ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

BÁO CÁO ĐỒ ÁN:
CHỮ KÝ SỐ

ĐỒ ÁN CUỐI KỲ

NHẬP MÔN MÃ HÓA MẬT MÃ

Giáo viên hướng dẫn:

ThS. Lê Phúc Lữ

ThS. Trần Hà Sơn

ThS. Nguyễn Đình Thúc

Thành viên:

21127056 – Lâm Thiều Huy

Tp. Hồ Chí Minh, tháng 12/2023

Lời cảm ơn

Trước khi bắt đầu trình bày về đồ án "Digital signature ", em muốn bày tỏ lòng cảm ơn đến cả ba giảng viên đã truyền đạt cho em nhiều kiến thức và kinh nghiệm để giúp em hoàn thành được đồ án này. Tuy còn nhiều thiếu sót bởi vì số lượng thành viên trong nhóm chỉ có 1 người. Những nhận xét, góp ý của thầy từ buổi vấn đáp cuối kì sẽ là nền tảng để chúng em cải thiện và rút kinh nghiệm cho những đồ án lần sau được tốt hơn.

Mục Lục

1. Chương 1 Thông tin thành viên

- Tên thành viên: Lâm Thiều Huy - 21127056
- Tên đồ án : Digital Signature

2. Chương 2 Báo cáo đồ án

2.1. Chữ ký số RSA:

2.1.1. Hàm modPow(long long base, long long exp, long long modulus)

- Mục đích: Tính base^exp % modulus.
- Cách hoạt động:
  - Lặp cho đến khi exp bằng 0.
  - Nếu exp lẻ, nhân result với base và lấy phần dư theo modulus.
  - Bình phương base và lấy phần dư theo modulus.
  - Chia exp cho 2.
- Ứng dụng trong RSA: Dùng cho việc mã hóa, giải mã và tạo chữ ký.

2.1.2. Hàm gcd(long long a, long long b)

- Mục đích: Tìm ước chung lớn nhất của a và b.
- Cách hoạt động:
  - Sử dụng thuật toán Euclid: Lặp lại việc thay a bằng b và b bằng a % b cho đến khi b bằng 0.
  - Kết quả là a.
- Ứng dụng trong RSA: Xác định e sao cho GCD(e, φ(n)) = 1.

2.1.3. Hàm modInverse(long long a, long long m)

- Mục đích: Tìm nghịch đảo mô-đun của a đối với m.
- Cách hoạt động:
  - Lặp qua từ 1 đến m-1 để tìm x sao cho (a * x) % m == 1.
- Ứng dụng trong RSA: Tính khóa riêng d từ khóa công khai e.

2.1.4. Hàm generateRSAKeys(long long p, long long q, long long& e, long long& d, long long& n)

- Mục đích: Sinh cặp khóa RSA.
- Cách hoạt động:
  - Tính n = p * q và phi = (p - 1) * (q - 1).
  - Chọn e sao cho 1 < e < phi và gcd(e, phi) = 1.
  - Tính d là nghịch đảo mô-đun của e đối với phi.
- Ứng dụng trong RSA: Tạo khóa công khai và riêng.

2.1.5. Hàm signMessage(long long message, long long d, long long n)

- Mục đích: Tạo chữ ký cho một thông điệp.
- Cách hoạt động:
  - Mã hóa thông điệp message bằng cách sử dụng khóa riêng d (tính message^d % n).

- Ứng dụng trong RSA: Ký thông điệp.

2.1.6. Hàm verifySignature(long long message, long long signature, long long e, long long n)

- Mục đích: Xác minh chữ ký số.
- Cách hoạt động:
  - Giải mã chữ ký signature bằng khóa công khai e (tính signature^e % n).
  - So sánh kết quả với thông điệp gốc.
- Ứng dụng trong RSA: Xác minh chữ ký.

2.1.7. Phần main()

- Mục đích: Chạy quy trình sinh khóa, ký và xác minh chữ ký RSA.
- Cách hoạt động:
  - Sinh khóa RSA từ hai số nguyên tố p và q.
  - Đọc từng số từ file, ký và lưu vào file khác.
  - Xác minh chữ ký từ file đã ký và so sánh với file gốc.
- Ứng dụng trong RSA: Mô phỏng quy trình sử dụng chữ ký số RSA.

