



24 ΝΟΕΜΒΡΙΟΥ 2022

# ΑΣΦΑΛΕΙΑ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΕΡΓΑΣΙΑ 2

**Συντελεστές εργασίας**

Χριστοφορίδης Χαράλαμπος – Π19188

Γεωργιάδης Νικόλαος – Π19032

Καρκάνης Ευστράτιος – Π19064



# Περιεχόμενα

0. Εισαγωγή .....	2
1. Μέρος (α) – CrypTool .....	2
1.1 Δημιουργία αρχείου txt.....	2
1.2 Δημιουργία ενός ασύμμετρου ζεύγους κλειδιών RSA και προβολή πιστοποιητικού .....	2
1.3 Κρυπτογράφηση κειμένου με υβριδική κρυπτογραφία RSA-AES	6
1.4 Δημιουργία αρχείων txt .....	11
1.5 Εύρεση επικίνδυνου μηνύματος .....	12
1.6 Επίθεση στην τιμή hash και ψηφιακές υπογραφές .....	22
1.7 Επίθεση παραγοντοποίησης σε RSA moduli .....	22
2. Μέρος (β) – Χρήση GPG.....	26
2.1 Δημιουργήστε ένα ζεύγος κλειδιών.....	26
2.2 Ανέβασμα πιστοποιητικού σε server .....	28
2.3 Εγκατάσταση και υπογραφή κλειδιών.....	28
2.4 Αποστολή κρυπτογραφημένων email.....	30

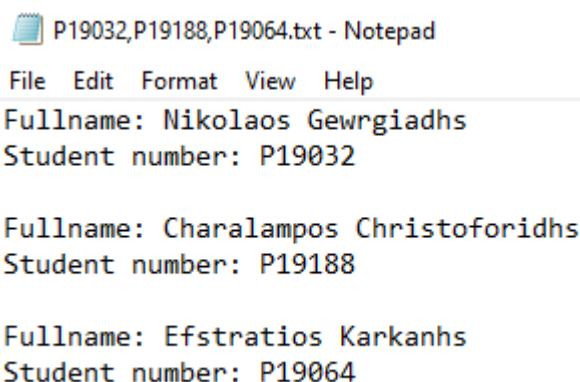
## 0. Εισαγωγή

Σε αυτή την εργασία θα χρησιμοποιήσουμε τα εργαλεία **CrypTool 1.4.42** και **GPG 4.0.4** για να εφαρμόσουμε τις γνώσεις κρυπτογραφίας που διδαχθήκαμε στα πλαίσια του μαθήματος.

## 1. Μέρος (α) – CrypTool

### 1.1 Δημιουργία αρχείου txt

Δημιουργούμε ένα αρχείο txt με τα στοιχεία της ομάδας (Ονοματεπώνυμο, αριθμός μητρώου):



A screenshot of a Notepad window titled "P19032,P19188,P19064.txt - Notepad". The window contains three lines of text, each representing a student's information. The first line is "Fullname: Nikolaos Gewrgiadhs" followed by "Student number: P19032". The second line is "Fullname: Charalampos Christoforidhs" followed by "Student number: P19188". The third line is "Fullname: Efstratios Karkanhs" followed by "Student number: P19064". The text is in a monospaced font.

```
P19032,P19188,P19064.txt - Notepad
File Edit Format View Help
Fullname: Nikolaos Gewrgiadhs
Student number: P19032

Fullname: Charalampos Christoforidhs
Student number: P19188

Fullname: Efstratios Karkanhs
Student number: P19064
```

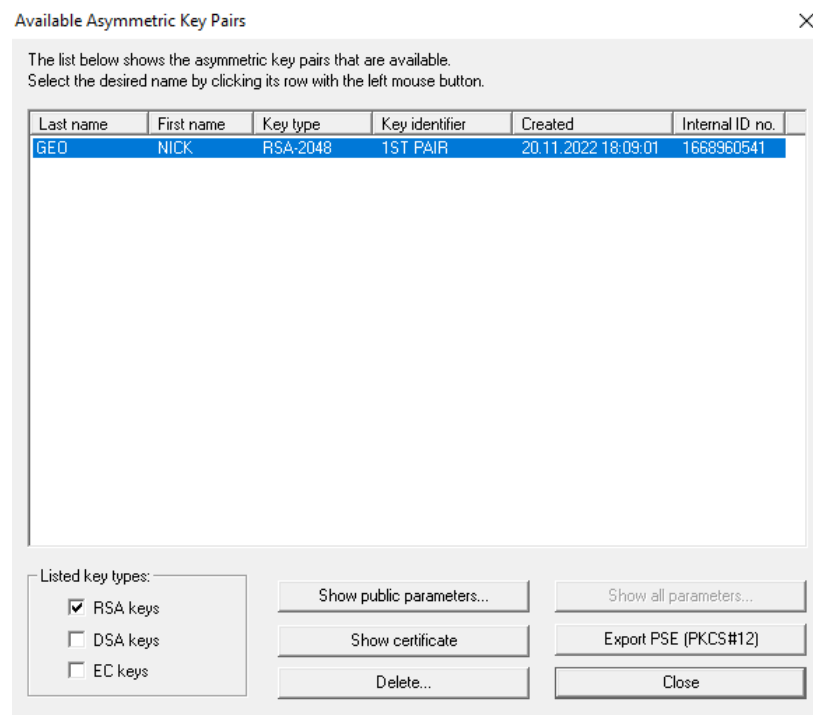
*Αρχείο txt*

### 1.2 Δημιουργία ενός ασύμμετρου ζεύγους κλειδιών RSA και προβολή πιστοποιητικού

Δημιουργούμε το ζεύγος ως εξής:

- Στο παράθυρο του CrypTool μεταβαίνουμε στο Digital Signatures/PKI → PKI → Generate/Import Keys
- Επιλέγουμε RSA και Bit Length 2048
- Συμπληρώνουμε Firstname, Lastname και ένα PIN
- Πατάμε Generate new key pair
- Στο παράθυρο του CrypTool μεταβαίνουμε στο Digital Signatures/PKI → PKI → Display/Export Keys

Μπορούμε να διακρίνουμε στη λίστα το key pair που μόλις φτιάξαμε:



Επιλέγουμε το key pair και στη συνέχεια πατάμε show certificate. Το περιεχόμενό του απεικονίζεται παρακάτω:

Version: 2 (X.509v3-1996)

SubjectName: CN=NICK GEO [1668960541], DC=cryptool, DC=org

IssuerName: CN=CrypTool CA 2, DC=cryptool, DC=org

SerialNumber: 4D:C9:2E:BE:84:FB:35:5B

Validity - NotBefore: Sun Nov 20 18:09:07 2022 (221120160907Z)

NotAfter: Mon Nov 20 18:09:07 2023 (231120160907Z)

Public Key Fingerprint: 9A53 CF57 7749 7D95 0E67 305B 70FF 6836

SubjectKey: Algorithm rsa (OID 2.5.8.1.1), Key size = 2048

Public modulus (no. of bits = 2048):

0 FEE19DB0 A40063D5 B703FC7D 0BB5889F

10 DA4165DC AF9E07BB B37FE667 9C7334C4  
20 FC873CA4 9B3ACD51 0AE277BE 576FB8EE  
30 21D0A829 56E41521 EEBF0A10 3A49133E  
40 14C1A7B8 EF4C430F 77EF13A5 CB51170B  
50 98E69728 058BAC78 045ADE19 ACF890B4  
60 721EB225 DE76A936 9512DDD8 E07C6C2A  
70 AD2B2D36 256F373E 8FF91001 7C02CA18  
80 1404FAB9 332164E5 AE4EDE2D 356714DD  
90 DA4A4160 5EA6EE01 1DE36FF9 CBB3ED21  
A0 FE7E6F91 9293E712 BF686285 72F65FED  
B0 49980844 0E0BF105 57B4D5BF B3CA1D12  
C0 54334854 B9AB2C3A 8DC3DDCC 8685E1B7  
D0 3EF6A2A1 BD39B261 98F9020A B6A0010A  
E0 422C46A9 A96FD403 4090BD5D F0678A04  
F0 C5817921 1ED59DDC 4E4096B3 5AF9D77B

Public exponent (no. of bits = 17):

0 010001  
1

Certificate extensions:

Private extensions:

OID 2.206.5.4.3.2:

PrintableString:

|[GEO][NICK][RSA-2048][1668960541 |  
|][1ST PAIR] |

SHA1 digest of DER code of ToBeSigned:

0 3B784AAD 2EB1827E 7F00A99D 516D680D  
10 A12D4C20

Signature: Algorithm sha1WithRSASignature (OID 1.3.14.3.2.29), NULL

```
0  C7F04F42 50EB89CC F6D4354C C1419615
10 FB3CF21C 6BF338CF 160DF6E4 3A650F20
20 2836615F 3B7860EC C108EA68 6B86339A
30 F658552E D89BD412 53F9D125 3B7E17B5
40 A8C30F7A 935D3511 242E95C4 76EC8E77
50 56DE126D 4F49225A 69FF7460 D496C70E
60 8D66B502 5EC2E888 77403C06 1793D086
70 881A6378 E8C1A023 0836ECE7 CC5FD0FD
80 4922EB23 FD46EAE2 EBE20773 D917968D
90 9C7811DD D84EE061 4A4FE46B D3F982AC
A0 5AC0CA31 48409B8D 8A51833B 2A86DDFA
B0 87098C8E 59FB8471 7FC07086 FE39EF79
C0 9515BE10 242E26FF 347AA64A B5CED073
D0 53AAD29D A16D958A 74D30143 E73F4387
E0 244320A1 3F812FB1 3219207C DA8C1733
F0 BEE24811 BA39DD4A 20578FFA 3E053C9D
```

Certificate Fingerprint (MD5):

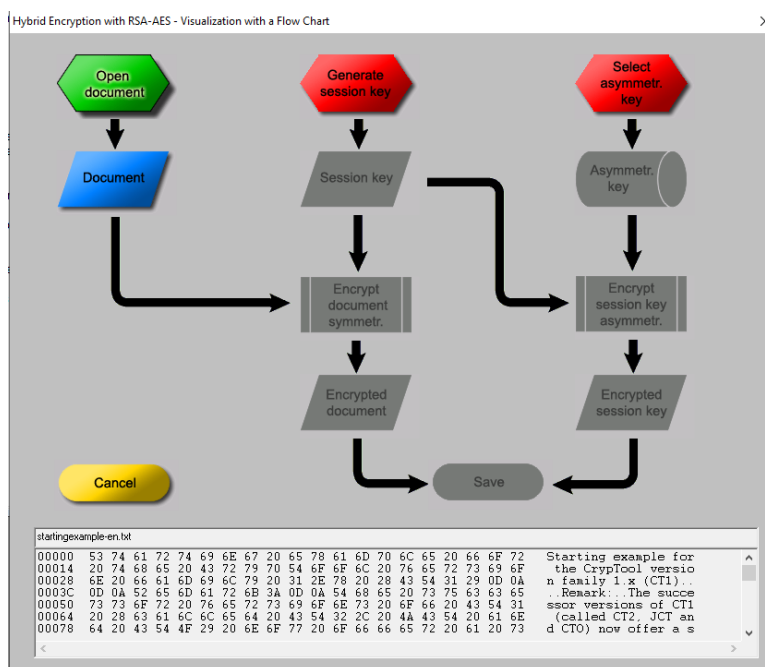
2A:52:04:00:57:D9:15:FF:86:57:0E:5A:4F:7B:F6:8D

Certificate Fingerprint (SHA-1): E9B1 4BDC C4D4 D37F C21C 53CD DA65 D651  
7B87 4DC9

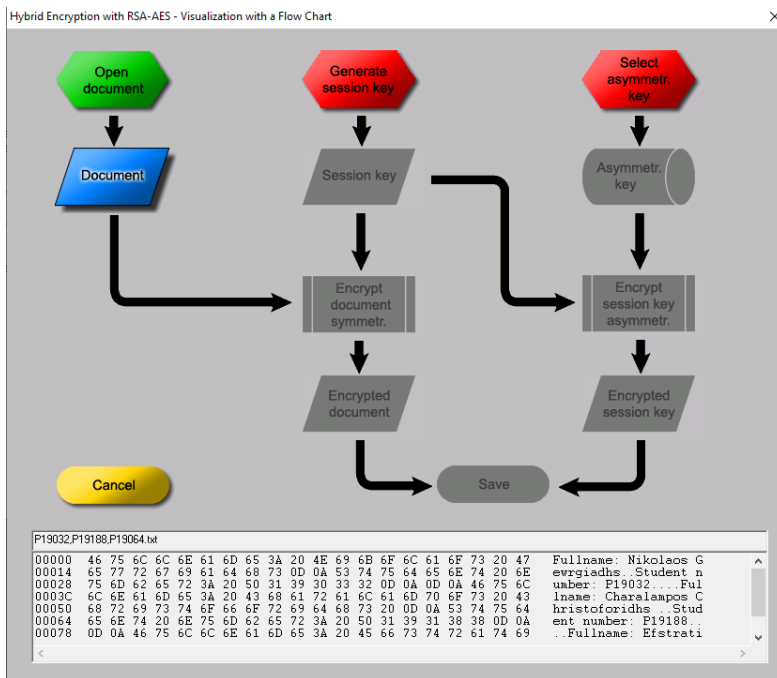
### 1.3 Κρυπτογράφηση κειμένου με υβριδική κρυπτογραφία RSA-AES

Στο μενού επιλογών επιλέγουμε Encrypt/Decrypt → Hybrid → RSA-AES Encryption.

Στο παράθυρο που εμφανίζεται επιλέγουμε Open Document και ανοίγουμε το αρχείο txt του ερωτήματος 1.

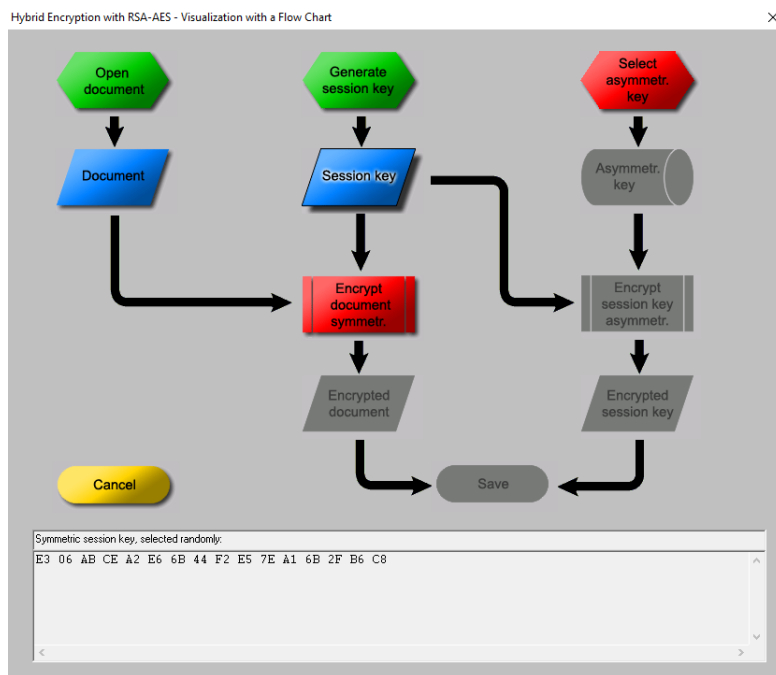


*Πριν το άνοιγμα του αρχείου*



*Μετά το άνοιγμα του αρχείου (μπορούμε να διακρίνουμε το plaintext στο παραθυράκι στο κάτω μέρος)*

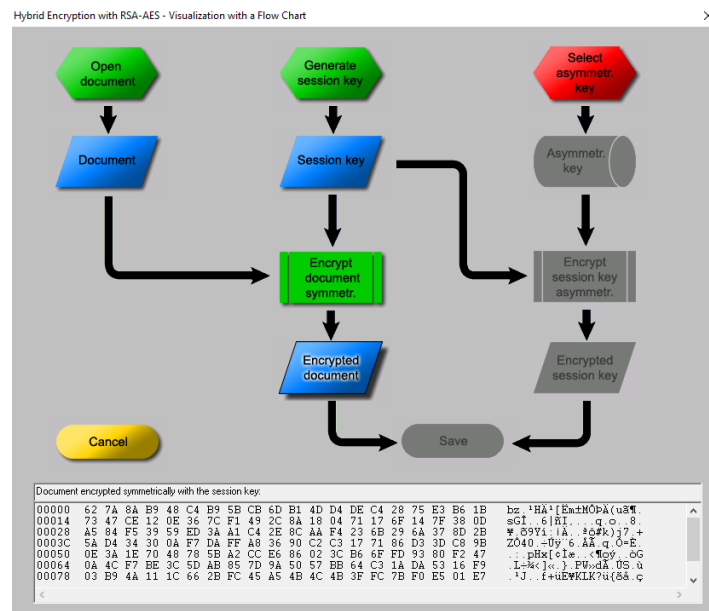
Επιλέγουμε Generate session key για να δημιουργηθεί ένα τυχαίο συμμετρικό κλειδί. Επιλέγουμε Session key για να προβληθεί το κλειδί:



**Προβολή Session key: E3 06 AB CE A2 E6 6B 44 F2 E5 7E A1 6B 2F B6 C8**



Επιλέγουμε Encrypt document symmetr., για να κρυπτογραφήσουμε το κείμενο με το συμμετρικό session κλειδί. Επιλέγουμε Encrypted document για να προβληθεί το κρυπτογραφημένο κείμενο:



*Κρυπτογραφημένο κείμενο(ciphertext) με το session key*

### Encrypted document:

```

00000 62 7A 8A B9 48 C4 B9 5B CB 6D B1 4D D4 DE C4 28 75 E3 B6 1B  bz.'HÄ'1[Ëm±MÔPÄ(uã¶.
00014 73 47 CE 12 0E 36 7C F1 49 2C 8A 18 04 71 17 6F 14 7F 38 0D  sGÎ..6|ñl,...q.o..8.
00028 A5 84 F5 39 59 ED 3A A1 C4 2E 8C AA F4 23 6B 29 6A 37 8D 2B  ¥.ö9Yí:;jÄ..ªð#k)j7.+
0003C 5A D4 34 30 0A F7 DA FF A8 36 90 C2 C3 17 71 86 D3 3D C8 9B  ZÔ40.÷Úÿ"6.ÃÃ.q.Ó=È.
00050 0E 3A 1E 70 48 78 5B A2 CC E6 86 02 3C B6 6F FD 93 80 F2 47  ..pHx[çlæ..<¶oý..òG
00064 0A 4C F7 BE 3C 5D AB 85 7D 9A 50 57 BB 64 C3 1A DA 53 16 F9  .L÷¾<]«.}.PW»dÃ.ÚS.ù
00078 03 B9 4A 11 1C 66 2B FC 45 A5 4B 4C 4B 3F FC 7B F0 E5 01 E7  .'J..f+üE¥KLK?ü{ðã.ç
0008C F1 4E 31 57 88 4A 96 C2 5F 31 63 94 2A 54 32 90 71 B8 03 8A  ñN1W.J.Â_1c.*T2.q,..
000A0 1C 65 C4 AD 50 0C B4 97 F0 72 26 7F 7B 09 D1 3F          .eÄ-P.'.'ðr&.{.Ñ?

```

Επιλέγουμε Select asymmetr. Key, για να εισάγουμε το ασύμμετρο κλειδί που είχαμε δημιουργήσει.

RSA key for the hybrid encryption

Select the receiver key from the list.

Last name	First name	Key type	Key identifier	Created	Internal ID no.
GEO	NICK	RSA-2048	1ST PAIR	20.11.2022 18:09:01	1668960541

Note: Here only names are displayed, which have an RSA key.

OK Cancel

**Επιλογή του RSA public key**

Public key of: NICK GEO

Modulus: 32175784845971575111311681191343821176774454550248771665235  
930831042385811940068662959972608627343757908820752250957660  
250916001554199511344749811771516202392744390494196940234409

Exponent: 65537

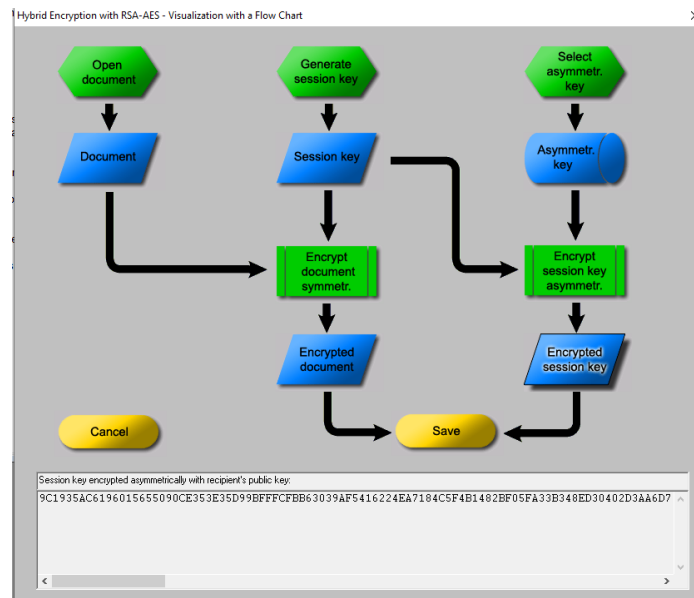
Base for presentation of numbers

☐ Octal ☒ Decimal ☐ Hexadecimal

Back

**Πληροφορίες των RSA παραμέτρων**

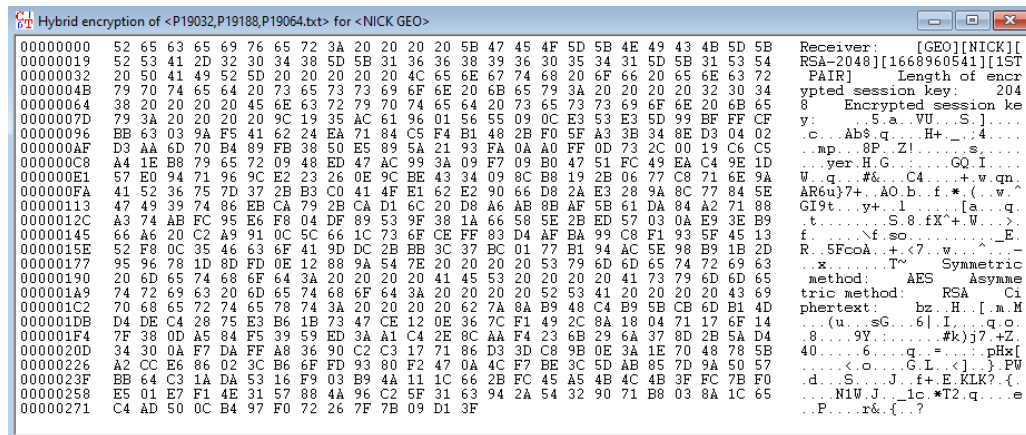
Επιλέγουμε Encrypt session key asymmetr., για να κρυπτογραφήσουμε το συμμετρικό session key με το ασύμμετρο public RSA key. Στη συνέχεια επιλέγουμε Encrypted session key:



*Κρυπτογραφημένο session key με το public RSA key:*

9C1935AC6196015655090CE353E35D99BFFFCFBB63039AF5416224EA71  
 84C5F4B1482BF05FA33B348ED30402D3AA6D70B489FB3850E5895A219  
 3FA0AA0FF0D732C0019C6C5A41EB87965720948ED47AC993A09F709B0  
 4751FC49EAC49E1D57E09471969CE223260E9CBE4334098CB8192B0677  
 C8716E9A415236757D372BB3C0414FE162E29066D82AE3289A8C77845E  
 4749397486EBCA792BCAD16C20D8A6AB8BAF5B61DA84A27188A374AB  
 FC95E6F804DF89539F381A66585E2BED57030AE93EB966A620C2A9910C  
 5C661C736FCEFF83D4AFBA99C8F1935F451352F80C3546636F419DDC2B  
 BB3C37BC0177B194AC5E98B91B2D9596781D8DFD0E12889A547E

Πατώντας save εμφανίζεται το εξής παράθυρο:



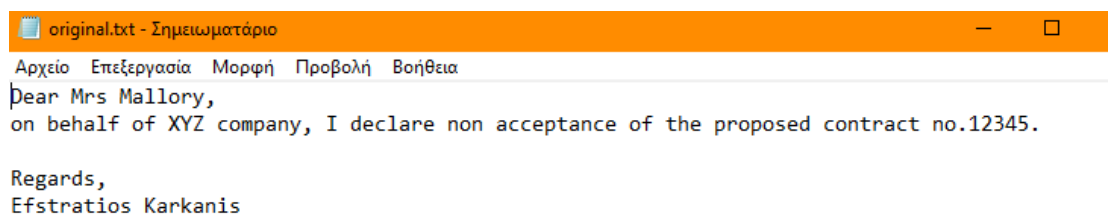
```
Hybrid encryption of <P19032,P19188,P19064.txt> for <NICK GEO>
Receiver: [GEO][NICK][
RSA-2048][1668960541][1ST
PAIR] Length of encr
ypted session key: 204
8 Encrypted session ke
y: 5 a..VU..S.]...
c...Ab$.q...H+...4...
mp...8P...Zl...s...
yer:H.G...GQ.I...
W.q...#...C4...+w.qn.
AR6u}7+...AO.b.f.*(..w.^
GI9t...y+...l...[a...q.
t...S.8.fX'+W...>
f...f.s...E.
R..5FcoA...<7..w...
x...T~ Symmetric
method: AES Asynae
tric method: RSA Ci
phertext: bz..H..[.m.M
(u...sG...6].I...q.o.
8...9Y...#k)j7.+Z.
40...6...q...:pHx[
...<.o...G.L.<...}PW
d...S...J...f+.E.KLK?...{
...N1W.J...le.*T2.q...e
...P...r&{...?
```

Αποτέλεσμα κρυπτογράφησης κειμένου και session key

Να σημειωθεί ξανά, πως το session key κρυπτογραφείται με τον αλγόριθμο RSA χρησιμοποιώντας το public key που δημιουργήσαμε στο βήμα 2 και το μήνυμα κρυπτογραφείται με τον αλγόριθμο AES χρησιμοποιώντας το τυχαίο session key που δημιουργήσαμε σε αυτό το βήμα.

