

ITSEC

LAB REPORT: 2



BY – AMANUEL LEGESSE

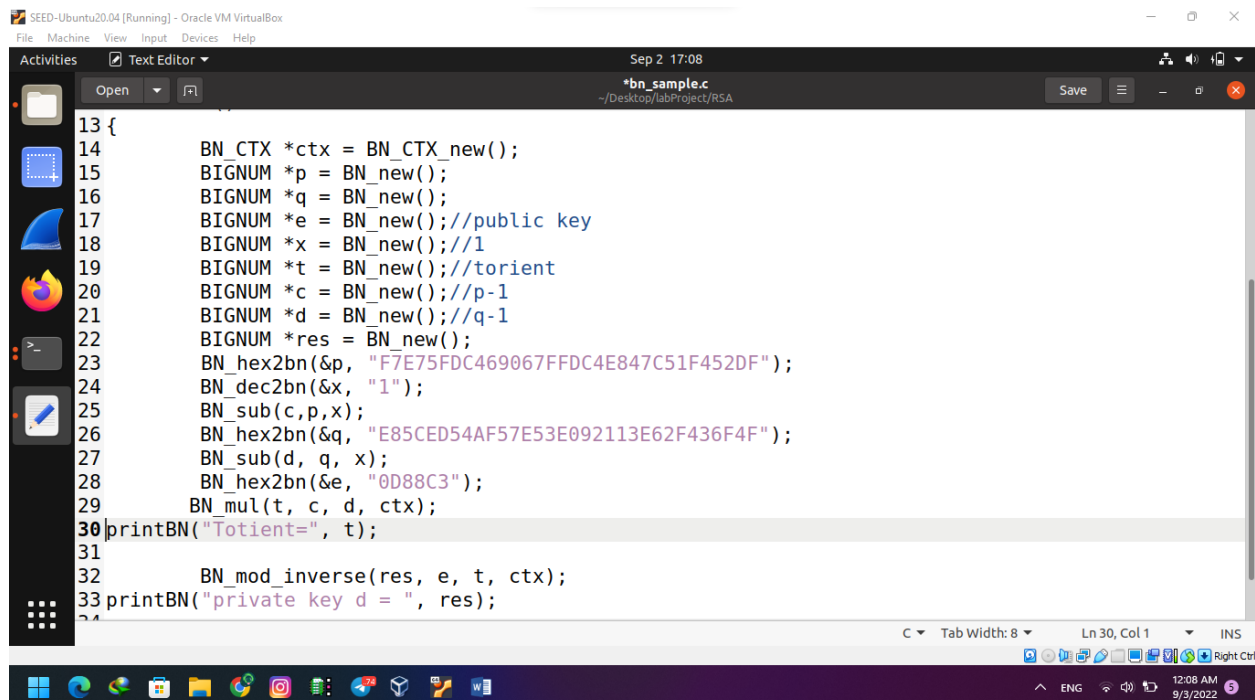
SOFTWARE ENG. EXT

ID – ATE/3628/1

RSA Encryption and Signature Lab

Task -1 Deriving the Private Key

- ✓ I compiled the program, bn_sample.c, given in the project description. I ran “ gcc bn_sample.c -lcrypto ” to compile the program using the crypto library. The following was the result from running the program.



The screenshot shows a terminal window titled "SEED-Ubuntu20.04 [Running] - Oracle VM VirtualBox". The window contains a C program named "bn_sample.c" which is being executed. The program's output is as follows:

```
13 {
14     BN_CTX *ctx = BN_CTX_new();
15     BIGNUM *p = BN_new();
16     BIGNUM *q = BN_new();
17     BIGNUM *e = BN_new();//public key
18     BIGNUM *x = BN_new();//1
19     BIGNUM *t = BN_new();//torient
20     BIGNUM *c = BN_new();//p-1
21     BIGNUM *d = BN_new();//q-1
22     BIGNUM *res = BN_new();
23     BN_hex2bn(&p, "F7E75FDC469067FFDC4E847C51F452DF");
24     BN_dec2bn(&x, "1");
25     BN_sub(c,p,x);
26     BN_hex2bn(&q, "E85CED54AF57E53E092113E62F436F4F");
27     BN_sub(d, q, x);
28     BN_hex2bn(&e, "0D88C3");
29     BN_mul(t, c, d, ctx);
30 printBN("Totient=", t);
31
32     BN_mod_inverse(res, e, t, ctx);
33 printBN("private key d = ", res);
34 }
```