2.2. Chữ ký số DSA:

2.2.1. Hàm modPow(long long base, long long exp, long long modulus)

- Mục đích: Tính base^exp % modulus, một phép tính quan trọng trong nhiều thuật toán mật mã.
- Cách hoạt động: Sử dụng phương pháp lũy thừa bình phương (Exponentiation by Squaring) để giảm độ phức tạp tính toán.
- Ứng dụng trong DSA: Được dùng để tính g^k % p trong hàm signMessageDSA và modPow(g, u1, p) * modPow(y, u2, p) % p % q trong hàm verifySignatureDSA.

2.2.2. Hàm modInverse(long long a, long long m)

- Mục đích: Tìm nghịch đảo mô-đun của a trong m, tức là tìm x sao cho a*x % m = 1.
- Cách hoạt động: Duyệt qua các số từ 1 đến m-1 để tìm giá trị thích hợp.
- Ứng dụng trong DSA: Dùng để tính s trong hàm signMessageDSA và w trong hàm verifySignatureDSA.

2.2.3. Hàm findGenerator(long long p, long long q)

- Mục đích: Tìm phần tử sinh g cho nhóm Z_p*.
- Cách hoạt động: Tìm số g sao cho g = h^((p - 1) / q) % p không bằng 1 hoặc 0, với h là số bắt đầu từ 2 và tăng dần.
- Ứng dụng trong DSA: Tạo ra g, một phần quan trọng của khóa công khai.

2.2.4. Hàm generateDSAKeys(long long p, long long q, long long g, long long& y, long long& x)

- Mục đích: Tạo cặp khóa cho DSA.
- Cách hoạt động:
  - Tạo khóa riêng x ngẫu nhiên trong khoảng từ 1 đến q-1.
  - Tính khóa công khai y = g^x % p.
- Ứng dụng trong DSA: Tạo ra cặp khóa công khai và riêng tư cho người dùng.

2.2.5. Hàm signMessageDSA(long long message, long long p, long long q, long long g, long long x, long long& r, long long& s)

- Mục đích: Ký một thông điệp.
- Cách hoạt động:
  - Tạo số ngẫu nhiên k từ 1 đến q-1.
  - Tính r = (g^k % p) % q và s = (k^(-1) * (message + x * r)) % q.
- Ứng dụng trong DSA: Tạo chữ ký gồm cặp (r, s) cho thông điệp.

2.2.6. Hàm verifySignatureDSA(long long message, long long r, long long s, long long p, long long q, long long g, long long y)

- Mục đích: Xác minh chữ ký.
- Cách hoạt động:
  - Tính w = s^(-1) % q, u1 = (message * w) % q, u2 = (r * w) % q.
  - Tính v = ((g^u1 % p * y^u2 % p) % p) % q.
  - So sánh v với r để xác định tính hợp lệ của chữ ký.
- Ứng dụng trong DSA: Kiểm tra xem chữ ký có phải là của thông điệp đã cho hay không.

2.2.7. Phần main()

- Mục đích: Thực thi quy trình tạo khóa, ký và xác minh chữ ký.
- Cách hoạt động:
  - Tạo khóa DSA từ p, q, và g.
  - Đọc thông điệp từ file, tạo chữ ký và lưu vào file mới.
  - Đọc chữ ký từ file đã ký và xác minh chữ ký.
- Ứng dụng trong DSA: Mô phỏng quy trình sử dụng chữ ký số DSA trong thực tế.

2.3. Chữ ký số Elgamal:

2.3.1. Hàm modPow(long long base, long long exp, long long modulus)

- Mục đích: Tính base^exp % modulus.
- Cách hoạt động:
  - Lặp qua exp khi exp > 0.

        - Nếu exp lẻ, nhân result với base và lấy phần dư theo modulus.
        - Bình phương base và lấy phần dư theo modulus.
        - Chia exp cho 2.
    - Ứng dụng trong ElGamal:
        - Tính g^k % p trong hàm signMessage.
        - Tính modPow(h, s1, p) * modPow(s1, s2, p) % p trong hàm verifySignature.

2.3.2. Hàm modInverse(long long a, long long m)

- Mục đích: Tìm nghịch đảo mô-đun của a trong m.
- Cách hoạt động: Duyệt qua các số từ 1 đến m-1 để tìm x sao cho (a * x) % m == 1.
- Ứng dụng trong ElGamal: Tính k^(-1) và s2 trong hàm signMessage.

2.3.3. Hàm randomInRange(long long min, long long max)

- Mục đích: Sinh số ngẫu nhiên trong khoảng từ min đến max.
- Cách hoạt động: Sử dụng rand() để tạo số ngẫu nhiên và điều chỉnh nó nằm trong khoảng mong muốn.
- Ứng dụng trong ElGamal: Sinh khóa riêng x trong hàm generateElGamalKeys.

