

1. Δραστηριότητα 1

Τεκμηρίωση

Για την τιμή p επιλέξαμε "953AAB9B3F23ED593FBDC690CA10E703". Για την τιμή q επιλέξαμε "C34EFC7C4C2369164E953553CDF94945". Για την τιμή e (δημόσιο κλειδί) επιλέξαμε "0D88C3". Για να βρούμε το d (ιδιωτικό κλειδί) πρέπει να χρησιμοποιήσουμε τον τύπο:

$$e \cdot \text{mod } (p-1)(q-1) = 1$$

Για να γίνει αυτό θα υπολογίσουμε τα $p-1$ και $q-1$ καθώς και το $(p-1)(q-1)$. Για να μπορέσουμε να το κάνουμε αυτό αναθέτουμε στην τιμή i το 1 και εκτελούμε τις παρακάτω πράξεις:

```
BN_sub(res1, p, i);  
BN_sub(res2, q, i);  
BN_mul(res0, res1, res2, ctx);
```

Τέλος για να υπολογίσουμε το ιδιωτικό κλειδί εκτελούμε:

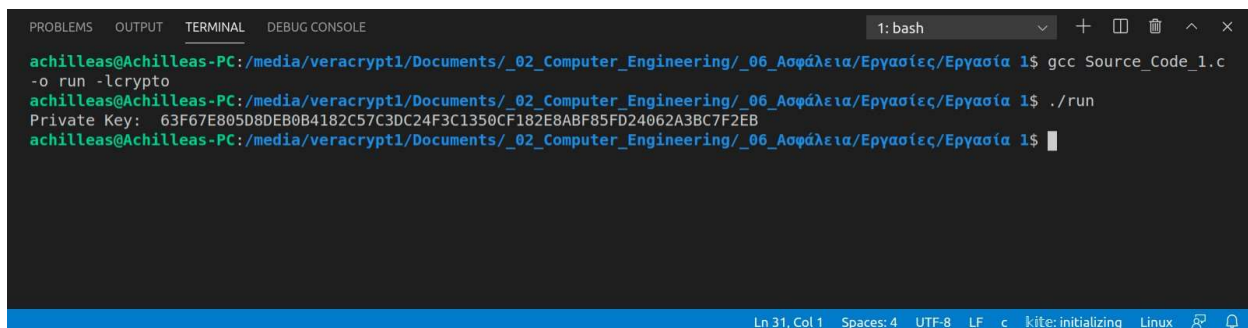
```
BN_mod_inverse(d, e, res0, ctx);
```

Για να εκτελέσουμε το προγράμμα, αρχικά το κάνουμε compile μέσω του terminal με την εντολή: `gcc Source_Code_1.c -o run -lcrypto`
Στην συνέχεια εκτελούμε το αρχείο με την εντολή `./run`

Αποτελέσματα:

Private Key:

63F67E805D8DEB0B4182C57C3DC24F3C1350CF182E8ABF85FD24062A3BC7F2EB



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: bash
achilleas@Achilleas-PC:/media/veracrypt1/Documents/_02_Computer_Engineering/_06_Ασφάλεια/Εργασίες/Εργασία 1$ gcc Source_Code_1.c
-o run -lcrypto
achilleas@Achilleas-PC:/media/veracrypt1/Documents/_02_Computer_Engineering/_06_Ασφάλεια/Εργασίες/Εργασία 1$ ./run
Private Key: 63F67E805D8DEB0B4182C57C3DC24F3C1350CF182E8ABF85FD24062A3BC7F2EB
achilleas@Achilleas-PC:/media/veracrypt1/Documents/_02_Computer_Engineering/_06_Ασφάλεια/Εργασίες/Εργασία 1$
```

Εικόνα 1. Μεταγλώττιση και εκτύπωση του ιδιωτικού κλειδιού.

Κώδικας

```
#include <stdio.h>
#include <openssl/bn.h>

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

int main() {
    BN_CTX *ctx = BN_CTX_new();
    BIGNUM *p = BN_new();
    BIGNUM *q = BN_new();
    BIGNUM *n = BN_new();
    BIGNUM *e = BN_new();
    BIGNUM *i = BN_new();
    BIGNUM *d = BN_new();
    BIGNUM *res0 = BN_new();
    BIGNUM *res1 = BN_new();
    BIGNUM *res2 = BN_new();

    BN_hex2bn(&p, "953AAB9B3F23ED593FBDC690CA10E703");
    BN_hex2bn(&q, "C34EFC7C4C2369164E953553CDF94945");
    BN_hex2bn(&e, "0D88C3");
    BN_hex2bn(&i, "1");

