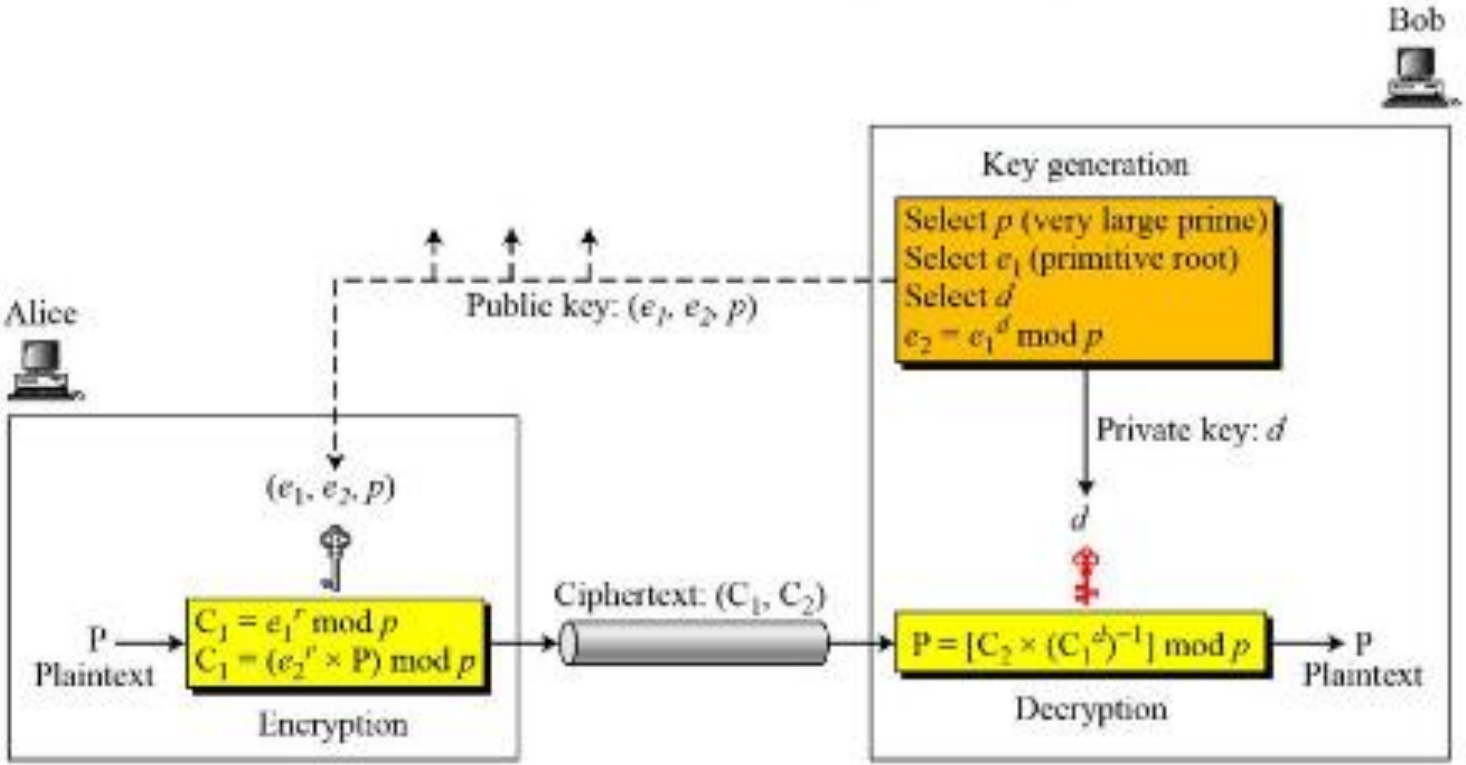
Bản mã gửi đi: $(19, 5)$.

VÍ DỤ MINH HỌA

Người nhận giải mã:

1. $s = 19^6 \mod 23 = 2$.
2. $s^{-1} \mod 23 = 12$ (vì $2 \times 12 \mod 23 = 1$).
3. $M = 5 \times 12 \mod 23 = 10$ (khôi phục được bản rõ).