2.3.4. Hàm generateElGamalKeys(long long p, long long g, long long& h, long long& x)

- Mục đích: Tạo cặp khóa cho ElGamal.
- Cách hoạt động:
    - Sinh x ngẫu nhiên trong khoảng từ 1 đến p-2.
    - Tính h = g^x % p.
- Ứng dụng trong ElGamal: Tạo khóa riêng x và khóa công khai h.

2.3.5. Hàm signMessage(long long message, long long p, long long g, long long x, long long& s1, long long& s2)

- Mục đích: Ký một thông điệp.
- Cách hoạt động:
    - Chọn k cố định hoặc ngẫu nhiên, tạo s1 = g^k % p.
    - Tính s2 = (k^(-1) * (message - x * s1)) % (p-1).
- Ứng dụng trong ElGamal: Tạo chữ ký gồm hai phần s1 và s2.

2.3.6. Hàm verifySignature(long long message, long long s1, long long s2, long long p, long long g, long long h)

- Mục đích: Xác minh chữ ký.
- Cách hoạt động:

    - Kiểm tra điều kiện hợp lệ của s1 và s2.
    - Tính v1 = g^message % p và v2 = (h^s1 * s1^s2) % p.
    - So sánh v1 và v2.
- Ứng dụng trong ElGamal: Kiểm tra xem chữ ký có hợp lệ không.

2.3.7. Phần main()

- Mục đích: Thực thi quy trình tạo khóa, ký và xác minh chữ ký.
- Cách hoạt động:
    - Sinh khóa ElGamal.
    - Đọc từng dòng từ file, ký chúng và ghi vào file khác.
    - Đọc và xác minh chữ ký từ file đã ký.
- Ứng dụng trong ElGamal: Mô phỏng quy trình sử dụng chữ ký số ElGamal.

3. Chương 3 Tổng kết

3.1. RSA (Rivest–Shamir–Adleman)

- Cơ sở toán học: Dựa trên khó khăn của việc phân tích thừa số nguyên tố lớn.
- Khóa: Sử dụng cặp khóa công khai và riêng tư. Khóa công khai là (n, e) và khóa riêng là (n, d).
- Bảo mật: Rất an toàn với kích thước khóa lớn. Độ an toàn tăng tỷ lệ với kích thước khóa.
- Hiệu suất: Có hiệu suất kém hơn DSA và ElGamal trong việc ký nhưng tốt hơn trong xác minh.
- Ứng dụng: Rộng rãi trong mã hóa và chữ ký số. Được sử dụng trong nhiều giao thức và hệ thống an ninh mạng.

3.2. DSA (Digital Signature Algorithm)

- Cơ sở toán học: Dựa trên bài toán logarit rời rạc trong nhóm hữu hạn.
- Khóa: Sử dụng cặp khóa công khai và riêng tư. Khóa công khai bao gồm (p, q, g, y) và khóa riêng là x.
- Bảo mật: An toàn nhưng yêu cầu việc chọn tham số cẩn thận. Cần phải chọn số k ngẫu nhiên và duy nhất mỗi khi ký.
- Hiệu suất: Ký nhanh hơn RSA nhưng xác minh chậm hơn.
- Ứng dụng: Phổ biến trong các hệ thống chứng thực và chữ ký số, đặc biệt là khi cần ký nhanh.

3.3. ElGamal

- Cơ sở toán học: Cũng dựa trên bài toán logarit rời rạc.
- Khóa: Sử dụng cặp khóa công khai và riêng tư. Khóa công khai là (p, g, h) và khóa riêng là x.
- Bảo mật: An toàn nhưng kích thước chữ ký lớn có thể là một hạn chế.
- Hiệu suất: Kém hiệu quả hơn cả RSA và DSA về mặt kích thước chữ ký và tốc độ xử lý.
- Ứng dụng: Ít phổ biến hơn so với RSA và DSA, thường được dùng trong các tình huống cụ thể.

3.4. Tổng hợp

- RSA nổi bật với sự đa dụng và an toàn, đặc biệt phù hợp trong các ứng dụng cần cả mã hóa và chữ ký số.
- DSA được ưa chuộng trong các hệ thống yêu cầu hiệu suất ký cao và đảm bảo mức độ bảo mật ổn định, nhưng nó kém hiệu quả khi xác minh.
- ElGamal cung cấp một lựa chọn an toàn nhưng thường không được ưa chuộng do kích thước chữ ký lớn và hiệu suất kém.