    /* Calculate (p-1)*(q-1) */
    BN_sub(res1, p, i);
    BN_sub(res2, q, i);
    BN_mul(res0, res1, res2, ctx);

    /* Calculate the private key */
    BN_mod_inverse(d, e, res0, ctx);

    /* Print values */
    printBN("Private Key: ", d);

    return 0;
}
```

```
/* Prints BIGNUM */  
void printBN(char *msg, BIGNUM * a) {  
    char * number_str = BN_bn2hex(a);  
    printf("%s %s\n", msg, number_str);  
    OPENSSL_free(number_str);  
}
```

2. Δραστηριότητα 2

Τεκμηρίωση

Για να συνεχίσουμε στην δεύτερη δραστηριότητα θα κάνουμε χρήση των προηγούμενων δημόσιων και ιδιωτικών κλειδιών. Το ονοματεπώνυμο που θα χρησιμοποιηθεί είναι “Achilleas Pappas” που έχει δεκαεξαδική τιμή ίση με “416368696c6c65617320506170706173” και το αναθέσαμε στην μεταβλητή M τύπου BIGNUM. Για να μπορέσουμε να το κρυπτογραφήσουμε κάναμε χρήση της:

```
BN_mod_exp(C, M, e, n, ctx);
```

Για την αποκρυπτογράφηση κάνουμε χρήση της:

```
BN_mod_exp(temp, C, d, n, ctx);
```

Τέλος το πρόγραμμα μας επιστρέφει αν το όνομα που υπέστει κρυπτογράφηση και αποκρυπτογράφηση είναι άθικτο με την χρήση της καθώς μας ενημερώνει για τις τιμές:

```
printBN("Private Key: ", d);
printBN("Message: ", M);
printBN("Encrypted: ", C);
printBN("Dencrypted: ", temp);

if(!BN_cmp(temp, M)){
    printf("Message is OK\n");
}
else {
    printf("Message is NOT OK\n");
}
```

Αποτελέσματα:

Private Key:

63F67E805D8DEB0B4182C57C3DC24F3C1350CF182E8ABF85FD24062A3BC7F2EB

Message:

416368696C6C65617320506170706173

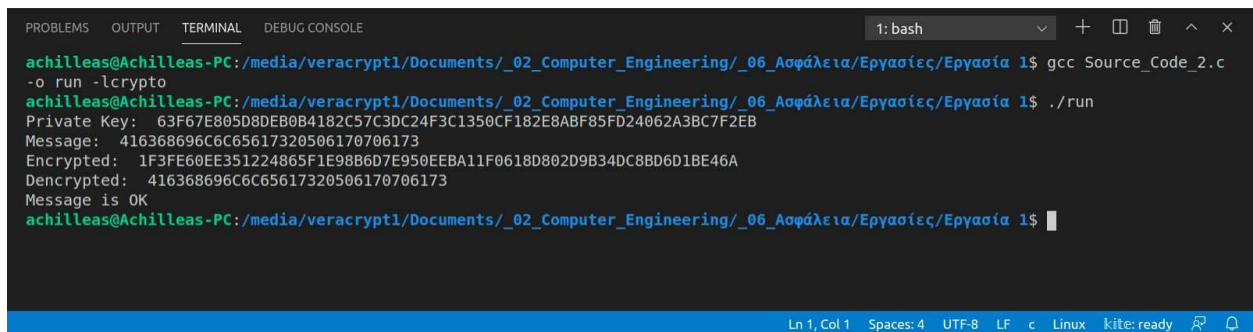
Encrypted:

1F3FE60EE351224865F1E98B6D7E950EEBA11F0618D802D9B34DC8BD6D1BE46A

Dencrypted:

416368696C6C65617320506170706173

Message is OK



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: bash
achilleas@Achilleas-PC:/media/veracrypt1/Documents/_02_Computer_Engineering/_06_Ασφάλεια/Εργασίες/Εργασία 1$ gcc Source_Code_2.c
-o run -lcrypto
achilleas@Achilleas-PC:/media/veracrypt1/Documents/_02_Computer_Engineering/_06_Ασφάλεια/Εργασίες/Εργασία 1$ ./run
Private Key: 63F67E805D8DEB0B4182C57C3DC24F3C1350CF182E8ABF85FD24062A3BC7F2EB
Message: 416368696C6C65617320506170706173
Encrypted: 1F3FE60EE351224865F1E98B6D7E950EEBA11F0618D802D9B34DC8BD6D1BE46A
Decrypted: 416368696C6C65617320506170706173
Message is OK
achilleas@Achilleas-PC:/media/veracrypt1/Documents/_02_Computer_Engineering/_06_Ασφάλεια/Εργασίες/Εργασία 1$
```

Εικόνα 2. Μεταγλώττιση και εκτέλεση.