THUẬT TOÁN ELGAMAL



ElGamal Encryption Algorithm

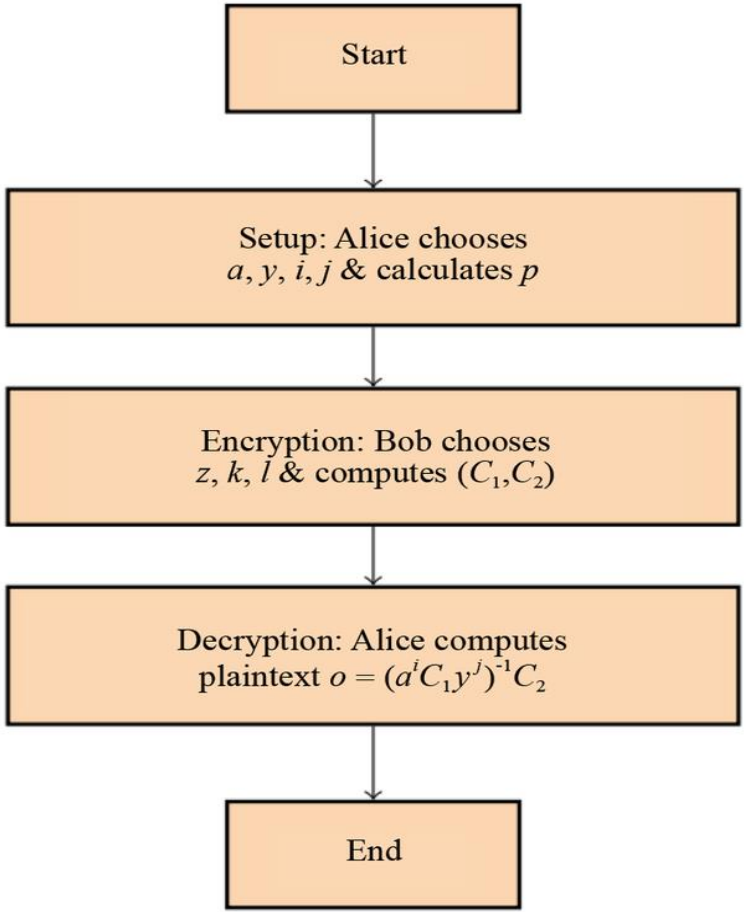
Last Updated : 29 Nov, 2024

ElGamal Encryption is a public-key cryptosystem. It uses asymmetric key encryption to communicate between two parties and encrypt the message. This cryptosystem is based on the difficulty of finding **discrete logarithms** in a cyclic group that is even if we know g^a and g^k, it is extremely difficult to compute g^{ak}. In this article, we will learn about the Elgamal algorithm, the components of its algorithm, its advantages & disadvantages, and the implementation of the ElGamal cryptosystem in Python.

Elgamal Cryptographic Algorithm

The ElGamal cryptographic algorithm is an asymmetric key encryption scheme based on the Diffie-Hellman key exchange. It was invented by Taher ElGamal in 1985. The algorithm is widely used for secure data transmission and has digital signatures and encryption applications. Here's an overview of its components and how it works:

SƠ ĐỒ THUẬT TOÁN



Ưu điểm & Nhược điểm

Ưu điểm:

- Độ bảo mật cao do dựa vào bài toán logarit rời rạc.
- Dùng trong nhiều ứng dụng như chữ ký số, bảo mật dữ liệu.

Nhược điểm:

- Bản mã dài gấp đôi bản rõ.
- Cần tạo số ngẫu nhiên mới cho mỗi lần mã hóa để đảm bảo an toàn.

THỰC HÀNH ONLINE ELGAMAL

- Quay lại bài thực hành website Elgamal online: https://cryptocalc.com.au/elgamal-crypto-calc/
- Plain Text to Encrypt: DNU
- Encrypted Output: ???
- Try decrypt them: ???

Dùng ChatGPT hỏi bằng tiếng Anh và em hãy giải thích ý hiểu của em? Your marks will be graded higher if you can explain in English.

- What is Elgamal?
- How can Elgamal work?
- What are applications of Elgamal in reality?

Elgamal Digital Signature Scheme:

Elgamal encryption scheme is designed to enable encryption by a user's public key with decryption by the user's private key. The Elgamal signature scheme involves the use of the private key for digital signature generation and the public key for digital signature verification.

Preliminary:

If q is a prime number and α is a primitive root of *q*, then

1. For any integer m, $\alpha^m \equiv 1 \pmod{q}$ if and only if $m \equiv 0 \pmod{q-1}$.
2. For any integers, i, j, $\alpha^i \equiv \alpha^j \pmod{q}$ if and only if $i \equiv j \pmod{q-1}$.