The output of the program is:

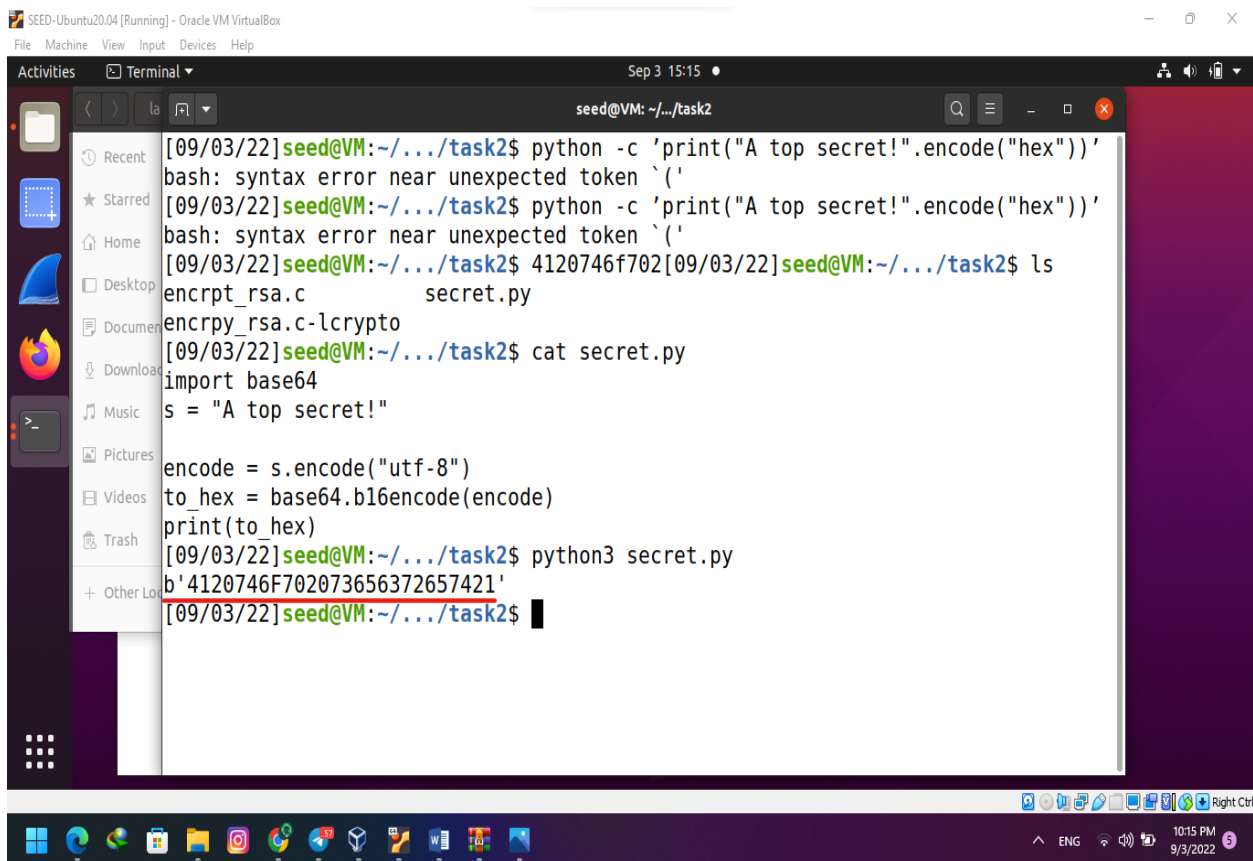
```
Totient= 1048576
private key d = 1048576
```

- I calculated n by multiplying $p * q$ and To find $\phi(n)$, I calculated $(p - 1)*(q - 1)$.