Κώδικας

```
#include <stdio.h>
#include <openssl/bn.h>

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

int main() {
    BN_CTX *ctx = BN_CTX_new();
    BIGNUM *p = BN_new();
    BIGNUM *q = BN_new();
    BIGNUM *n = BN_new();
    BIGNUM *e = BN_new();
    BIGNUM *i = BN_new();
    BIGNUM *d = BN_new();
    BIGNUM *res0 = BN_new();
    BIGNUM *res1 = BN_new();
    BIGNUM *res2 = BN_new();
    BIGNUM *M = BN_new();
    BIGNUM *C = BN_new();
    BIGNUM *temp = BN_new();

    BN_hex2bn(&p, "953AAB9B3F23ED593FBDC690CA10E703");
    BN_hex2bn(&q, "C34EFC7C4C2369164E953553CDF94945");
    BN_hex2bn(&e, "0D88C3");
    BN_hex2bn(&i, "1");

    /* Calculate (p-1)*(q-1) */
    BN_sub(res1, p, i);
    BN_sub(res2, q, i);
```

```

    BN_mul(res0, res1, res2, ctx);
    /* Caclulate the private key */
    BN_mod_inverse(d, e, res0, ctx);

/* Calculate n */
    BN_mul(n, p, q, ctx);

/* Hex value of Achilleas Pappas */
    BN_hex2bn(&M, "416368696c6c65617320506170706173");
/* Encryption */
    BN_mod_exp(C, M, e, n, ctx);

/* Decryption */
    BN_mod_exp(temp, C, d, n, ctx);

/* Print values */
    printBN("Private Key: ", d);
    printBN("Message: ", M);
    printBN("Encrypted: ", C);
    printBN("Dencrypted: ", temp);

/* BN_cmp return 0 if the parameters are equal. */
    if(!BN_cmp(temp, M)){
        printf("Message is OK\n");
    }
    else {
        printf("Message is NOT OK\n");
    }
    return 0;
}

/* Prints BIGNUM */
void printBN(char *msg, BIGNUM * a) {
    char * number_str = BN_bn2hex(a);
    printf("%s %s\n", msg, number_str);
    OPENSSL_free(number_str);
}

```

3. Δραστηριότητα 3

Τεκμηρίωση

Για το n θέτουμε: “DCBFFE3E51F62E09CE7032E2677A78946A849DC4CDDE3A4D0CB81629242FB1A5”. Για το e (δημόσιο κλειδί) θέτουμε: “010001”. Για το d (ιδιωτικό κλειδί) θέτουμε: “74D806F9F3A62BAE331FFE3F0A68AFE35B3D2E4794148AACBC26AA381CD7D30D”. Το μήνυμα C (cipher) που πρέπει να αποκρυπτογραφήσουμε είναι το: “B3AF0A70793BB53492B5311AED5EA843D94661924C97A446E9DD75846DF860DF”

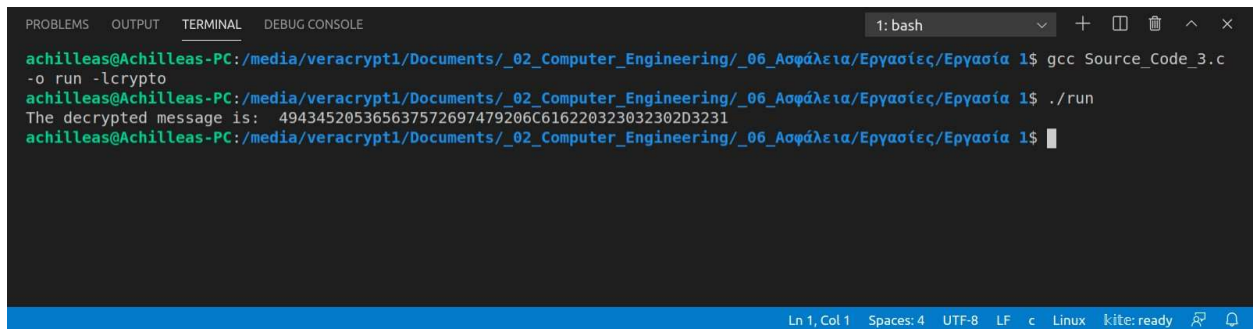
Κάνουμε λοιπόν χρήση της:

```
BN_mod_exp(temp, C, d, n, ctx);
```

