Darrel Chang
Tingting Chen
CS 4600
3/29/24

Homework 3

Task 1

homework2.c

```c
int main()
{
    /* Task 1*/
    BN_CTX *ctx = BN_CTX_new();

    BIGNUM *p = BN_new();
    BIGNUM *q = BN_new();
    BIGNUM *e = BN_new();

    BIGNUM *phiN = BN_new();
    BIGNUM *d = BN_new();

    BN_hex2bn(&p, "F7E75FDC469067FFDC4E847C51F452DF");
    BN_hex2bn(&q, "E85CED54AF57E53E092113E62F436F4F");
    BN_hex2bn(&e, "0D88C3");

    // Calculate phi n = (p-1)(q-1)
    BIGNUM *p_minus_1 = BN_new();
    BIGNUM *q_minus_1 = BN_new();

    BN_sub(p_minus_1, p, BN_value_one());
    BN_sub(q_minus_1, q, BN_value_one());
    BN_mul(phiN, p_minus_1, q_minus_1, ctx);

    // Calculate mod inverse of d = e mod phi
    BN_mod_inverse(d, e, phiN, ctx);

    printBN("phi N = ", phiN);
    printBN("e = ", e);
    printBN("d = ", d);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\darre\Desktop\CPP\CS 4800 Cryptography> gcc homework2.c -o task1 -lcrypto
PS C:\Users\darre\Desktop\CPP\CS 4800 Cryptography> ./task1
phi N =  E103ABD94892E3E74AFD724BF28E78348D52298BD687C44DEB3A81065A7981A4
e =  0D88C3
d =  3587A24598E5F2A21DB007D89D18CC50ABA5075BA19A33890FE7C28A9B496AEB
PS C:\Users\darre\Desktop\CPP\CS 4800 Cryptography>
```

After calculating p-1 and q-1 in order to get the toltien function of n, which is phi of n, We find the mod inverse of e and phi(N) to get our private key D =
0x3587A24598E5F2A21DB007D89D18CC50ABA5075BA19A33890FE7C28A9B496AEB

Task 2

homework2.c

```c
int main()
{
    /* Task 2*/
    BN_CTX *ctx = BN_CTX_new();

    BIGNUM *n = BN_new();
    BIGNUM *e = BN_new();
    BIGNUM *m = BN_new();
    BIGNUM *d = BN_new();
    BIGNUM *c = BN_new();



    BN_hex2bn(&n, "DCBFFE3E51F62E09CE7032E2677A78946A849DC4CDDE3A4D0CB81629242FB1A5");
    BN_hex2bn(&e, "010001");
    // Message "A top secret!" in hex is "4120746f702073656372657421"
    BN_hex2bn(&m, "4120746f702073656372657421");
    BN_hex2bn(&d, "74D806F9F3A62BAE331FFE3F0A68AFE35B3D2E4794148AACBC26AA381CD7D30D");

    // C = M^e mod n
    BN_mod_exp(c, m, e, n, ctx);

    printBN("For m = ", m);
    printBN("e = ", e);
    printBN("and n = ", n);
    printBN("C = ", c);

    BN_mod_exp(c, c, d, n, ctx);

    printBN("C after it has been decrypted using d = ", c);

}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\darre\Desktop\CPP\CS 4800 Cryptography> gcc homework2.c -o task2 -lcrypto
PS C:\Users\darre\Desktop\CPP\CS 4800 Cryptography> ./task2
For m =  4120746F702073656372657421
e =  010001
and n =  DCBFFE3E51F62E09CE7032E2677A78946A849DC4CDDE3A4D0CB81629242FB1A5
C =  6FB078DA550B2650832661E14F4F8D2CFAEF475A0DF3A75CACDC5DE5CFC5FADC
C after it has been decrypted using d =  4120746F702073656372657421
PS C:\Users\darre\Desktop\CPP\CS 4800 Cryptography>
```

We convert the ascii string “A top secret!’ to hex, which is “4120746f702073656372657421”. We calculate m^e mod n to find our ciphertext C. To check if it was enciphered correctly, we decrypt it by calculating C^d mod n. The encryption was correct because C after being decrypted again is the same as original message