[illegible]

Task -2 Encrypting a Message

- ✓ In order to encrypt the message “A top secret!” first I converted it into hex string then to a BIGNUM using hex-to-bn. Using s python command



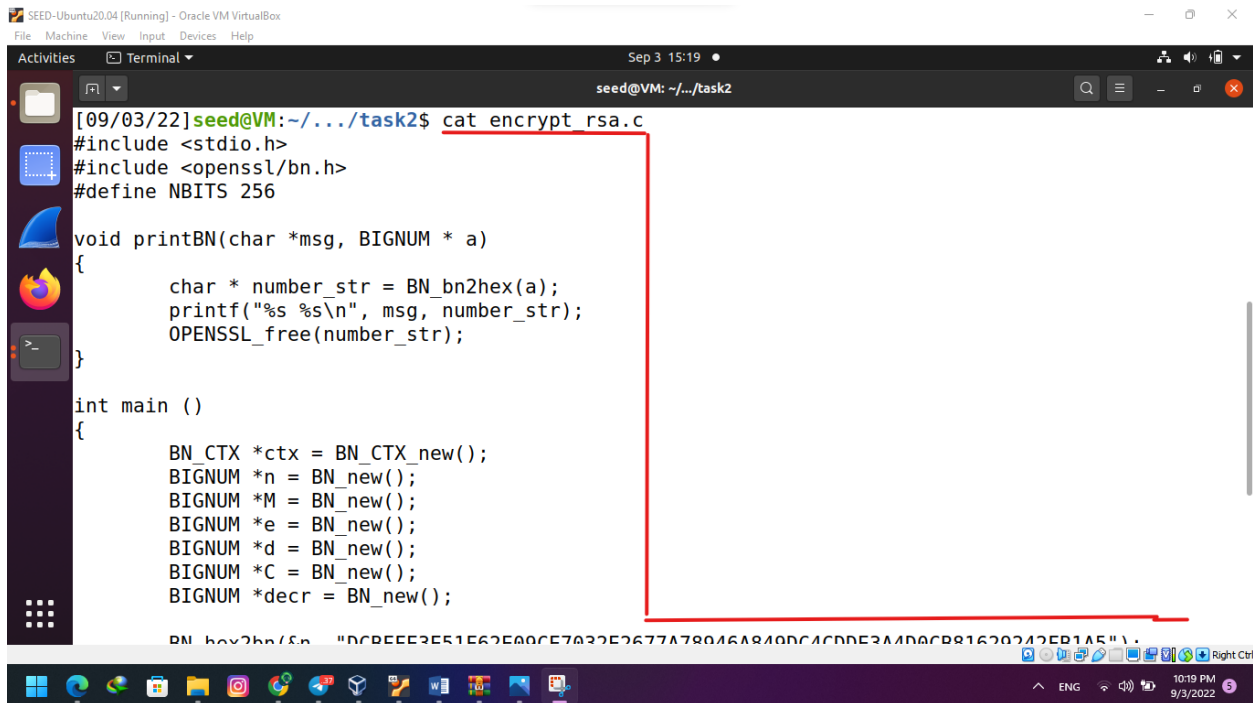
The screenshot shows a terminal window titled 'seed@VM: ~/.../task2' with a date and time of 'Sep 3 15:15'. The terminal output is as follows:

```
[09/03/22]seed@VM:~/.../task2$ python -c 'print("A top secret!".encode("hex"))'
bash: syntax error near unexpected token `('
[09/03/22]seed@VM:~/.../task2$ python -c 'print("A top secret!".encode("hex"))'
bash: syntax error near unexpected token `('
[09/03/22]seed@VM:~/.../task2$ 4120746f702[09/03/22]seed@VM:~/.../task2$ ls
encrpt_rsa.c          secret.py
encrpy_rsa.c-lcrypto
[09/03/22]seed@VM:~/.../task2$ cat secret.py
import base64
s = "A top secret!"

encode = s.encode("utf-8")
to_hex = base64.b16encode(encode)
print(to_hex)
[09/03/22]seed@VM:~/.../task2$ python3 secret.py
b'4120746F702073656372657421'
```

The final output, `b'4120746F702073656372657421'`, is highlighted with a red line. The terminal window is part of an Oracle VM VirtualBox environment, with a file manager sidebar on the left and a taskbar at the bottom.