Αποτελέσματα:

The decrypted message is:

494345205365637572697479206C616220323032302D3231



```
achilleas@Achilleas-PC:/media/veracrypt1/Documents/_02_Computer_Engineering/_06_Ασφάλεια/Εργασίες/Εργασία 1$ gcc Source_Code_3.c
-o run -lcrypto
achilleas@Achilleas-PC:/media/veracrypt1/Documents/_02_Computer_Engineering/_06_Ασφάλεια/Εργασίες/Εργασία 1$ ./run
The decrypted message is: 494345205365637572697479206C616220323032302D3231
achilleas@Achilleas-PC:/media/veracrypt1/Documents/_02_Computer_Engineering/_06_Ασφάλεια/Εργασίες/Εργασία 1$
```

Εικόνα 2. Μεταγλώττιση και έλεγχος του μηνύματος.

Κώδικας

```
#include <stdio.h>
#include <openssl/bn.h>

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

int main() {
    BN_CTX *ctx = BN_CTX_new();
    BIGNUM *n = BN_new();
    BIGNUM *e = BN_new();
    BIGNUM *d = BN_new();
    BIGNUM *C = BN_new();
    BIGNUM *temp = BN_new();

    BN_hex2bn(&n, "DCBFFE3E51F62E09CE7032E2677A78946A849DC4CDDE3A4D0CB81629242FB1A5");
    BN_hex2bn(&e, "010001");
    BN_hex2bn(&d, "74D806F9F3A62BAE331FFE3F0A68AFE35B3D2E4794148AACBC26AA381CD7D30D");
    BN_hex2bn(&C, "B3AF0A70793BB53492B5311AED5EA843D94661924C97A446E9DD75846DF860DF");

    /* Decrypt the cipher */
    BN_mod_exp(temp, C, d, n, ctx);

    /* Prints result */
    printBN("The decrypted message is: ", temp);

    return 0;
}

/* Prints BIGNUM */
void printBN(char *msg, BIGNUM * a) {
    char * number_str = BN_bn2hex(a);
    printf("%s %s\n", msg, number_str);
    OPENSSL_free(number_str);
}
```


4. Δραστηριότητα 4

Τεκμηρίωση

Στε αυτήν την δραστηριότητα κάνουμε χρήση του δημόσιου και ιδιωτικού κλειδιού απο την προηγούμενη δραστηριότητα. Ός μήνυμα θέτουμε το “I will be back!” και ως παραπονημένο μήνυμα το “I will be bk!”. Άρα:

M = “492077696c6c206265206261636b21”.

M_new = “492077696c6c2062652062616b21”.

Παρατηρούμε, λοιπόν, στα αποτελέσματα ότι οι δύο υπογραφές δεν ταυτίζονται, αναμενόμενο αφού τα τα μηνύματα που υπογράφηκαν ήταν διαφορετικά.

Αποτελέσματα:

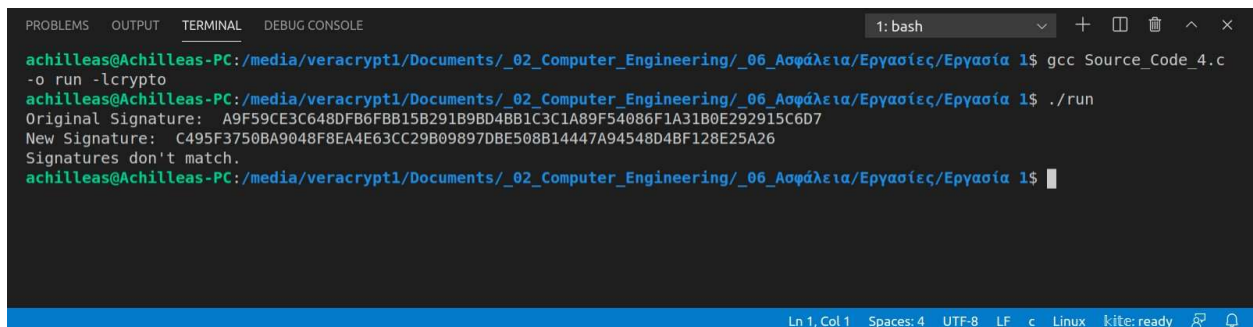
Original Signature:

A9F59CE3C648DFB6FBB15B291B9BD4BB1C3C1A89F54086F1A31B0E292915C6D7

New Signature:

C495F3750BA9048F8EA4E63CC29B09897DBE508B14447A94548D4BF128E25A26

Signatures don't match.



