Task 2: Encrypting a Message

จากไฟล์ Lab เราจะได้ hex ของ A top secret (message) ดังนี้ โดยจะนำไปใช้ใน ชุดคำสั่งต่อไป

```
$ python  -c 'print("A top secret!".encode("hex"))'
4120746f702073656372657421
```

2.2.1. ชุดคำสั่งทั้งภาษาซีและภาษาไพธอนที่นักศึกษาใช้ในการทำงานตามข้อกำหนด

```
#include <stdio.h>

#include <openssl/bn.h>

#define NBITS 256

//ปริ้นท์ BIGNUMBER นั่นคือผลลัพธ์ที่ได้

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

char * number_str = BN_bn2hex(a);

printf("%s %s\n", msg, number_str);

OPENSSL_free(number_str);

}

int main(){ //ประกาศตัวแปรที่จะใช้คำนวณใน main

    BN_CTX *ctx = BN_CTX_new();

    BIGNUM *m = BN_new();

    BIGNUM *e = BN_new();

    BIGNUM *n = BN_new();

    BIGNUM *d = BN_new();

    BIGNUM *enc = BN_new(); //for encrypt

    BIGNUM *dec = BN_new(); //for decrypt

BN_hex2bn(&e,"010001"); // แปลง Hexadecimal to BIGNUM และเก็บไว้ที่ตัวแปร e
```

```
BN_hex2bn(&n,"DCBFFE3E51F62E09CE7032E2677A78946A849DC4CDDE3A4D0CB81629242FB1
A5"); // แปลง Hexadecimal n ไป BIGNUM และเก็บไว้ที่ ตัวแปร n

BN_hex2bn(&m,"4120746f702073656372657421");     // message "A top secret!"

BN_hex2bn(&d,"74D806F9F3A62BAE331FFE3F0A68AFE35B3D2E4794148AACBC26AA381CD7D3
0D"); //key สำหรับ การเช็คว่าผลลัพธ์ถูกไหม


BN_mod_exp(enc, m, e, n, ctx);  // encrypt ด้วย m^e mod n

printBN("encrypt message = ", enc);


BN_mod_exp(dec, enc, d, n, ctx); // decrypt ด้วย enc^d mod n

if (BN_cmp(dec, m) == 0) {    // สำหรับปริ้นท์ข้อความบอกว่าสำเร็จหรือไม่

        printf("Encryption is successful.\n");

    } else {

        printf("Encryption failed.\n");

    }

    BN_free(m); //free memories

    BN_free(e);

    BN_free(n);

    BN_free(d);

    BN_free(enc);

    BN_free(dec);

    BN_CTX_free(ctx);

    return 0;
```

}

2.2.2. ภาพหน้าจอแสดงผลลัพธ์ของการทำงาน

```
rew@LAPTOP-F65IQ8R2:~/Lab1$ ./t2
encrypt message =  6FB078DA550B2650832661E14F4F8D2CFAEF475A0DF3A75CACDC5DE5CFC5FADC
Encryption is successful.
```

2.2.3. อภิปราย ผลลัพธ์ / สิ่งที่สังเกตได้

เราได้มีการใช้ค่า hexadecimal ของ Message และ ค่า public key, private key , n , e ที่ให้มาเข้าไปคำนวณรหัส Encryption ใน BN_mod_exp(enc, m, e, n, ctx) และได้ผลลัพธ์ออกมาเป็น 6FB078DA550B2650832661E14F4F8D2CFAEF475A0DF3A75CACDC5DE5CFC5FADC

โดยเราสามารถตรวจสอบความถูกต้องได้โดยใช้ฟังก์ชั่น BN_mod_exp(dec, enc, d, n, ctx) ในการ decrypt และใช้

if (BN_cmp(dec, m) == 0) ในการตรวจสอบว่าหากส่งคืน 0 ถ้า BIGNUM สองตัวเท่ากัน ดังนั้น == 0 ใช้เพื่อตรวจสอบว่าผลลัพธ์ของ BN_cmp บ่งชี้ความเท่ากันระหว่าง BIGNUM ทั้งสอง (dec และ m) หรือไม่ ถ้าผลลัพธ์เป็น 0 แสดงว่าเท่ากัน จึงปริ้นท์ Encryption is successful