➤ Next I compiled the encryption to C.



```
[09/03/22]seed@VM: ~/.../task2$ cat encrypt_rsa.c
#include <stdio.h>
#include <openssl/bn.h>
#define NBITS 256

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 ()
{
    BN_CTX *ctx = BN_CTX_new();
    BIGNUM *n = BN_new();
    BIGNUM *M = BN_new();
    BIGNUM *e = BN_new();
    BIGNUM *d = BN_new();
    BIGNUM *C = BN_new();
    BIGNUM *decr = BN_new();

    BN_hex2bn(&n, "DCBEEF3E51E62E90CE7A32E2677A78046A840DC4CDD53A4DA0CB81620242EB1A5");
```

✓ After running the C program encrpt_rsa.c I got the encrypted and decrypted message in hexDecimal.

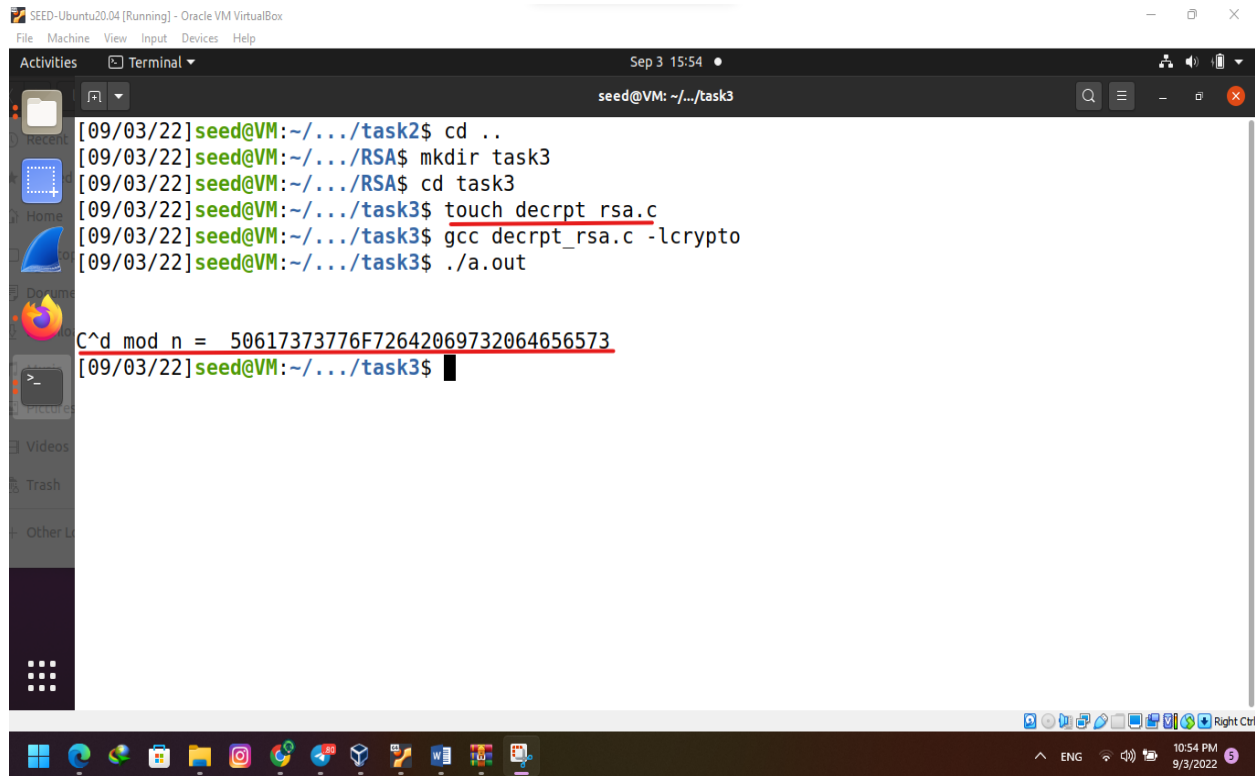
```
SEED-Ubuntu20.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Sep 3 15:46 seed@VM: ~/.../task2

/usr/bin/ld: /tmp/ccvSyaU2.o:encrpt_rsa.c:(.text+0xad): more undefined references to `BN_new' follow
/usr/bin/ld: /tmp/ccvSyaU2.o: in function `main':
encrpt_rsa.c:(.text+0xc4): undefined reference to `BN_hex2bn'
/usr/bin/ld: encrpt_rsa.c:(.text+0xd7): undefined reference to `BN_hex2bn'
/usr/bin/ld: encrpt_rsa.c:(.text+0xea): undefined reference to `BN_hex2bn'
/usr/bin/ld: encrpt_rsa.c:(.text+0xfd): undefined reference to `BN_hex2bn'
/usr/bin/ld: encrpt_rsa.c:(.text+0x11c): undefined reference to `BN_mod_exp'
/usr/bin/ld: encrpt_rsa.c:(.text+0x15a): undefined reference to `BN_mod_exp'
collect2: error: ld returned 1 exit status
[09/03/22] seed@VM:~/.../task2$ gcc encrpt_rsa.c -lcrypto
[09/03/22] seed@VM:~/.../task2$ ./a.out
M^e mod n = 6FB078DA550B2650832661E14F4F8D2CFAEF475A0DF3A75CACDC5DE5CFC5FADC

C^d mod n = 4120746F702073656372657421
[09/03/22] seed@VM:~/.../task2$
[09/03/22] seed@VM:~/.../task2$
[09/03/22] seed@VM:~/.../task2$
[09/03/22] seed@VM:~/.../task2$
[09/03/22] seed@VM:~/.../task2$
[09/03/22] seed@VM:~/.../task2$
```