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: bash
achilleas@Achilleas-PC:/media/veracrypt1/Documents/_02_Computer_Engineering/_06_Ασφάλεια/Εργασίες/Εργασία 1$ gcc Source_Code_4.c
-o run -lcrypto
achilleas@Achilleas-PC:/media/veracrypt1/Documents/_02_Computer_Engineering/_06_Ασφάλεια/Εργασίες/Εργασία 1$ ./run
Original Signature: A9F59CE3C648DFB6FBB15B291B9BD4BB1C3C1A89F54086F1A31B0E292915C6D7
New Signature: C495F3750BA9048F8EA4E63CC29B09897DBE508B14447A94548D4BF128E25A26
Signatures don't match.
achilleas@Achilleas-PC:/media/veracrypt1/Documents/_02_Computer_Engineering/_06_Ασφάλεια/Εργασίες/Εργασία 1$
```

Εικόνα 4. Μεταγλώττιση και έλεγχος αν ταυτίζονται οι υπογραφές.

Κώδικας

```
#include <stdio.h>
#include <openssl/bn.h>

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

int main() {
    BN_CTX *ctx = BN_CTX_new();
    BIGNUM *n = BN_new();
    BIGNUM *e = BN_new();
    BIGNUM *d = BN_new();
    BIGNUM *M = BN_new();
    BIGNUM *M_new = BN_new();
    BIGNUM *S = BN_new();
    BIGNUM *S_new = BN_new();

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

    /* The hex value of "I will be back!" */
    BN_hex2bn(&M, "492077696c6c206265206261636b21");
    BN_mod_exp(S, M, d, n, ctx);

    /* The hex value of "I will be bk!" */
    BN_hex2bn(&M, "492077696c6c2062652062616b21");
    BN_mod_exp(S_new, M, d, n, ctx);

    /* Print values */
    printBN("Original Signature: ", S);
    printBN("New Signature: ", S_new);

    if(!BN_cmp(S, S_new)) {
        printf("Signatures match.\n");
    }
    else {
        printf("Signatures don't match.\n");
    }
    return 0;
}
```

```
/* Prints BIGNUM */  
void printBN(char *msg, BIGNUM * a) {  
    char * number_str = BN_bn2hex(a);  
    printf("%s %s\n", msg, number_str);  
    OPENSSL_free(number_str);  
}
```

5. Δραστηριότητα 5

Μέρος 5A

Τεκμηρίωση

Το μήνυμα του Bob είναι : “Launch a missile.” με δεκαεξαδική τιμή ίση με:
“4c61756e63682061206d697373696c652e”. Επίσης έχουμε:

S = 643D6F34902D9C7EC90CB0B2BCA36C47FA37165C0005CAB026C0542CBDB6802F

e = 010001

n = AE1CD4DC432798D933779FBD46C6E1247F0CF1233595113AA51B450F18116115

Στην συνέχεια αλλοιώνουμε την υπογραφή:

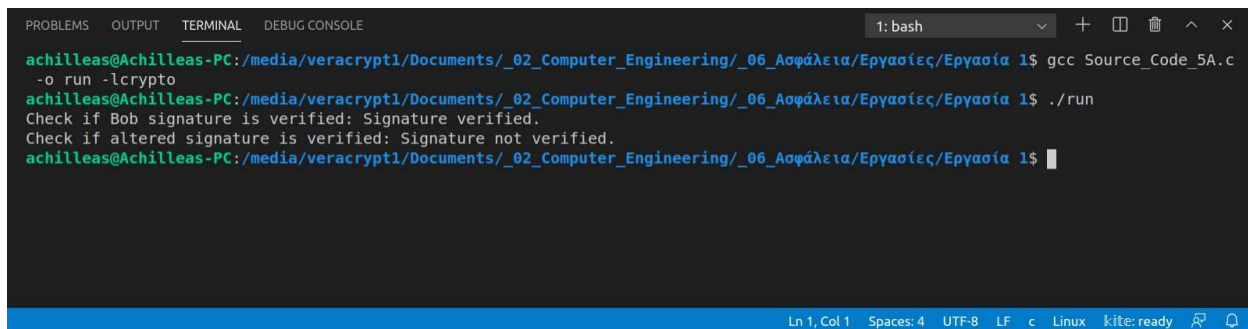
S_new = 643D6F34902D9C7EC90CB0B2BCA36C47FA37165C0005CAB026C0542CBDB6803F

Από τα αποτελέσματα, συμπεραίνουμε ότι η υπογραφή του Bob είναι έγκυρη και όταν την αλλοιώσαμε είδαμε ότι προκύπτει πρόβλημα.

Αποτελέσματα:

Check if Bob signature is verified: Signature verified.

Check if altered signature is verified: Signature not verified.



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: bash
achilleas@Achilleas-PC:/media/veracrypt1/Documents/_02_Computer_Engineering/_06_Ασφάλεια/Εργασίες/Εργασία 1$ gcc Source_Code_5A.c
-o run -lcrypto
achilleas@Achilleas-PC:/media/veracrypt1/Documents/_02_Computer_Engineering/_06_Ασφάλεια/Εργασίες/Εργασία 1$ ./run
Check if Bob signature is verified: Signature verified.
Check if altered signature is verified: Signature not verified.
achilleas@Achilleas-PC:/media/veracrypt1/Documents/_02_Computer_Engineering/_06_Ασφάλεια/Εργασίες/Εργασία 1$
```