## 1.4 Δημιουργία αρχείων txt

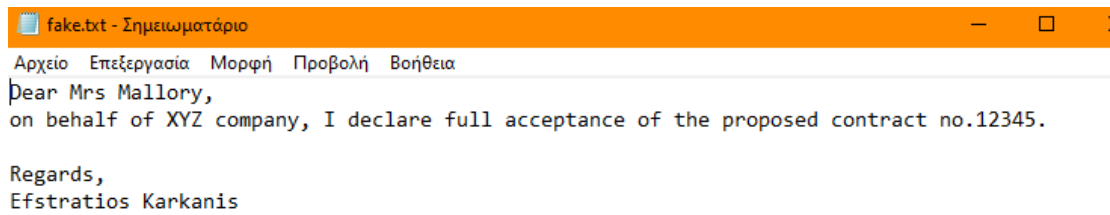
Σε αυτό το βήμα, δημιουργούμε δύο αρχεία κειμένου **original.txt** και **fake.txt**



```
original.txt - Σημειωματάριο
Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια
Dear Mrs Mallory,
on behalf of XYZ company, I declare non acceptance of the proposed contract no.12345.

Regards,
Efstratios Karkanis
```

Αρχείο original.txt



*Αρχείο fake.txt*

## 1.5 Εύρεση επικίνδυνου μηνύματος

Στο συγκεκριμένο ερώτημα καλούμαστε, με βάση τα κείμενα που δημιουργήσαμε στο παραπάνω βήμα, να βρούμε ένα «επικίνδυνο» μήνυμα το οποίο μοιάζει στο fake.txt και να έχει την ίδια τιμή hash με το original.txt για τις παρακάτω περιπτώσεις:

- Για τον αλγόριθμο MD5 και τα πρώτα 16 bit της τιμής hash.
- Για τον αλγόριθμο MD5 και τα πρώτα 50 bit της τιμής hash.
- Για τον αλγόριθμο SHA1 και τα πρώτα 80 bit της τιμής hash.
- Για τον αλγόριθμο SHA1 και όλα (160) τα bit της τιμής hash.

Η διαδικασία που θα ακολουθήσουμε είναι η εξής. Αρχικά θα μετακινηθούμε μέσω του μενού σε Analysis → Hash → Attack on the hash value of the digital signature. Ύστερα στο βοηθητικό παράθυρο θα επιλέξουμε τον αλγόριθμο και τα αρχεία που επιθυμούμε (στην περίπτωση μας πάντα τα original.txt και fake.txt).

## 1. Για τον αλγόριθμο MD5 και τα πρώτα 16 bit της τιμής hash.

Αφού βάλουμε τις απαραίτητες παραμέτρους.

Attack on the Hash Value of the Digital Signature ×

This attack attempts to find two different messages that hash to the same value.

Use default messages

Choose "harmless" file

The attacker assumes that his victim will digitally sign the "harmless" message due to its non-malicious content.

C:\Users\30698\Desktop\original.txt.txt Browse ...

Choose "dangerous" file

If the attack is successful, the attacker can argue that the victim has digitally signed the "dangerous" instead of the "harmless" message.

C:\Users\30698\Desktop\fake.txt.txt Browse ...

Start search / Set options

Click "Start search" to initiate the attack. The program will search for modifications of the two messages that hash to the same value.

The message will not appear to change, since only unprintable characters will be used to modify them.

In the "Options" you can select the hash function, the required minimum number of matching bits, and the message modification method.

Start search Options ... Cancel

Options for the Attack on the Hash Value of the Digital Signature X

Hash function

Choose a hash function and the minimum required number of matching bits for the attack to be considered successful.

☐ MD2 ☐ MD4 ☒ MD5

☐ SHA ☐ SHA-1 ☐ RIPEMD-160

Significant bit length  (Co-domain: 1 - 128)

Options for the modification of messages

Determine the way messages are modified throughout the attack.

☒ Insert blanks ☐ In front of end of line

☒ Double blanks

☐ Attach characters ☐ Printable characters (demonstration)

☒ Unprintable characters

Apply Restore defaults Cancel

Πραγματοποιούμε την επίθεση. Ο χρόνος είναι σχεδόν μηδενικός.

Assumed efforts

Calculation time

Steps required

Efforts made to find a pair of messages

Calculation time

Steps required

Hash operations performed

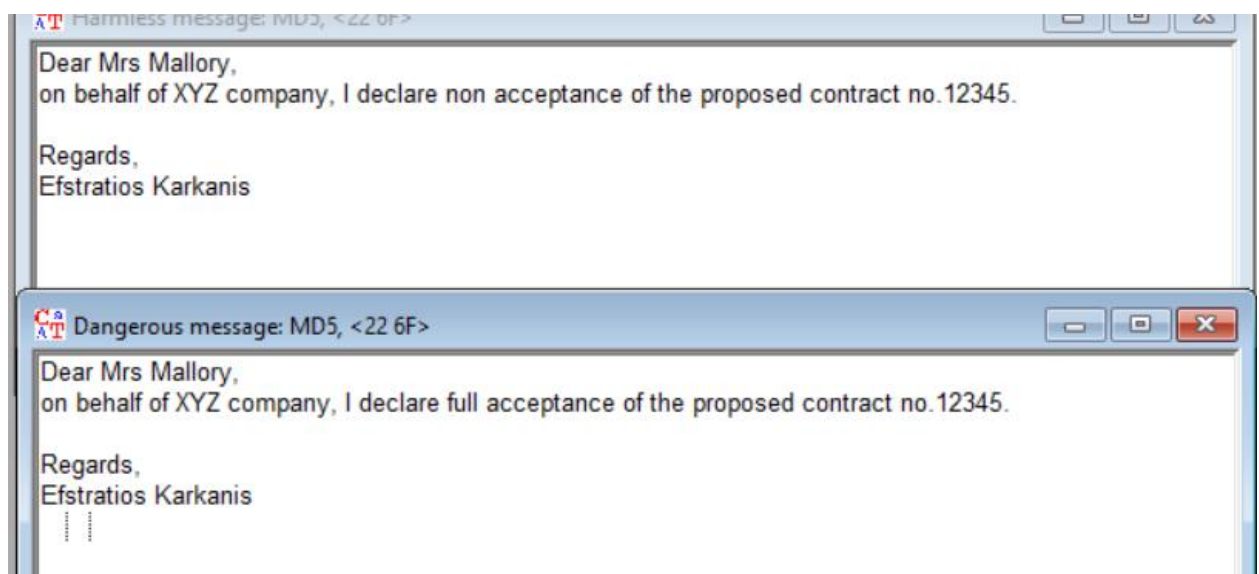
Steps required sorted by run

Run ...	Steps until collision	Collision check	Total steps
1	148	65	213
2	231	38	269

Additional bytes

10 bytes were added to the harmless message.

10 bytes were added to the dangerous message.





## 2. Για τον αλγόριθμο MD5 και τα πρώτα 50 bit της τιμής hash.

Φορτώνουμε και πάλι τα δεδομένα, και αφού κάνουμε τις απαραίτητες αλλαγές, πραγματοποιούμε την επίθεση. Ο χρόνος ολοκλήρωσης ήταν περίπου 3 λεπτά.

The image shows a Windows-style dialog box titled "Options for the Attack on the Hash Value of the Digital Signature". It contains two main sections: "Hash function" and "Options for the modification of messages".

**Hash function section:**

- Instruction: "Choose a hash function and the minimum required number of matching bits for the attack to be considered successful."
- Radio buttons for hash functions: MD2, MD4, MD5 (selected), SHA, SHA-1, RIPEMD-160.
- Text input for "Significant bit length": 50. A note "(Co-domain: 1 - 128)" is shown to the right.

**Options for the modification of messages section:**

- Instruction: "Determine the way messages are modified throughout the attack."
- Radio buttons for modification options: Insert blanks (selected), Attach characters.
- Checkboxes for additional options: In front of end of line, Double blanks (checked).
- Radio buttons for character sets: Printable characters (demonstration), Unprintable characters (selected).

At the bottom of the dialog are three buttons: "Apply", "Restore defaults", and "Cancel".