Task 3: Decrypting a Message

- ✓ Previously I already implemented the decryption equation, I used this code to decrypt the ciphertext provided by creating a `decrpt_rsa.c` file with an out put if the decrypted number.



The screenshot shows a terminal window titled 'SEED-Ubuntu20.04 [Running] - Oracle VM VirtualBox'. The terminal output is as follows:

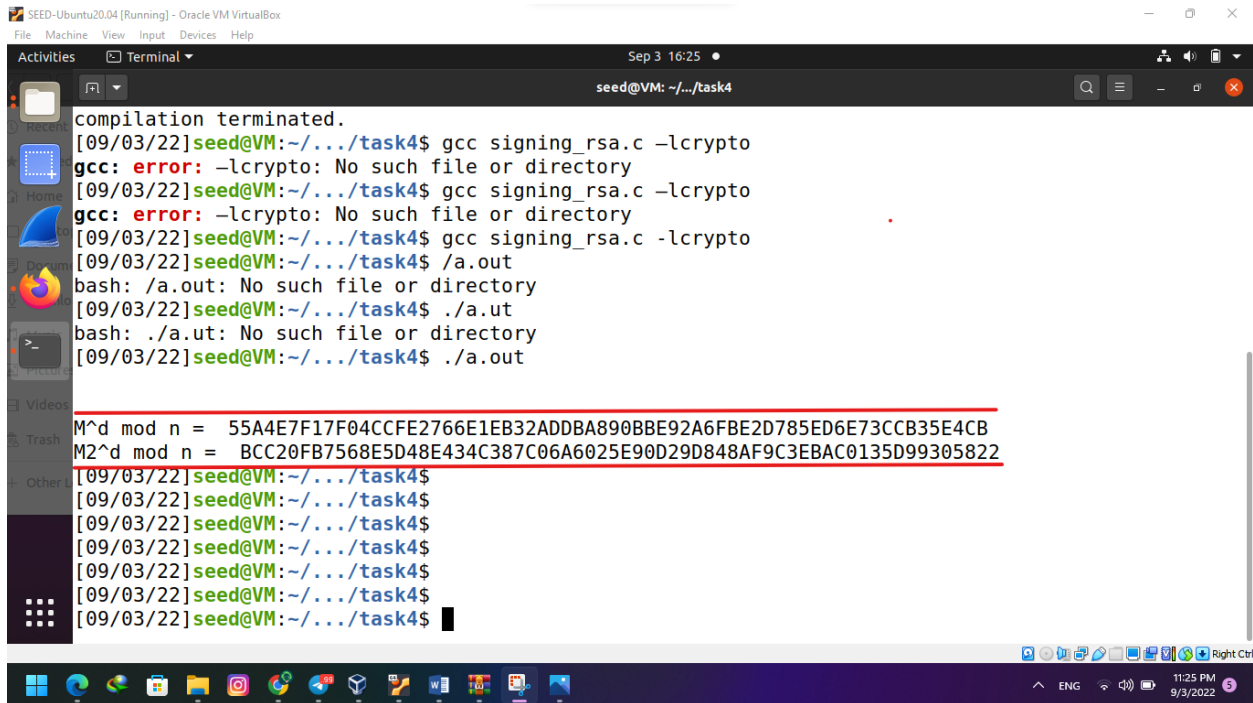
```
[09/03/22]seed@VM:~/.../task2$ cd ..
[09/03/22]seed@VM:~/.../RSA$ mkdir task3
[09/03/22]seed@VM:~/.../RSA$ cd task3
[09/03/22]seed@VM:~/.../task3$ touch decrpt rsa.c
[09/03/22]seed@VM:~/.../task3$ gcc decrpt_rsa.c -lcrypto
[09/03/22]seed@VM:~/.../task3$ ./a.out

C^d mod n = 50617373776F72642069732064656573
[09/03/22]seed@VM:~/.../task3$
```