Εικόνα 5A. Μεταγλώττιση και έλεγχος των δύο υπογραφών.

Κώδικας

```
#include <stdio.h>
#include <openssl/bn.h>

int main() {
    BN_CTX *ctx = BN_CTX_new();
    BIGNUM *n = BN_new();
    BIGNUM *e = BN_new();
    BIGNUM *S = BN_new();
    BIGNUM *S_new = BN_new();
    BIGNUM *M = BN_new();
    BIGNUM *temp1 = BN_new();
    BIGNUM *temp2 = BN_new();

    /* Initialize values */
    BN_hex2bn(&n, "AE1CD4DC432798D933779FBD46C6E1247F0CF1233595113AA51B450F18116115");
    BN_hex2bn(&e, "010001");
    BN_hex2bn(&S, "643D6F34902D9C7EC90CB0B2BCA36C47FA37165C0005CAB026C0542CBDB6802F");
    BN_hex2bn(&S_new, "643D6F34902D9C7EC90CB0B2BCA36C47FA37165C0005CAB026C0542CBDB6803F");

    /* Hex value of "Launch a missile." */
    BN_hex2bn(&M, "4c61756e63682061206d697373696c652e");

    /* Check signatures */
    BN_mod_exp(temp1, S, e, n, ctx);
    BN_mod_exp(temp2, S_new, e, n, ctx);

    printf("Check if Bob signature is verified: ");
    if(!BN_cmp(M, temp1)) {
        printf("Signature verified.\n");
    }
    else {
        printf("Signature not verified.\n");
    }

    printf("Check if altered signature is verified: ");
```

```
if(!BN_cmp(M, temp2)) {  
    printf("Signature verified.\n");  
}  
else {  
    printf("Signature not verified.\n");  
}  
return 0;  
}
```

Μέρος 5B

Τεκμηρίωση

Το μήνυμα M από την Alice είναι “Please transfer me \$2000.Alice.” με την δεκαεξαδική τιμή: “506c65617365207472616e73666572206d652024323030302e416c6963652e”.

Ακόμα έχουμε:

S = DB3F7CDB93483FC1E70E4EACA650E3C6505A3E5F49EA6EDF3E95E9A7C6C7A320

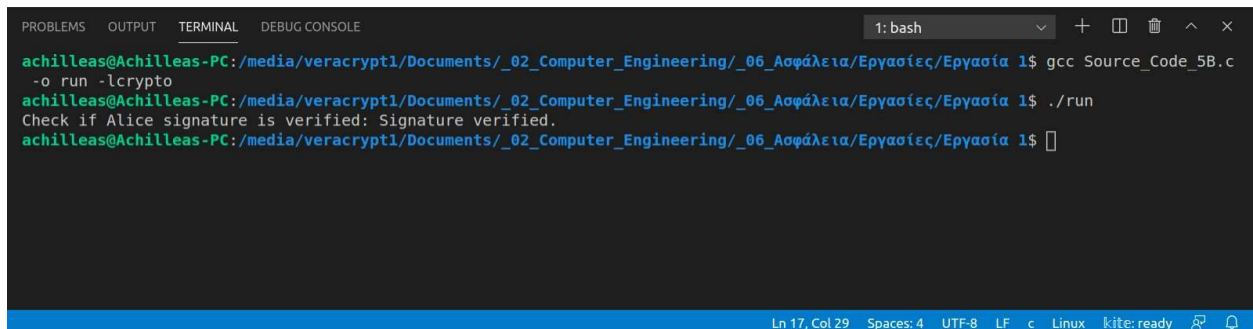
e = 010001

n = DCBFFE3E51F62E09CE7032E2677A78946A849DC4CDDE3A4D0CB81629242FB1A5

Το πρόγραμμά μας ενημερώνει ότι η υπογραφή του μηνύματος είναι έγκυρη.