Task 3

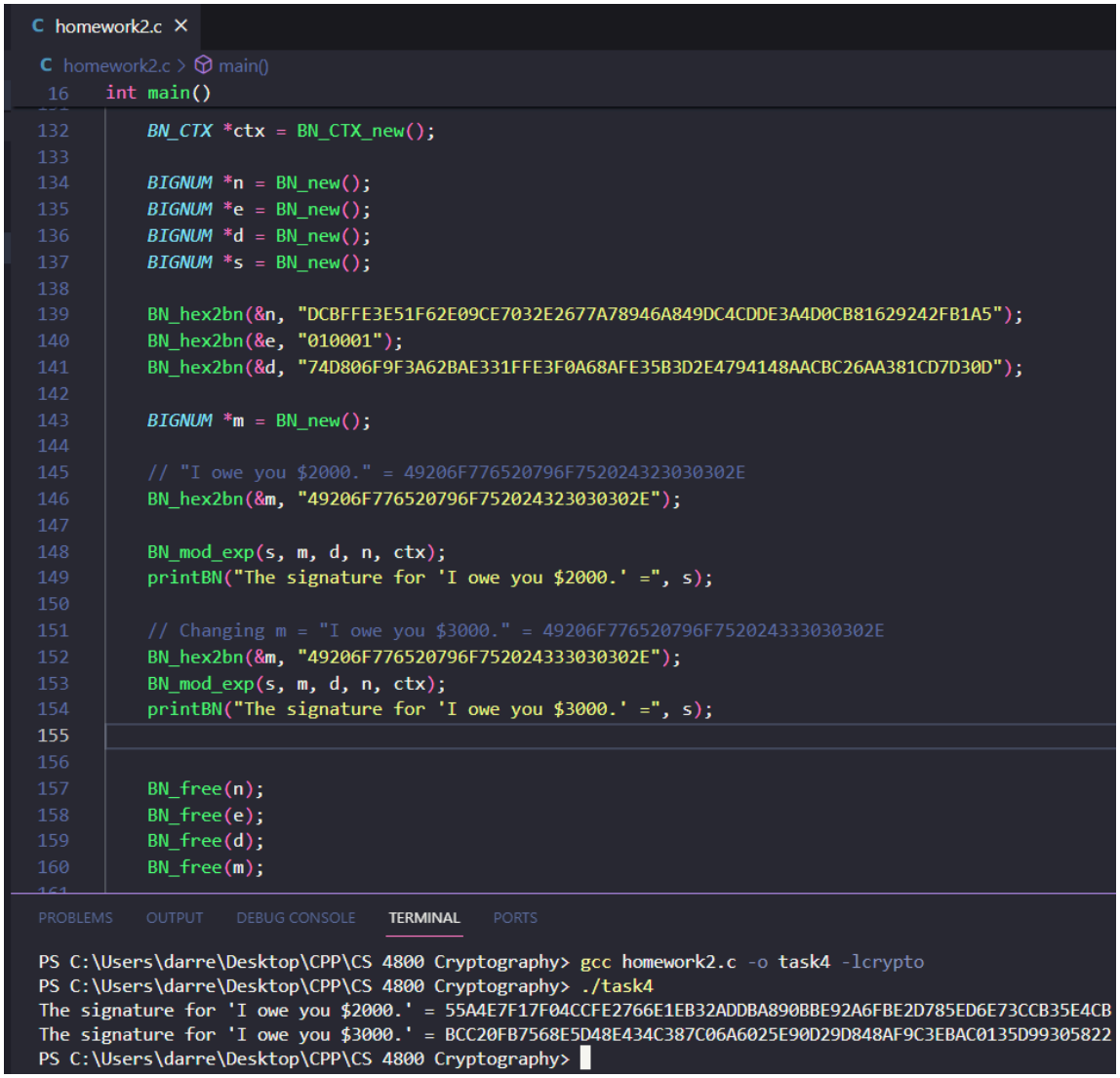
homework2.c

```c
int main()
{

    // Task 3 n, e, d same as task 2
    BN_CTX *ctx = BN_CTX_new();

    BIGNUM *n = BN_new();
    BIGNUM *e = BN_new();
    BIGNUM *d = BN_new();

    BN_hex2bn(&n, "DCBFFE3E51F62E09CE7032E2677A78946A849DC4CDDE3A4D0CB81629242FB1A5");
    BN_hex2bn(&e, "010001");
    BN_hex2bn(&d, "74D806F9F3A62BAE331FFE3F0A68AFE35B3D2E4794148AACBC26AA381CD7D30D");

    BIGNUM *c = BN_new();
    BIGNUM *m = BN_new();
    BN_hex2bn(&c, "8C0F971DF2F3672B28811407E2DABBE1DA0FEBBBDFC7DCB67396567EA1E2493F");

    // decoding c using M = C^d mod n
    BN_mod_exp(m, c, d, n, ctx);

    printBN("C =", c);
    printBN("d =", d);
    printBN("n =", n);
    printBN("Decoded message =", m);

    BN_free(n);
    BN_free(e);
    BN_free(d);
    BN_free(c);
    BN_free(m);
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\darre\Desktop\CPP\CS 4800 Cryptography> ./task3
C = 8C0F971DF2F3672B28811407E2DABBE1DA0FEBBBDFC7DCB67396567EA1E2493F
d = 74D806F9F3A62BAE331FFE3F0A68AFE35B3D2E4794148AACBC26AA381CD7D30D
n = DCBFFE3E51F62E09CE7032E2677A78946A849DC4CDDE3A4D0CB81629242FB1A5
Decoded message = 50617373776F72642069732064656573
PS C:\Users\darre\Desktop\CPP\CS 4800 Cryptography> python -c "print(bytes.fromhex('50617373776F72642069732064656573').decode('utf-8'))"
>>
Password is dees
PS C:\Users\darre\Desktop\CPP\CS 4800 Cryptography>
```