The line C^d mod n = 50617373776F72642069732064656573 is underlined in red in the original image. The terminal window has a sidebar on the left with icons for Recent, Home, Documents, Videos, Trash, and Other Locations. The bottom of the window shows a taskbar with various application icons and a system tray with the date 9/3/2022 and time 10:54 PM.

Task 4: Signing a Message

- ✓ Created 2 python scripts with “I owe you \$2000” and “I owe you \$3000 to sign and change it from \$2000 to \$300 by using a compiled C program signing_rsa.c. The result is as shown below.

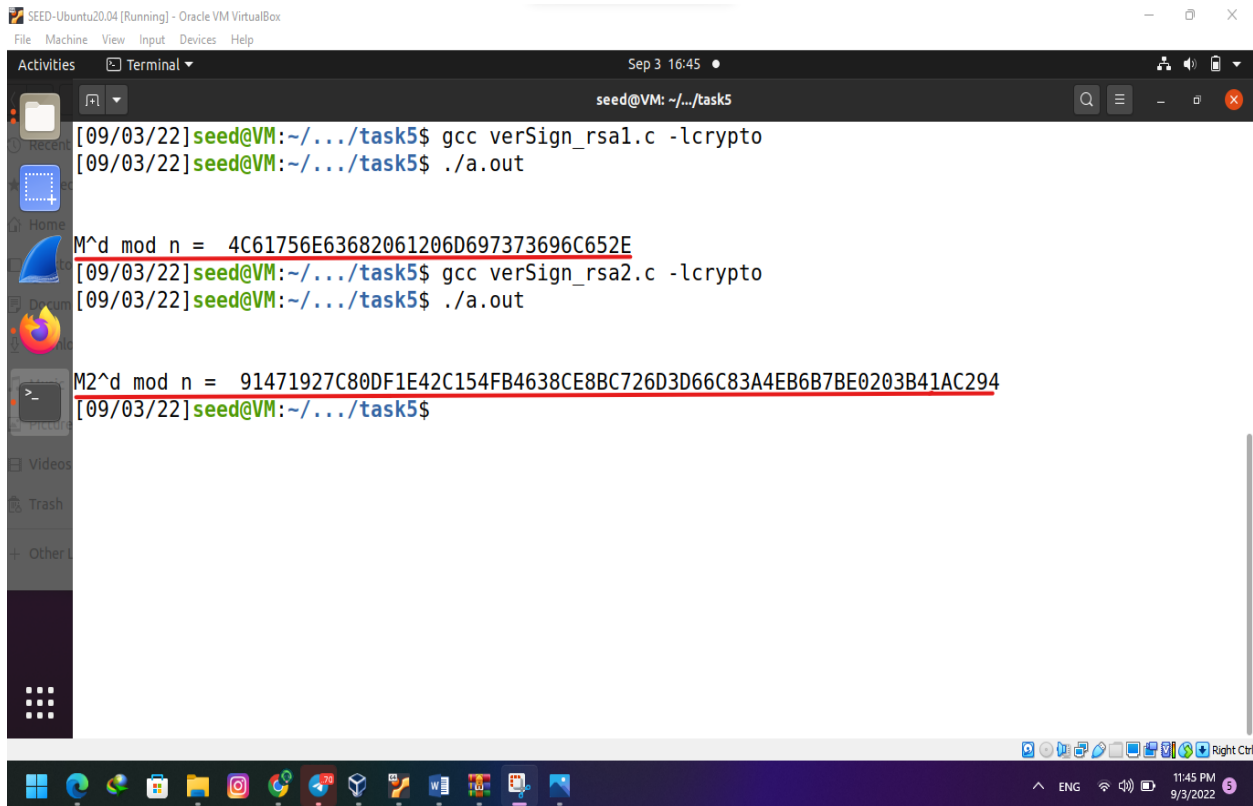


```
SEED-Ubuntu20.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Sep 3 16:25
seed@VM: ~/.../task4
compilation terminated.
[09/03/22]seed@VM:~/.../task4$ gcc signing_rsa.c -lcrypto
gcc: error: -lcrypto: No such file or directory
[09/03/22]seed@VM:~/.../task4$ gcc signing_rsa.c -lcrypto
gcc: error: -lcrypto: No such file or directory
[09/03/22]seed@VM:~/.../task4$ gcc signing_rsa.c -lcrypto
[09/03/22]seed@VM:~/.../task4$ ./a.out
bash: ./a.out: No such file or directory
[09/03/22]seed@VM:~/.../task4$ ./a.out
bash: ./a.out: No such file or directory
[09/03/22]seed@VM:~/.../task4$ ./a.out
[09/03/22]seed@VM:~/.../task4$
M^d mod n = 55A4E7F17F04CCFE2766E1EB32ADDBA890BBE92A6FBE2D785ED6E73CCB35E4CB