## Statistics of the Attack



Assumed efforts

Calculation time: 0 year(s), 0 day(s), 0 hour(s), 9 minute(s) und 6.45 second(s)

Steps required: 83,886,080

Efforts made to find a pair of messages

Calculation time: 0 year(s), 0 day(s), 0 hour(s), 2 minute(s) und 37.57 second(s)

Steps required: 189,428,999

Hash operations performed: 500,617,564

Steps required sorted by run

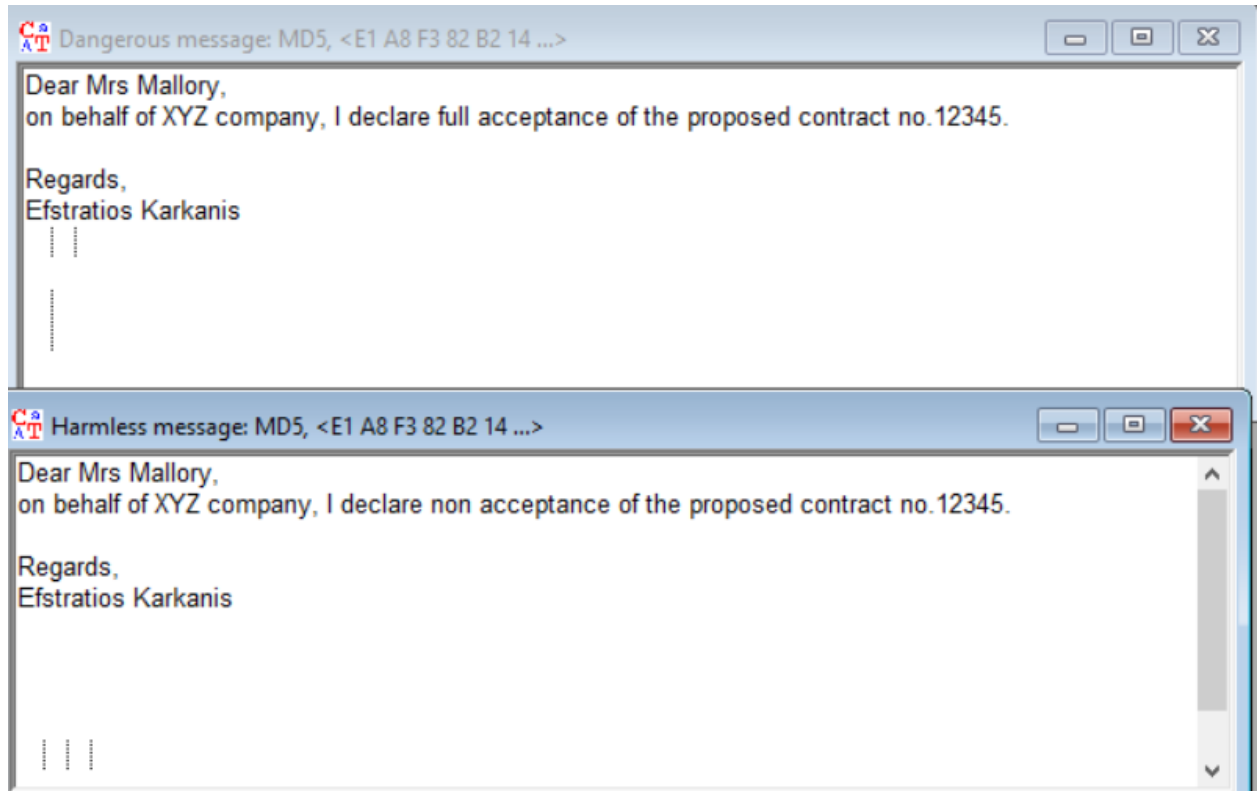
Run ...	Steps until collision	Collision check	Total steps
1	46,196,054	29,333,417	75,529,471
2	75,563,512	38,336,016	113,899,528

Additional bytes

27 bytes were added to the harmless message.

27 bytes were added to the dangerous message.

Print statistics Cancel



### 3. Για τον αλγόριθμο SHA1 και τα πρώτα 80 bit της τιμής hash.

Φορτώνουμε και πάλι τα δεδομένα μας και αφού κάνουμε τις απαραίτητες αλλαγές εκτελούμε την επίθεσή μας, ο χρόνος ολοκλήρωσης ήταν περίπου 30 ημέρες.

Options for the Attack on the Hash Value of the Digital Signature

Hash function

Choose a hash function and the minimum required number of matching bits for the attack to be considered successful.

☐ MD2      ☐ MD4      ☐ MD5

☐ SHA      ☒ SHA-1      ☐ RIPEMD-160

Significant bit length  (Co-domain: 1 - 160)

Options for the modification of messages

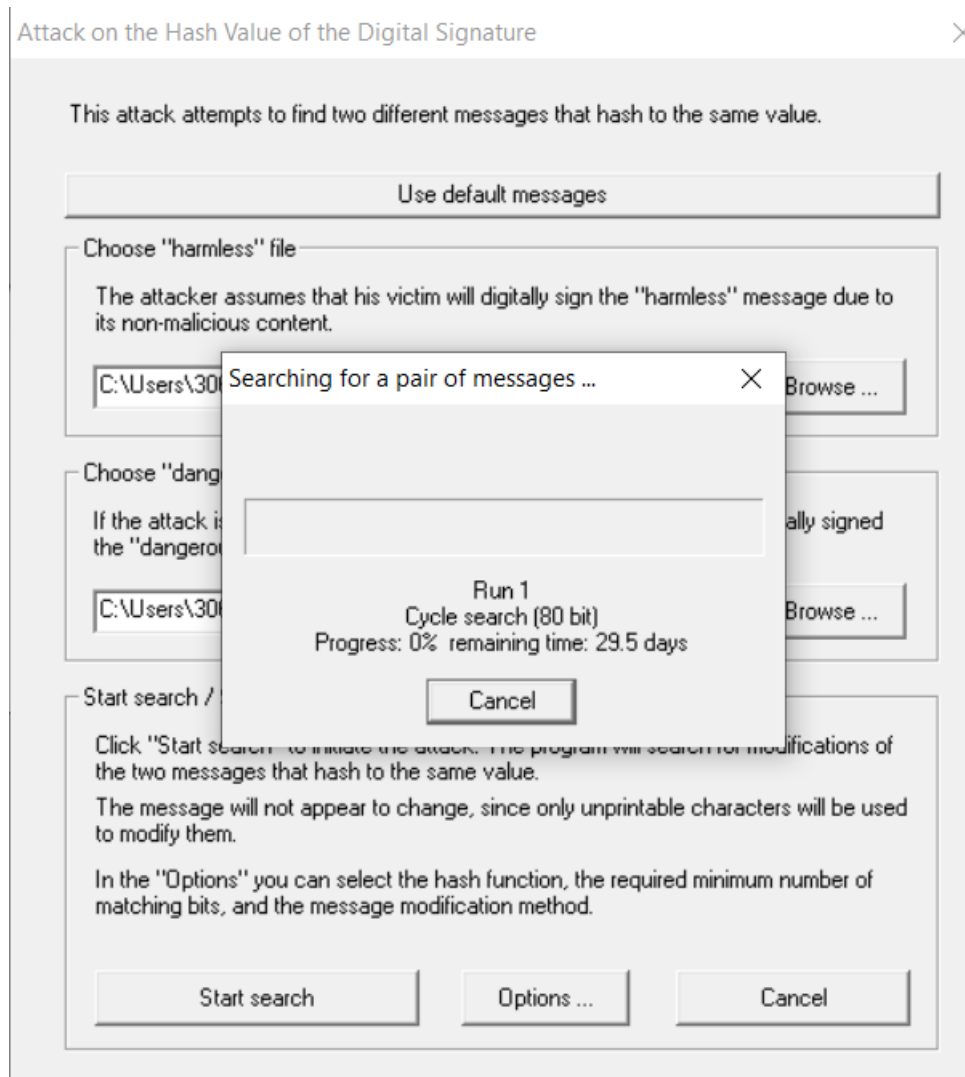
Determine the way messages are modified throughout the attack.

☒ Insert blanks      ☐ In front of end of line

☒ Double blanks

☐ Attach characters      ☐ Printable characters (demonstration)

☒ Unprintable characters



#### 4. Για τον αλγόριθμο SHA1 και όλα (160) τα bit της τιμής hash.

Φορτώνουμε και πάλι τα δεδομένα μας, και αφού κάνουμε τις απαραίτητες αλλαγές στον αλγόριθμο, εκτελούμε την επίθεσή μας. Ο χρόνος ολοκλήρωσης ήταν υπερβολικά μεγάλος.

**Hash function**

Choose a hash function and the minimum required number of matching bits for the attack to be considered successful.