Αποτελέσματα:

Check if Alice signature is verified: Signature verified.



```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
1: bash
achilleas@Achilleas-PC: /media/veracrypt1/Documents/_02_Computer_Engineering/_06_Ασφάλεια/Εργασίες/Εργασία 1$ gcc Source_Code_5B.c
-o run -lcrypto
achilleas@Achilleas-PC: /media/veracrypt1/Documents/_02_Computer_Engineering/_06_Ασφάλεια/Εργασίες/Εργασία 1$ ./run
Check if Alice signature is verified: Signature verified.
achilleas@Achilleas-PC: /media/veracrypt1/Documents/_02_Computer_Engineering/_06_Ασφάλεια/Εργασίες/Εργασία 1$
```

Εικόνα 5B. Μεταγλώττιση και έλεγχος της υπογραφής της Alice.

Κώδικας

```
#include <stdio.h>
#include <openssl/bn.h>

int main() {
    BN_CTX *ctx = BN_CTX_new();
    BIGNUM *n = BN_new();
    BIGNUM *e = BN_new();
    BIGNUM *S = BN_new();
    BIGNUM *S_new = BN_new();
    BIGNUM *M = BN_new();
    BIGNUM *temp = BN_new();

    /* Initialize values */
    BN_hex2bn(&n, "DCBFFE3E51F62E09CE7032E2677A78946A849DC4CDDE3A4D0CB81629242FB1A5");
    BN_hex2bn(&e, "010001");
    BN_hex2bn(&S, "DB3F7CDB93483FC1E70E4EACA650E3C6505A3E5F49EA6EDF3E95E9A7C6C7A320");

    /* Hex value of "Please transfer me $2000.Alice." */
    BN_hex2bn(&M, "506c65617365207472616e73666572206d6520243230302e416c6963652e");

    /* Check signatures */
    BN_mod_exp(temp, S, e, n, ctx);

    printf("Check if Alice signature is verified: ");
    if(!BN_cmp(M, temp)) {
        printf("Signature verified.\n");
    }
    else {
        printf("Signature not verified.\n");
    }

    return 0;
}
```


6. Δραστηριότητα 6

Τεκμηρίωση

Για την επιλογή της ιστοσελίδας επιλέξαμε την www.twitch.tv. Ακολουθώντας τα αναλυτικά βήματα των οδηγιών του εγραστηρίου ανακτήσαμε τα πιστοποιητικά και εξάγαμε τα παρακάτω:

n =

```
"A3C075E13298E5D9AE847C8DE8235F46955B4CA22570D790048580C9B5F48A654
D92CBA5C442A0B6792531EDF18520CD13513D67AC974D689B33865CB37B2DAAD
F77A061D1F53CFB9AFCD3D594CAC91E801B9090C8AC8DF660179C31B8C561A2E
26E5725086F249999CF94BFC78B6BB01FCA14FA189B6C107C992BDA4A63E5B24E
C2FD3E100B48F4770B2FF0964B3AEEDB35DE858DDA130ECE01C471D3D377C508
A6603925A727695C83D16F7678EEC5445B45BD293BE2C6090FA2BE2BDCE35CDA
5A6F8EE7C9076B7EA1C053958289E0785C72A86CBE676BABE733D987F2F85C27F
4F62A3B87EFDAC247DABFACEB27647B4C53EB34E12F9B204D54126B7D28BD"
```

e = "10001"

hash = "4309f90f8aff4f37b2870152055d7fd11ec735a62caeab957b323b90b7a66f32"

s =

```
"8975786dcff62a3b9b26bc290c73147a3206cc35824867a48a2146cdf9899b7b5410f51a
ea979a24271c310f7ab2a79a6d0245041f77f95d589fc722ab9793d63bd51645ce70a4e4
e175daf4d4f0d590bfe1f27ba9cbda542bbaf81a2337e3057d67295b2b35ca965a3249da
1ada02469964c7870c659b991b0407debc25b149735b56a54db1d59ce192383b49df13d
23dc7092658fd307ff81b2d3b5961dd137af10e5146e1bda424fea26a0cf22253ebf6d890
a1c080e75c33be6bf6f1c8f7aba80ee170029b81137a23a64e6efd8244629a6cae328e1bf
6a6adae7bbde44c1814d5ebb4ef4eec6460faa832723e0257c5fd9d10bf877122856fe90
e70ad15"
```

Τρέχοντας το πρόγραμμα ότι στο τέλος της υπογραφής υπάρχει το hash μας.