We found this by calculating m = c^d mod n to get the original message back in hex. Then we convert this hex value to ascii. M in hex 50617373776F72642069732064656573, which is “Password is dees”

Task 4

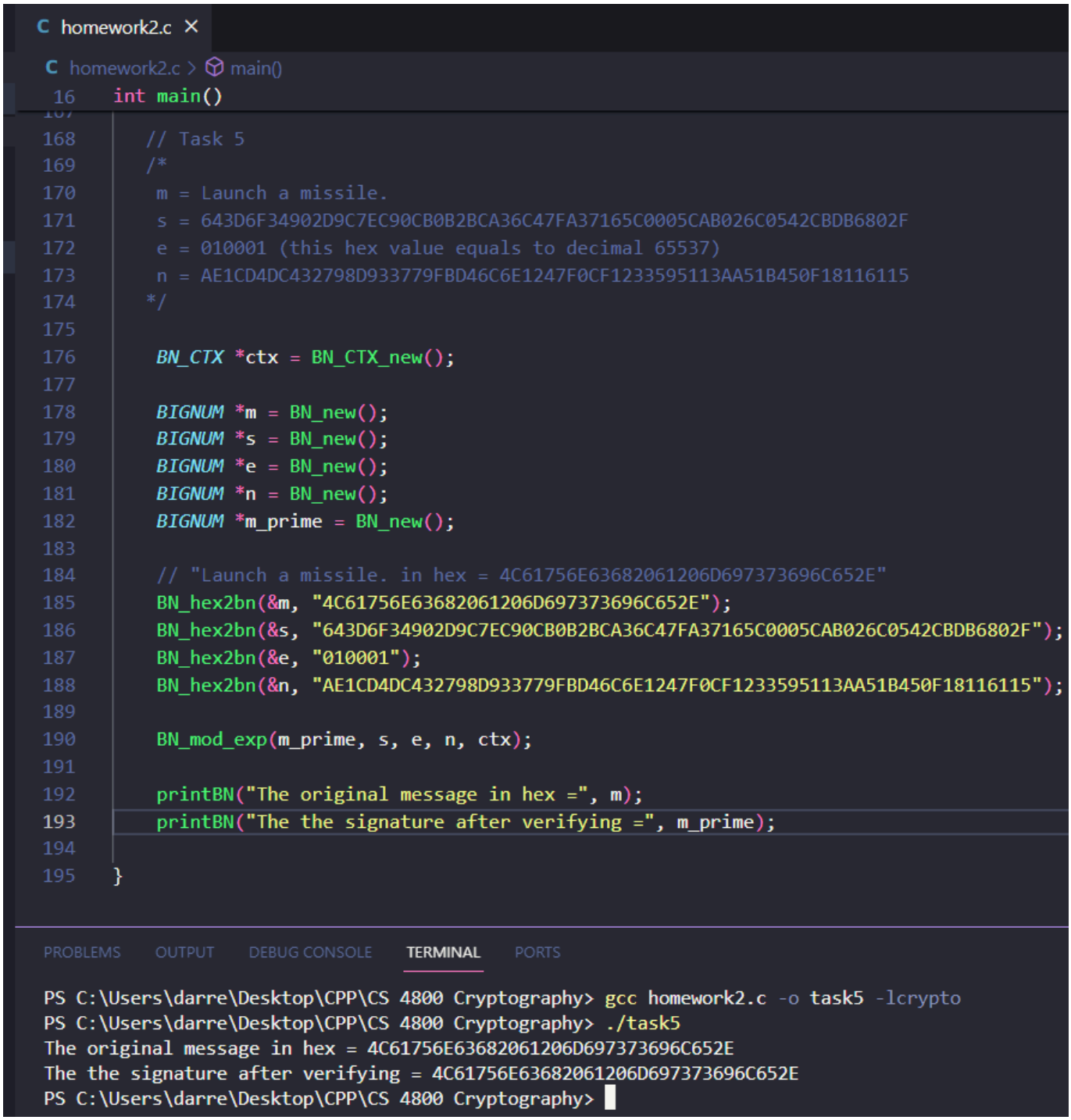


```
homework2.c
homework2.c > main()
16  int main()
132     BN_CTX *ctx = BN_CTX_new();
133
134     BIGNUM *n = BN_new();
135     BIGNUM *e = BN_new();
136     BIGNUM *d = BN_new();
137     BIGNUM *s = BN_new();
138
139     BN_hex2bn(&n, "DCBFFE3E51F62E09CE7032E2677A78946A849DC4CDDE3A4D0CB81629242FB1A5");
140     BN_hex2bn(&e, "010001");
141     BN_hex2bn(&d, "74D806F9F3A62BAE331FFE3F0A68AFE35B3D2E4794148AACBC26AA381CD7D30D");
142
143     BIGNUM *m = BN_new();
144
145     // "I owe you $2000." = 49206F776520796F752024323030302E
146     BN_hex2bn(&m, "49206F776520796F752024323030302E");
147
148     BN_mod_exp(s, m, d, n, ctx);
149     printBN("The signature for 'I owe you $2000.' =", s);
150
151     // Changing m = "I owe you $3000." = 49206F776520796F752024333030302E
152     BN_hex2bn(&m, "49206F776520796F752024333030302E");
153     BN_mod_exp(s, m, d, n, ctx);
154     printBN("The signature for 'I owe you $3000.' =", s);
155
156
157     BN_free(n);
158     BN_free(e);
159     BN_free(d);
160     BN_free(m);

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\darre\Desktop\CPP\CS 4800 Cryptography> gcc homework2.c -o task4 -lcrypto
PS C:\Users\darre\Desktop\CPP\CS 4800 Cryptography> ./task4
The signature for 'I owe you $2000.' = 55A4E7F17F04CCFE2766E1EB32ADDBA890BBE92A6FBE2D785ED6E73CCB35E4CB
The signature for 'I owe you $3000.' = BCC20FB7568E5D48E434C387C06A6025E90D29D848AF9C3EBAC0135D99305822
PS C:\Users\darre\Desktop\CPP\CS 4800 Cryptography>
```

