***Họ tên: Phạm Anh Tú***

***MSV: 1811505310348***

**Bài tập Lab 3 - RSA with OpenSSL**

**Bài 1. Tính khóa bí mật**

* **Code**

#include <stdio.h>

#include <openssl/bn.h>

#define NBITS 128

void printBN(char \*msg, BIGNUM \*a, BIGNUM \*b)

{

char \*number\_str\_a = BN\_bn2hex(a);

char \*number\_str\_b = BN\_bn2hex(b);

printf("%s (%s,%s)\n", msg, number\_str\_a, number\_str\_b);

OPENSSL\_free(number\_str\_a);

OPENSSL\_free(number\_str\_b);

}

int main()

{

// Khởi tạo

BN\_CTX \*ctx = BN\_CTX\_new();

BIGNUM \*p = BN\_new();

BIGNUM \*q = BN\_new();

BIGNUM \*n = BN\_new();

BIGNUM \*phi = BN\_new();

BIGNUM \*e = BN\_new();

BIGNUM \*d = BN\_new();

BIGNUM \*res = BN\_new();

BIGNUM \*p\_minus\_1 = BN\_new();

BIGNUM \*q\_minus\_1 = BN\_new();

// Gán giá trị

BN\_hex2bn(&p, "F7E75FDC469067FFDC4E847C51F452DF");

BN\_hex2bn(&q, "E85CED54AF57E53E092113E62F436F4F");

BN\_hex2bn(&e, "0D88C3");

// n = pq

BN\_mul(n, p, q, ctx);

printBN("public key", e, n);

// Tính phi(n) = (p-1)\*(q-1)

BN\_sub(p\_minus\_1, p, BN\_value\_one());

BN\_sub(q\_minus\_1, q, BN\_value\_one());

BN\_mul(phi, p\_minus\_1, q\_minus\_1, ctx);

// Tính d

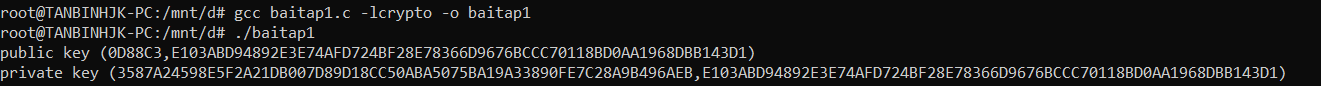
BN\_mod\_inverse(d, e, phi, ctx);

printBN("private key", d, n);

return 0;

}

* **Kết quả chạy:**



**Bài 2. Mã hóa tin nhắn**

* **Code:**

#include <stdio.h>

#include <openssl/bn.h>

void printBN(char \*msg, BIGNUM \*a)

{

char \*number\_str\_a = BN\_bn2hex(a);

printf("%s %s\n", msg, number\_str\_a);

OPENSSL\_free(number\_str\_a);

}

int main()

{

// khởi tạo

BN\_CTX \*ctx = BN\_CTX\_new();

BIGNUM \*n = BN\_new();

BIGNUM \*e = BN\_new();

BIGNUM \*M = BN\_new();

BIGNUM \*checkM = BN\_new();

BIGNUM \*d = BN\_new();

BIGNUM \*C = BN\_new();

// gán giá trị

BN\_hex2bn(&n, "DCBFFE3E51F62E09CE7032E2677A78946A849DC4CDDE3A4D0CB81629242FB1A5");

BN\_dec2bn(&e, "65537");

BN\_hex2bn(&M, "436f6e67206e6768652074686f6e672074696e21"); //hex encode for "Cong nghe thong tin!"

BN\_hex2bn(&d, "74D806F9F3A62BAE331FFE3F0A68AFE35B3D2E4794148AACBC26AA381CD7D30D");

// encrypt M: M^e mod n

BN\_mod\_exp(C, M, e, n, ctx);

printBN("Encryption result:", C);

//decrypt C: C^d mod n

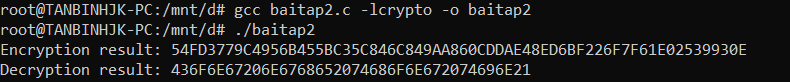
BN\_mod\_exp(checkM , C, d, n, ctx);

printBN("Decryption result:", checkM);

return 0;

}

* **Kết quả chạy:**



**Bài 3. Giải mã tin nhắn**

* **Code:**

#include <stdio.h>

#include <openssl/bn.h>

void printBN(char \*msg, BIGNUM \*a)

{

char \*number\_str\_a = BN\_bn2hex(a);

printf("%s %s\n", msg, number\_str\_a);

OPENSSL\_free(number\_str\_a);

}

int main()

{

// khởi tạo

BN\_CTX \*ctx = BN\_CTX\_new();

BIGNUM \*n = BN\_new();

BIGNUM \*M = BN\_new();

BIGNUM \*d = BN\_new();

BIGNUM \*C = BN\_new();