☐ MD2      ☐ MD4      ☐ MD5

☐ SHA      ☒ SHA-1      ☐ RIPEMD-160

Significant bit length  (Co-domain: 1 - 160)

**Options for the modification of messages**

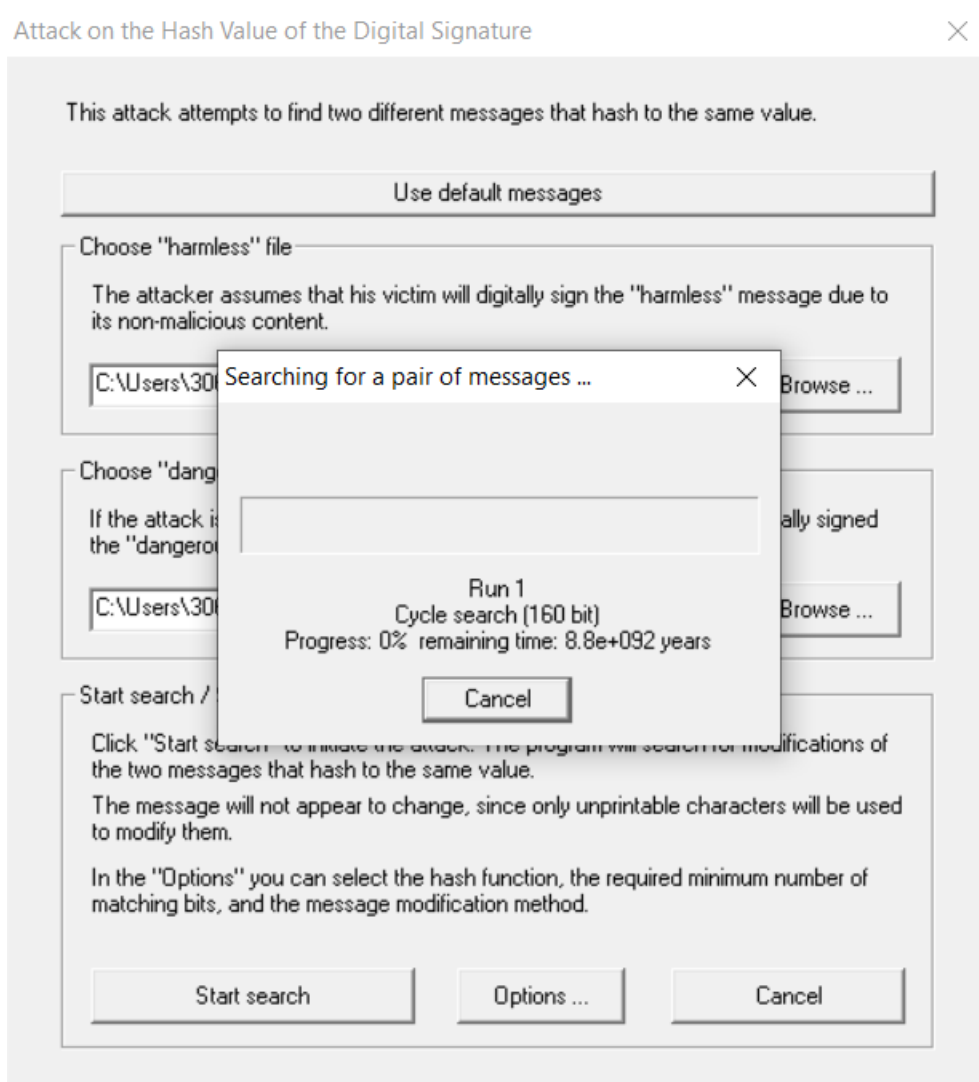
Determine the way messages are modified throughout the attack.

☒ Insert blanks      ☐ In front of end of line

☒ Double blanks

☐ Attach characters      ☐ Printable characters (demonstration)

☒ Unprintable characters



Σχόλια:

- Μπορούμε να καταλάβουμε μέσω αυτού του ερωτήματος την τεράστια διαφορά στην ασφάλεια των δύο αλγορίθμων.
- Προφανώς δεν ολοκληρώσαμε τα δύο τελευταία παραδείγματα καθώς δεν ήταν χρονικά βιώσιμα.
- Στις αλλαγές των αρχείων με το ίδιο hash-value (κατηγορία 1 και 2) οι διαφορές είναι είτε σε χαρακτήρες που δεν φαίνονται, είτε σε μικρούς χαρακτήρες κ.λπ., όπως αναγράφεται και στην εφαρμογή.

## 1.6 Επίθεση στην τιμή hash και ψηφιακές υπογραφές

Προφανώς, μία τέτοια επίθεση όπως περιγράφουμε και εκτελούμε στο 1.5 θα ήταν καταστροφική για την ασφάλεια μίας ψηφιακής υπογραφής, καθώς αν καταφέρει κάποιος να πετύχει ένα collision-attack στη συνάρτηση Hash, θα του δοθεί η δυνατότητα να υπογράψει ο ίδιος με το όνομα και τα στοιχεία ενός άλλου. Κάτι τέτοιο θα μπορούσε να φανεί ιδιαίτερα ζημιογόνο σε ατομικό επίπεδο και πόσο μάλλον σε επίπεδο ενός οργανισμού.

## 1.7 Επίθεση παραγοντοποίησης σε RSA moduli

Σε αυτό το κομμάτι της εργασίας χρησιμοποιούμε το Cryptool, προκειμένου να κάνουμε την παραγοντοποίηση ορισμένων RSA moduli με διαφορετικό μέγεθος bit το κάθε ένα. Για κάθε μία περίπτωση, αναφέρεται ο χρόνος που παρήλθε έως την ολοκλήρωση της παραγοντοποίησης, όπως και τους δύο πρώτους αριθμούς  $p$  και  $q$ , στους οποίους παραγοντοποιείται το  $N$ .

Για την επίθεση παραγοντοποίησης στο Cryptool πηγαίνουμε στο μενού Analysis → Asymmetric Encryption → Factorization of a Number και στο παράθυρο που ανοίγει, εισάγουμε κάθε φορά τον αριθμό  $n$  και πατάμε «Complete factorization into numbers”.

- **$n = 2254841323226656761983237$  (80 bit modulo)**

Η παραγοντοποίηση του  $n$  (80 bit) είναι η εξής:

$$p = 1358962494041$$

$$q = 1659237346957$$

και ο χρόνος παραγοντοποίησης είναι 0.141 δευτερόλεπτα

Details for the Current Factorization

Input number:  
2254841323226656761983237

Factorized number	Factor 1	Factor 2	Method	Time
2254841323226656...	1659237346957	1358962494041	Quadratic sieve	0.141 seconds.

Information on the selected factorization (select by double-click on the row)

Factorized number:

First factor:

Second factor:

Save list into main window

Close

○  **$n = 940841942934961834804074337225577099057$  (128 bit)**

Η παραγοντοποίηση του  $n$  (128 bit) είναι η εξής:

$p = 28139159073535488133$

$q = 33435325500533930429$

και ο χρόνος παραγοντοποίησης είναι 3.412 δευτερόλεπτα



Details for the Current Factorization

Input number:

940841942934961834804074337225577099057

Factorized number	Factor 1	Factor 2	Method	Time
9408419429349618...	281391590735354...	334353255005339...	Quadratic sieve	3.412 seconds.

Information on the selected factorization (select by double-click on the row)

Factorized number:

First factor:

Second factor:

Save list into main window

Close

- **n**  
**=2895277316676774308077719327576371022454923546437 (160 bit)**

Η παραγοντοποίηση του n (160 bit) είναι η εξής:

p = 1507559604653228521407907

q = 1920506033552650825356791

και ο χρόνος παραγοντοποίησης είναι 18.670 δευτερόλεπτα

Details for the Current Factorization

Input number:  
2895277316676774308077719327576371022454923546437

Factorized number	Factor 1	Factor 2	Method	Time
2895277316676774...	150755960465322...	192050603355265...	Quadratic sieve	18.670 seconds.

Information on the selected factorization (select by double-click on the row)

Factorized number:

First factor:

Second factor:

Save list into main window

Close

- **n= 46447630664227253828834624577737400936236373765414  
123721785766493529824756761 (256 bit)**

Η παραγοντοποίηση του συγκεκριμένου RSA modulus των 256bit δεν ήταν επιτυχής στο μηχάνημα (χρειάζεται πάρα πολύ χρόνος για να σπάσει).

- **n =  
34949428219027603669916737263191942467425261103383711  
03696477768681664728709362878384114602968564594305134  
0117620310565676227110109274458253713189806502779 (512  
bit)**

Ομοίως, η παραγοντοποίηση του συγκεκριμένου RSA modulus των 512bit δεν ήταν επιτυχής στο μηχάνημα (χρειάζεται πάρα πολύ χρόνος για να σπάσει).

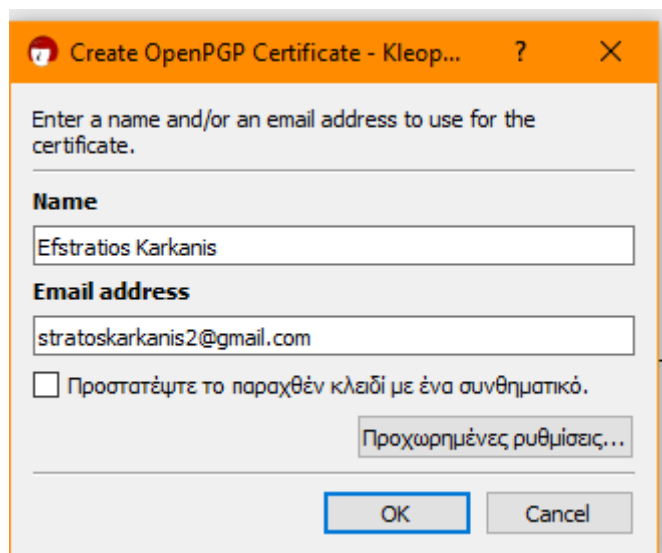
## 2. Μέρος (β) – Χρήση GPG

Στο συγκεκριμένο κομμάτι της εργασίας, ακολουθήσαμε (όλα τα μέλη της ομάδας) τα βήματα που περιγράφονται στις επόμενες ενότητες.