Αποτελέσματα:

Signature:

```
01FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
003031300D06096086480165030402010500042
04309F90F8AFF4F37B2870152055D7FD11EC735A62CAEAB957B323B90B7A66F3
2
```

Hash:

4309F90F8AFF4F37B2870152055D7FD11EC735A62CAEAB957B323B90B7A66F32

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
```

```
1: bash  
achilleas@Achilleas-PC:/media/veracrypt1/Documents/_02_Computer_Engineering/_06_Ασφάλεια/Εργασίες/Εργασία 1$ gcc Source_Code_6.c -o run -lcrypto  
achilleas@Achilleas-PC:/media/veracrypt1/Documents/_02_Computer_Engineering/_06_Ασφάλεια/Εργασίες/Εργασία 1$ ./run  
Result: 01FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF  
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF  
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF  
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF  
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF083031300D069968648016503040201050004204309F90F8AFF4F37B287015205D7FD11EC735A62CAEAB957B323B90B7A66  
F32  
Hash: 4309F90F8AFF4F37B287015205D7FD11EC735A62CAEAB957B323B90B7A66F32  
achilleas@Achilleas-PC:/media/veracrypt1/Documents/_02_Computer_Engineering/_06_Ασφάλεια/Εργασίες/Εργασία 1$ █  
  
Ln 18, Col 26 Spaces: 4 UTF-8 LF c Linux kite: ready
```

Εικόνα 6. Μεταγλώττιση και αν τα hash είναι ίδια.

Κώδικας

```
#include <stdio.h>
#include <openssl/bn.h>

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

int main() {
    BN_CTX *ctx = BN_CTX_new();
    BIGNUM *public_key = BN_new();
    BIGNUM *signature = BN_new();
    BIGNUM *n = BN_new();
    BIGNUM *hash = BN_new();
    BIGNUM *temp = BN_new();

    BN_hex2bn(&public_key, "10001");

    BN_hex2bn(&n,
"A3C075E13298E5D9AE847C8DE8235F46955B4CA22570D790048580C9B5F48A654D92CBA5C442A0B6792531EDF18520CD
13513D67AC974D689B33865CB37B2DAADF77A061D1F53CFB9AFCD3D594CAC91E801B9090C8AC8DF660179C31B8C561A2E
26E5725086F249999CF94BFC78B6BB01FCA14FA189B6C107C992BDA4A63E5B24EC2FD3E100B48F4770B2FF0964B3AEEDB
35DE858DDA130ECE01C471D3D377C508A6603925A727695C83D16F7678EEC5445B45BD293BE2C6090FA2BE2BDCE35CDA5
A6F8EE7C9076B7EA1C053958289E0785C72A86CBE676BABE733D987F2F85C27F4F62A3B87EFDAC247DABFACED27647B4C
53EB34E12F9B204D54126B7D28BD");

    BN_hex2bn(&signature,
"8975786dcff62a3b9b26bc290c73147a3206cc35824867a48a2146cdf9899b7b5410f51aea979a24271c310f7ab2a79a
6d0245041f77f95d589fc722ab9793d63bd51645ce70a4e4e175daf4d4f0d590bfe1f27ba9cbda542bbaf81a2337e3057
d67295b2b35ca965a3249da1ada02469964c7870c659b991b0407debc25b149735b56a54db1d59ce192383b49df13d23d
c7092658fd307ff81b2d3b5961dd137af10e5146e1bda424fea26a0cf22253ebf6d890a1c080e75c33be6bf6f1c8f7aba
80ee170029b81137a23a64e6efd8244629a6cae328e1bf6a6adae7bbde44c1814d5ebb4ef4eec6460faa832723e0257c5
fd9d10bf877122856fe90e70ad15");

    BN_hex2bn(&hash, "4309f90f8aff4f37b2870152055d7fd11ec735a62caeab957b323b90b7a66f32");

    BN_mod_exp(temp, signature, public_key, n, ctx);

    printBN("Result: ", temp);
```

```
    printBN("Hash:", hash);  
    return 0;  
}  
  
/* Prints BIGNUM */  
void printBN(char *msg, BIGNUM * a) {  
    char * number_str = BN_bn2hex(a);  
    printf("%s %s\n", msg, number_str);  
    OPENSSL_free(number_str);  
}
```