We found the signature by calculating s = m^d mod n, where s is our signature
The signature for 'I owe you $2000.' = 55A4E7F17F04CCFE2766E1EB32ADDBA890BBE92A6FBE2D785ED6E73CCB35E4CB

The signature for 'I owe you $3000.' = BCC20FB7568E5D48E434C387C06A6025E90D29D848AF9C3EBAC0135D99305822

Task 5



```c
// Task 5
/*
 m = Launch a missile.
 s = 643D6F34902D9C7EC90CB0B2BCA36C47FA37165C0005CAB026C0542CBDB6802F
 e = 010001 (this hex value equals to decimal 65537)
 n = AE1CD4DC432798D933779FBD46C6E1247F0CF1233595113AA51B450F18116115
*/

BN_CTX *ctx = BN_CTX_new();

BIGNUM *m = BN_new();
BIGNUM *s = BN_new();
BIGNUM *e = BN_new();
BIGNUM *n = BN_new();
BIGNUM *m_prime = BN_new();

// "Launch a missile. in hex = 4C61756E63682061206D697373696C652E"
BN_hex2bn(&m, "4C61756E63682061206D697373696C652E");
BN_hex2bn(&s, "643D6F34902D9C7EC90CB0B2BCA36C47FA37165C0005CAB026C0542CBDB6802F");
BN_hex2bn(&e, "010001");
BN_hex2bn(&n, "AE1CD4DC432798D933779FBD46C6E1247F0CF1233595113AA51B450F18116115");

BN_mod_exp(m_prime, s, e, n, ctx);

printBN("The original message in hex =", m);
printBN("The the signature after verifying =", m_prime);

}
```

```
PS C:\Users\darre\Desktop\CPP\CS 4800 Cryptography> gcc homework2.c -o task5 -lcrypto
PS C:\Users\darre\Desktop\CPP\CS 4800 Cryptography> ./task5
The original message in hex = 4C61756E63682061206D697373696C652E
The the signature after verifying = 4C61756E63682061206D697373696C652E
PS C:\Users\darre\Desktop\CPP\CS 4800 Cryptography>
```

We verify the signature by ensuring the original message is the same as the verified signature. To verify the signature we calculate s_verified = s^e mod n, which should be the same as the original message m

The original hex message is the same as the signature after verifying.

= 4C61756E63682061206D697373696C652E