### 2.1 Δημιουργήστε ένα ζεύγος κλειδιών

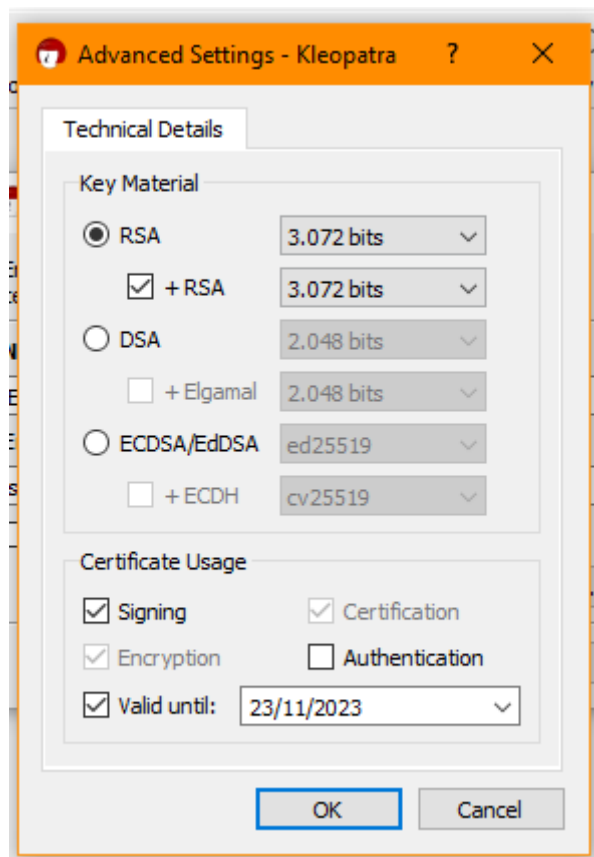
Το πρώτο βήμα, είναι η δημιουργία κλειδιών με τη χρήση του GPG, το οποίο θα έχει ημερομηνία λήξης 1 έτος.

Για να επιτευχθεί αυτό, πατάμε την αντίστοιχη επιλογή για δημιουργία ζεύγους κλειδιών. Στην συνέχεια, εισάγουμε τα στοιχεία μας:



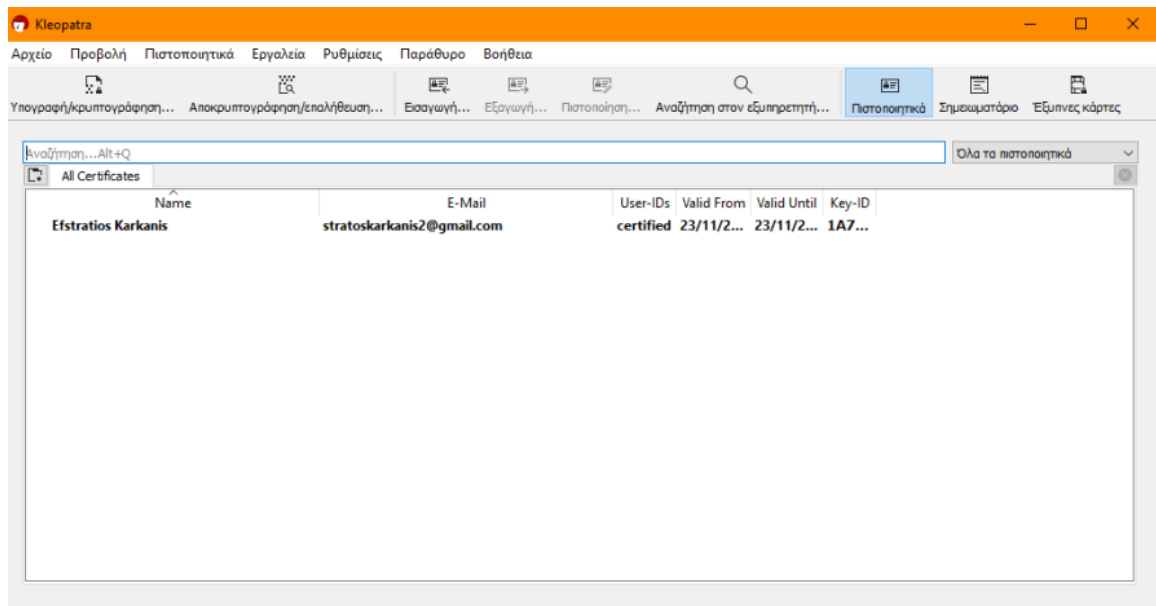
*Εισαγωγή στοιχείων*

Έπειτα, πατάμε το κουμπί «Προχωρημένες ρυθμίσεις», για να δώσουμε ημερομηνία λήξης στο κλειδί μας.



Εικόνα 1 Δημιουργία ζεύγους κλειδιών RSA 3.072 bits με ημερομηνία λήξης 23/11/2023

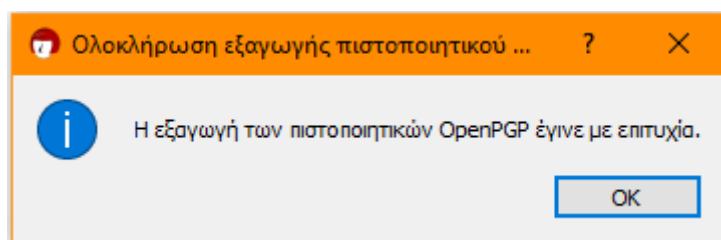
Πατώντας OK, το ζεύγος κλειδιών μας είναι πλέον έτοιμο και φαίνεται παρακάτω:



*Η δημιουργία ζεύγους κλειδιών ήταν επιτυχής!*

## 2.2 Ανέβασμα πιστοποιητικού σε server

Για να ανεβάσουμε το πιστοποιητικό που δημιουργήσαμε στο προηγούμενο βήμα σε έναν server κλειδιών, πατάμε δεξί κλικ πάνω στο κλειδί μας και μετά «Δημοσίευση στον εξυπηρετητή...». Μόλις το πιστοποιητικό ανέβει, εμφανίζεται το ακόλουθο μήνυμα στον χρήστη:

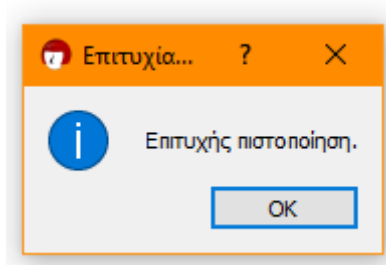


## 2.3 Εγκατάσταση και υπογραφή κλειδιών

Σε αυτό το βήμα, εφόσον όλα τα μέλη της ομάδας έχουν ανεβάσει (δημοσιεύσει) τα πιστοποιητικά τους στον key server, είναι δυνατή πλέον η αναζήτηση, η εγκατάσταση και η υπογραφή των κλειδιών αυτών με το δικό μας κλειδί.

Για να γίνει αυτό, για κάθε μέλος εκτελούμε τα ακόλουθα βήματα:

- 0 Μέσα στο πρόγραμμα, πατάμε την επιλογή «Αναζήτηση στον εξυπηρετητή...».
- 1 Εισάγουμε το όνομα του ατόμου, του οποίου το πιστοποιητικό θέλουμε να εγκαταστήσουμε.
- 2 Μόλις βρούμε το πιστοποιητικό, πατάμε την επιλογή «Εισαγωγή»
- 3 Για να υπογράψουμε το εν λόγω πιστοποιητικό, πατάμε στο επόμενο βήμα το κουμπί «Πιστοποίηση».
- 4 Στην συνέχεια, το ακόλουθο μήνυμα επιτυχίας εμφανίζεται στην οθόνη:

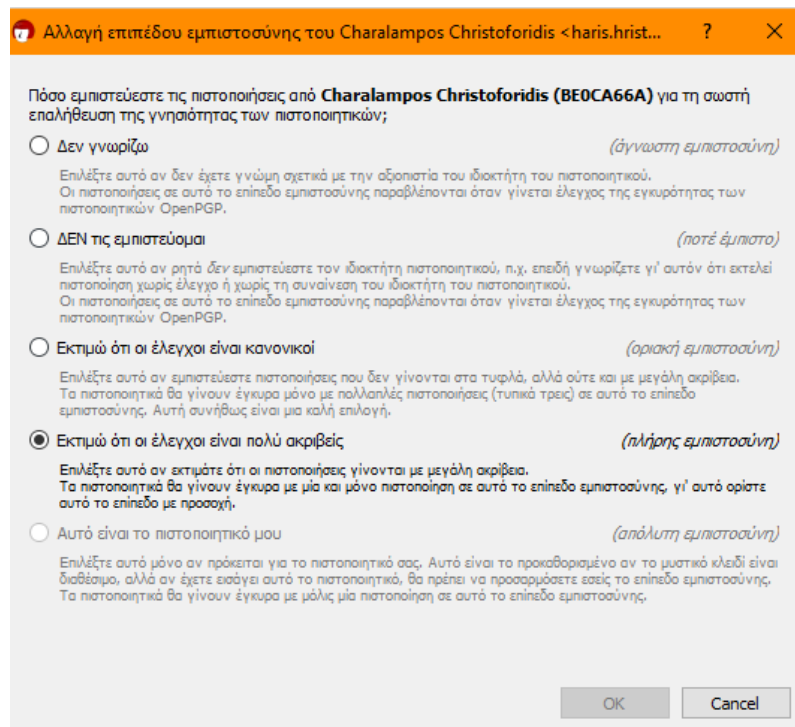


Μέχρι και αυτό το σημείο, έχουμε βρει, εγκαταστήσει και υπογράψει τα πιστοποιητικά όλων των μελών της ομάδας μας, τα οποία φαίνονται στην επόμενη εικόνα:

Search... <Alt+Q>						
All Certificates						
Name	E-Mail	User-IDs	Valid From	Valid Until	Key-ID	
Charalampos Christoforidis	haris.hristof@gmail.com	certified	23/11/2022	20/11/2023	C45C...	
<b>Nick Geo</b>	<b>nikosgeorgiadis2001@gmail.com</b>	<b>certified</b>	<b>22/11/2...</b>	<b>22/11/2...</b>	<b>315...</b>	
Efstratios Karkanis	stratoskarkanis2@gmail.com	certified	23/11/2022	23/11/2023	1A74...	

Για να αλλάξουμε το επίπεδο εμπιστοσύνης των πιστοποιητικών αυτών σε έμπιστα, ακολουθούμε (για κάθε ένα πιστοποιητικό που εισαγάγαμε) τα ακόλουθα βήματα:

- 0 Πατάμε δεξί κλικ στο πιστοποιητικό.
- 1 Πατάμε την επιλογή «Τροποποίηση εμπιστοσύνης πιστοποιητικού».
- 2 Πατάμε την επιλογή της πλήρους εμπιστοσύνης και μετά OK.



## 2.4 Αποστολή κρυπτογραφημένων email

Για την ολοκλήρωση του συγκεκριμένου ερωτήματος, χρησιμοποιήσαμε το plugin **Mailvelope** για τον gmail server.

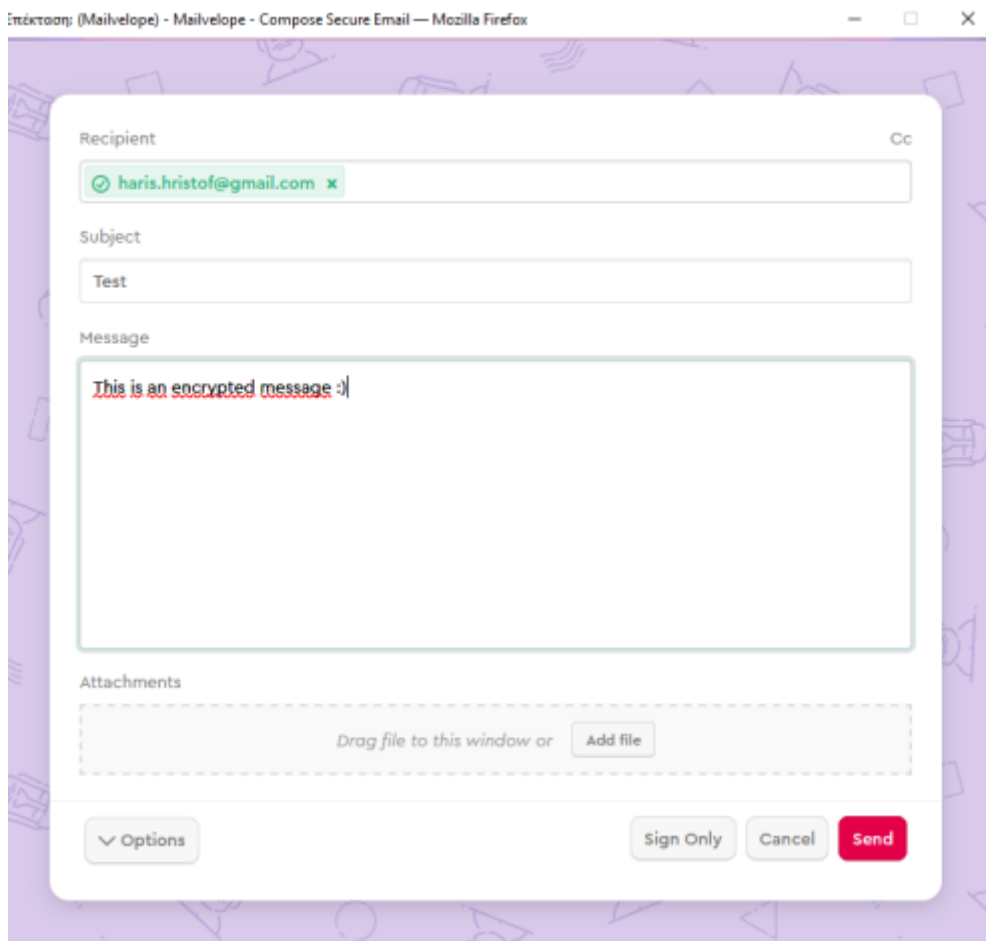
Σε αυτόν τον mail client εγκαταστήσαμε τα public keys των μελών της ομάδας. Επιπλέον, κάθε μέλος ξεχωριστά, ανέβασε και το δικό του private key στον client. Τα αποτελέσματα φαίνονται στην επόμενη εικόνα:

## Key Management

<div><div>+ Generate</div><div>⬇ Import</div><div>🔍 Search</div><div>⬆ Export</div><div>🔄 Refresh</div></div> <div>🔍 Filters: All ▾</div>				
	Name	Email	Key ID	Created
🔑	Charalampos Christoforidis	haris.hristof@gmail.com	C45CB0C4BE0CA66A	2022-11-23 >
👤	Efstratios Karkanis <span>Default</span>	stratoskarkanis2@gmail.com	1A74EC1E057F2D21	2022-11-23 >
🔑	Nick Geo	nikosgeorgiadis2001@gmail.com	3157661051AFCFE9	2022-11-22 >

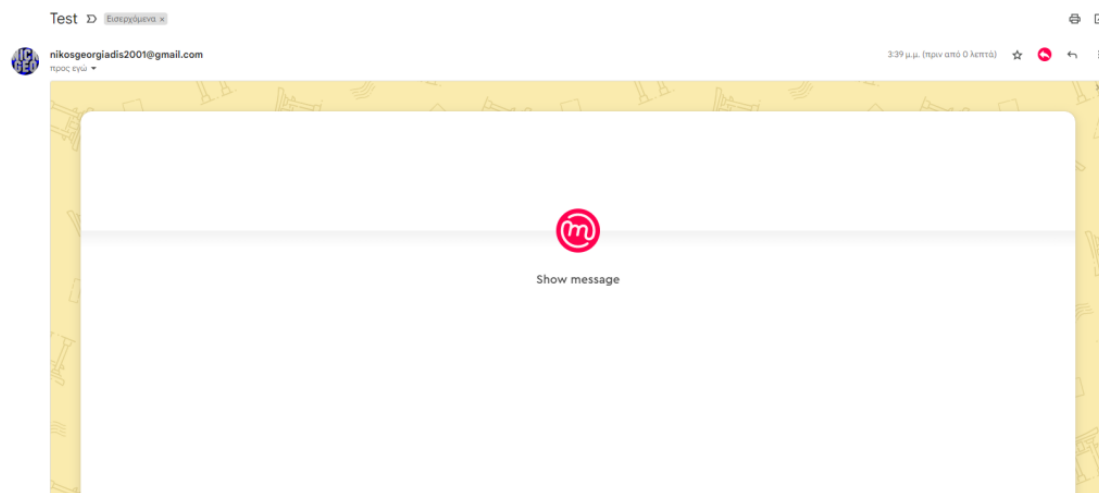
Στη συνέχεια, προσπαθήσαμε να στείλουμε ένα κρυπτογραφημένο email από ένα μέλος της ομάδας σε ένα άλλο, όπως φαίνεται παρακάτω:





Αποστολή κρυπτογραφημένου μηνύματος

Όπως φαίνεται και παρακάτω, το email στάλθηκε ως κρυπτογραφημένο όπως και αναμέναμε:



Κρυπτογραφημένο μήνυμα



Περιεχόμενο κρυπτογραφημένου μηνύματος