M2^d mod n = BCC20FB7568E5D48E434C387C06A6025E90D29D848AF9C3EBAC0135D99305822
[09/03/22]seed@VM:~/.../task4$
[09/03/22]seed@VM:~/.../task4$
[09/03/22]seed@VM:~/.../task4$
[09/03/22]seed@VM:~/.../task4$
[09/03/22]seed@VM:~/.../task4$
[09/03/22]seed@VM:~/.../task4$
[09/03/22]seed@VM:~/.../task4$
```

- ✓ I converted the message to hexadecimal and ran the digital signature equation on it.
- The original hexadecimal message encodings only differed by one byte but, the digital signatures of the two messages are different.

Task 5: Verifying a Signature

- ✓ To verify that a signature is the correct message, I used the following formula to get the hexadecimal representation of the message: $\text{Signature}^e \bmod n$. My program returned the hex string.



```
SEED-Ubuntu20.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Sep 3 16:45
seed@VM: ~/.../task5
[09/03/22]seed@VM:~/.../task5$ gcc verSign_rsa1.c -lcrypto
[09/03/22]seed@VM:~/.../task5$ ./a.out
M^d mod n = 4C61756E63682061206D697373696C652E
[09/03/22]seed@VM:~/.../task5$ gcc verSign_rsa2.c -lcrypto
[09/03/22]seed@VM:~/.../task5$ ./a.out
M2^d mod n = 91471927C80DF1E42C154FB4638CE8BC726D3D66C83A4EB6B7BE0203B41AC294
[09/03/22]seed@VM:~/.../task5$
```

- After I compiled the C program receiving the first output I modified the signature from 2F to 3F and ran the program again. I got the second hex string displayed above.