Elgamal Key Generation:

As with Elgamal encryption, the global elements of **Elgamal digital signature** are a prime number ***q*** and α, which is a primitive root of ***q***. User **A** generates a private/public key pair as follows:

1. Generate a random integer X_A, such that $1 < X_A < q - 1$.
2. Compute $Y_A = \alpha^{X_A} \bmod q$.
3. A's private key is X_A; A's pubic key is $\{q, \alpha, Y_A\}$.

Elgamal Digital Signature Generation:

To sign a message ***M***, user **A** first computes the hash ***m* = H(*M*),** such that ***m*** is an integer in the range **0 ≤ *m* ≤ *q* - 1**. **A** then forms a digital signature as follows:

Elgamal Digital Signature Generation (Continued):

1. Choose a random integer K such that $1 \leq K \leq q - 1$ and $\gcd(K, q - 1) = 1$. That is, K is relatively prime to $q - 1$.
2. Compute $S_1 = \alpha^K \bmod q$. Note that this is the same as the computation of C_1 for Elgamal encryption.
3. Compute $K^{-1} \bmod (q - 1)$. That is, compute the inverse of K modulo $q - 1$.
4. Compute $S_2 = K^{-1}(m - X_A S_1) \bmod (q - 1)$.
5. The signature consists of the pair (S_1, S_2).

Elgamal Digital Signature Verification:

Any user **B** can verify the signature as follows:

1. Compute $V_1 = \alpha^m \bmod q$.
2. Compute $V_2 = (Y_A)^{S_1}(S_1)^{S_2} \bmod q$.

The signature is valid if $\boldsymbol{V_1} = \boldsymbol{V_2}$. Let us demonstrate that this is so.

Example (1):

Let us start with the prime field GF(19); that is, $\boldsymbol{q}$ = 19. It has primitive roots {2, 3, 10, 13, 14, 15}. We choose $\boldsymbol{\alpha}$ = 10.

Alice generates a key pair as follows:

1. Alice chooses $X_A = 16$.
2. Then $Y_A = \alpha^{X_A} \bmod q = \alpha^{16} \bmod 19 = 4$.
3. Alice's private key is 16; Alice's pubic key is $\{q, \alpha, Y_A\} = \{19, 10, 4\}$.

Example (1): (Continued)

(Sign) Suppose Alice wants to sign a message with hash value ***m*** = 14.

1. Alice chooses $K = 5$, which is relatively prime to $q - 1 = 18$.
2. $S_1 = \alpha^K \bmod q = 10^5 \bmod 19 = 3$ (see Table 2.7).
3. $K^{-1} \bmod (q - 1) = 5^{-1} \bmod 18 = 11$.
4. $S_2 = K^{-1}(m - X_A S_1) \bmod (q - 1) = 11\,(14 - (16)(3)) \bmod 18 = -374 \bmod 18 = 4$.

(Verify): Bob can verify the signature as follows:

1. $V_1 = \alpha^m \bmod q = 10^{14} \bmod 19 = 16$.
2. $V_2 = (Y_A)^{S_1}(S_1)^{S_2} \bmod q = (4^3)(3^4) \bmod 19 = 5184 \bmod 19 = 16$.

Thus, the signature is valid because $V_1 = V_2$.

NIST Digital Signature Approach:

NIST has published Federal Information Processing Standard **FIPS 186**, known as the Digital Signature Algorithm (**DSA**). The **DSA** makes use of the Secure Hash Algorithm (**SHA**). The **DSA** was originally proposed in 1991. Several expanded versions of the standard were then issued as FIPS 186-2, FIPS 186-3 and FIPS 186-4 in response to public feedback concerning the security of the scheme. This latest version also incorporates digital signature algorithms based on RSA and on elliptic curve cryptography.

LAB LUYỆN TẬP

1. **Làm Lab8 trên LMS**
2. Tham khảo thêm source codes C++ tại links:

https://github.com/kevinhaeni/Crypto.Cipher.ElGamal.Cpp

https://onecompiler.com/cpp/3wvehmbxr

https://www.geeksforgeeks.org/elgamal-encryption-algorithm/

- Xây dựng chương trình thực hiện thuật toán Elgamal với ứng dụng nhắn tin có mã hóa giữa 2 bên A và B
- Xây dựng chương trình thực hiện gửi file dữ liệu có mã hóa Elgamal

- Hoạt động của thuật toán Elgamal
- Một số ví dụ minh họa
- Ứng dụng Elgamal
- Luyện tập với code C++

ĐẠI NAM
UNIVERSITY