Lab 8: Coding Elgamal with C++

Author: Trần Quý Nam

Date: 28/3/2025

I. SINH VIÊN CÀI ĐẶT MÔI TRƯỜNG:

Sinh viên chuẩn bị môi trường thực hành mã nguồn thuật toán Elgamal với C++, sử dụng IDE là Visual Studio Code, cài đặt các thư viện liên quan và mã nguồn C++ như bên dưới. Sinh viên thực hiện các Bài để cài đặt, cấu hình và chạy thử đoạn mã nguồn sau:

```cpp
#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;
typedef long long ll;

// Hàm tính (base^exp) % mod bằng phương pháp bình phương lặp
ll modExp(ll base, ll exp, ll mod) {
    ll result = 1;
    while (exp > 0) {
        if (exp % 2 == 1) {
            result = (result * base) % mod;
        }
        base = (base * base) % mod;
        exp /= 2;
    }
    return result;
}

// Hàm tìm nghịch đảo modulo bằng thuật toán Euclidean mở rộng
ll modInverse(ll a, ll mod) {
    ll m0 = mod, t, q;
    ll x0 = 0, x1 = 1;

    if (mod == 1) return 0; // Không tồn tại

    while (a > 1) {
        q = a / mod;
        t = mod;
        mod = a % mod, a = t;
        t = x0;
        x0 = x1 - q * x0;
        x1 = t;
```

```
    }
    if (x1 < 0) x1 += m0; // Đảm bảo giá trị dương
    return x1;
}

// Hàm sinh khóa ElGamal
void generateKeys(ll &p, ll &g, ll &x, ll &y) {
    srand(time(0)); // Seed ngẫu nhiên

    p = 23;  // Số nguyên tố (có thể thay bằng số lớn hơn)
    g = 5;   // Căn nguyên thủy modulo p
    x = rand() % (p - 2) + 1; // Chọn x trong khoảng [1, p-2]
    y = modExp(g, x, p); // Tính y = g^x mod p

    cout << "Public Key: (p=" << p << ", g=" << g << ", y=" << y << ")\n";
    cout << "Private Key: (x=" << x << ")\n";
}

// Hàm mã hóa thông điệp M
void encrypt(ll M, ll p, ll g, ll y, ll &c1, ll &c2) {
    ll k = rand() % (p - 2) + 1; // Chọn k ngẫu nhiên trong khoảng [1, p-2]
    c1 = modExp(g, k, p); // c1 = g^k mod p
    c2 = (M * modExp(y, k, p)) % p; // c2 = M * y^k mod p
}

// Hàm giải mã bản mã (c1, c2)
ll decrypt(ll c1, ll c2, ll p, ll x) {
    ll s = modExp(c1, x, p); // s = c1^x mod p
    ll s_inv = modInverse(s, p); // s^-1 mod p
    return (c2 * s_inv) % p; // M = c2 * s^-1 mod p
}

int main() {
    ll p, g, x, y; // Khai báo khóa
    generateKeys(p, g, x, y); // Sinh khóa

    ll M;
    cout << "Enter a message (integer): ";
    cin >> M;

    ll c1, c2;
    encrypt(M, p, g, y, c1, c2);
    cout << "Ciphertext: (" << c1 << ", " << c2 << ")\n";

    ll decryptedM = decrypt(c1, c2, p, x);
```

```
    cout << "Decrypted Message: " << decryptedM << endl;

    return 0;
}
```

II. SINH VIÊN THỰC HÀNH CÁC BÀI TẬP SAU:

Bài 1: Chạy chương trình

- Biên dịch và chạy chương trình, nhập một số để kiểm tra kết quả mã hóa/giải mã.
- Sinh viên giải thích trước cả lớp và giảng viên về ý nghĩa, hoạt động và kết quả các dòng code C++

Bài 2: Kiểm tra giá trị p, g khác nhau

- Thay đổi giá trị của p (số nguyên tố lớn hơn) và g (căn nguyên thủy) và xem kết quả.
- Thử nghiệm với các số cần mã hóa khác nhau

Bài 3: Kiểm tra với nhiều thông điệp

- Nhập các số lớn hơn 10 và xem kết quả mã hóa/giải mã có chính xác không.
- Thay đổi nhiều số khác nhau và xem kết quả

Bài 4: Xây dựng chương trình thực hiện thuật toán Elgamal với ứng dụng nhắn tin có mã hóa giữa 2 bên A và B. Có thể sử dụng một trong các cách sau:

- Sử dụng lập trình Socket Programming dựa trên giao thức TCP/IP dùng ngôn ngữ C++/Java hay Python.
- Sử dụng lập trình Web, dùng các framework với ngôn ngữ JavaScript hay Python.
- Sử dụng các giao diện đồ họa GUI trên bất cứ ngôn ngữ lập trình và thự viện đồ họa nào sinh viên yêu thích.

Bài 5: Xây dựng chương trình thực hiện thuật toán Elgamal với file dữ liệu có ghi nội dung liên quan đến Dai Nam University

- Sử dụng lập trình C++ vào ra với tệp dữ liệu.
- Mã hóa Elgamal dữ liệu trong file và ghi vào file khác.
- Đọc file mã hóa, giải mã Elgamal và chuyển thành file dữ liệu ban đầu.

Bài 6: Nâng cấp thuật toán

- Chuyển từ mã hóa số nguyên sang mã hóa chuỗi (chuyển từng ký tự thành số).
- Phát triển codes để mã hóa thông điệp "DaiNam", "FIT"…
- Viết thêm chức năng tạo số nguyên tố lớn tự động thay vì gán cố định.
- Thêm tính năng đo thời gian thực thi để xem hiệu suất với các số lớn.