// gán giá trị

BN\_hex2bn(&n, "DCBFFE3E51F62E09CE7032E2677A78946A849DC4CDDE3A4D0CB81629242FB1A5");

BN\_hex2bn(&d, "74D806F9F3A62BAE331FFE3F0A68AFE35B3D2E4794148AACBC26AA381CD7D30D");

BN\_hex2bn(&C, "8C0F971DF2F3672B28811407E2DABBE1DA0FEBBBDFC7DCB67396567EA1E2493F");

// decrypt C: C^d mod n

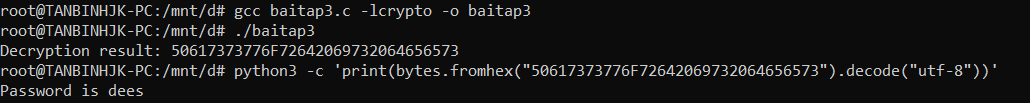
BN\_mod\_exp(M, C, d, n, ctx);

printBN("Decryption result:", M);

return 0;

}

* **Kết quả chạy:**



**Bài 4. Ký**

* **Code:**

#include <stdio.h>

#include <openssl/bn.h>

void printBN(char \*msg, BIGNUM \*a)

{

char \*number\_str\_a = BN\_bn2hex(a);

printf("%s %s\n", msg, number\_str\_a);

OPENSSL\_free(number\_str\_a);

}

int main()

{

// khởi tạo

BN\_CTX \*ctx = BN\_CTX\_new();

BIGNUM \*n = BN\_new();

BIGNUM \*d = BN\_new();

BIGNUM \*M1 = BN\_new();

BIGNUM \*M2 = BN\_new();

BIGNUM \*C1 = BN\_new();

BIGNUM \*C2 = BN\_new();

// Gán giá trị

BN\_hex2bn(&n, "DCBFFE3E51F62E09CE7032E2677A78946A849DC4CDDE3A4D0CB81629242FB1A5");

BN\_hex2bn(&d, "74D806F9F3A62BAE331FFE3F0A68AFE35B3D2E4794148AACBC26AA381CD7D30D");

//python3 -c 'print("Toi da vay cua ban 50,000,000vnd.".encode().hex())'

BN\_hex2bn(&M1, "546f6920646120766179206375612062616e2035302c3030302c303030766e642e"); // hex encode for "Toi da vay cua ban 50,000,000vnd."

//python3 -c 'print("Toi da vay cua ban 100,000,000vnd.".encode().hex())'

BN\_hex2bn(&M2, "546f6920646120766179206375612062616e203130302c3030302c303030766e642e"); // hex encode for "Toi da vay cua ban 100,000,000vnd."

//Nhận xét: Chuyển sang hệ thập lục phân chỉ khác đoạn "35302" thành "3130302" và khi được mã hóa thì M khác nhau hoàn toàn

// encrypt M: M^d mod n

BN\_mod\_exp(C1, M1, d, n, ctx);

BN\_mod\_exp(C2, M2, d, n, ctx);

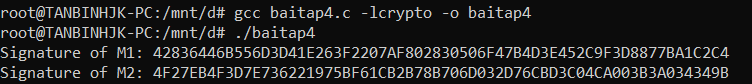
printBN("Signature of M1:", C1);

printBN("Signature of M2:", C2);

return 0;

}

* **Kết quả chạy:**



**Bài 5. Xác minh chữ ký**

* **Code:**

#include <stdio.h>

#include <openssl/bn.h>

void printBN(char \*msg, BIGNUM \*a)

{

char \*number\_str\_a = BN\_bn2hex(a);

printf("%s %s\n", msg, number\_str\_a);

OPENSSL\_free(number\_str\_a);

}

int main()

{

// khởi tạo

BN\_CTX \*ctx = BN\_CTX\_new();

BIGNUM \*n = BN\_new();

BIGNUM \*e = BN\_new();

BIGNUM \*M = BN\_new();

BIGNUM \*C = BN\_new();

BIGNUM \*S = BN\_new();

// gán giá trị

BN\_hex2bn(&n, "AE1CD4DC432798D933779FBD46C6E1247F0CF1233595113AA51B450F18116115");

BN\_dec2bn(&e, "65537");

//python3 -c 'print("Launch a missile.".encode().hex())'

BN\_hex2bn(&M, "4c61756e63682061206d697373696c652e"); //hex encode for " Launch a missile."

BN\_hex2bn(&S, "643D6F34902D9C7EC90CB0B2BCA36C47FA37165C0005CAB026C0542CBDB6802F");

// Get S^e mod: if S=M^d mod n, C=M

BN\_mod\_exp(C, S, e, n, ctx);

// Kiểm tra signature

if (BN\_cmp(C, M) == 0)

{

printf("Valid Signature! \n");

}

else

{

printf("Verification fails! \n");

}

return 0;

}

* **Kết quả chạy:**



*Hình a. Kết quả trước khi giá trị S đúng*

**



*Hình b. Kết quả sau khi thay đổi giá trị S*