Task 5 Verifying a Signature

หา hex value ของ Launch a missile. โดยคำสั่ง

```
[02/29/24]seed@VM:~$ python3 -c 'print(bytes("Launch a missile.", "utf-8").hex())'
4c61756e63682061206d697373696c652e
```

Source code ที่ใช้ในการ Verify signature

```
#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()
{

   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();

   BN_hex2bn(&n,
"AE1CD4DC432798D933779FBD46C6E1247F0CF1233595113AA51B450F18116115");
```

```
   BN_dec2bn(&e, "65537");
   BN_hex2bn(&M, "4c61756e63682061206d697373696c652e");  //hex encode for "Launch a missile.
   BN_hex2bn(&S, "643D6F34902D9C7EC90CB0B2BCA36C47FA37165C0005CAB026C0542CBDB6802F");


   BN_mod_exp(C, S, e, n, ctx);
   printBN("Original Message : ", M);
   printBN("Computed value : ", C);


   if (BN_cmp(C, M) == 0)
   {
      printf("Valid Signature! \n");
   }


    else
   {
      printf("Invalid Signature! \n");
   }


   return 0;
}
```

ทำการ compile source code

```
[02/29/24]seed@VM:~$ gcc -o task5 task5.c -lcrypto
[02/29/24]seed@VM:~$ ./task5
Original Message :  4C61756E63682061206D697373696C652E
Computed value :  4C61756E63682061206D697373696C652E
Valid Signature!
```

จากผลลัพธ์ที่ได้พบว่า ข้อความต้นฉบับและข้อความที่เข้ารหัสด้วย public key นั้นตรงกัน จะได้ว่า Signature นี้ valid

สมมติว่า signature ถูก corrupted โดยเปลี่ยน byte สุดท้ายจาก 2F เป็น 3F

BN_hex2bn(&S,
"643D6F34902D9C7EC90CB0B2BCA36C47FA37165C0005CAB026C0542CBDB6803F");

และทำการ Compile source code ใหม่

```
[02/29/24]seed@VM:~$ gcc -o task5 task5.c -lcrypto
[02/29/24]seed@VM:~$ ./task5
Original Message :  4C61756E636820612061206D697373696C652E
Computed value :  91471927C80DF1E42C154FB4638CE8BC726D3D66C83A4EB6B7BE0203B41AC294
Invalid Signature!
```

ผลลัพธ์ที่ออกมานั้นแตกต่างจาก Original message อย่างสิ้นเชิงแม้ signature จะถูกเปลี่ยนแค่ byte เดียว และส่งผลให้

การ Verification ล้มเหลว