ΑΣΦΑΛΕΙΑ ΠΛΗΡΟΦΟΡΙΑΚΩΝ  
ΣΥΣΤΗΜΑΤΩΝ

ΕΡΓΑΣΙΑ 2

**Συντελεστές εργασίας**

Χριστοφορίδης Χαράλαμπος – Π19188

Γεωργιάδης Νικόλαος – π19032

Καρκάνης Ευστράτιος – Π19064

26 νοεμβριου 2022

**Περιεχόμενα**

[1. Εισαγωγή 2](#_Toc120102155)

[1. Μέρος (α) – CrypTool: 2](#_Toc120102156)

[1.1 Δημιουργία αρχείου txt 2](#_Toc120102157)

[1.2 Δημιουργία ενός ασύμμετρου ζεύγους κλειδιών RSA και προβολή πιστοποιητικού 2](#_Toc120102158)

[1.3 Κρυπτογράφηση κειμένου με υβριδική κρυπτογραφία RSA-AES 5](#_Toc120102159)

[1.4 Δημιουργία αρχείων txt 11](#_Toc120102160)

[1.5 Εύρεση επικίνδυνου μηνύματος 12](#_Toc120102161)

[1.6 Επίθεση στην τιμή hash και ψηφιακές υπογραφές 21](#_Toc120102162)

[1.7 Επίθεση παραγοντοποίησης σε RSA moduli 21](#_Toc120102163)

[2. Μέρος (β) – Χρήση GPG: 25](#_Toc120102164)

[1.1 25](#_Toc120102165)

[1.2 25](#_Toc120102166)

[1.3 25](#_Toc120102167)

1. **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:**

## **Δημιουργία αρχείου txt**

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

Text

Description automatically generated

Αρχείο txt

## **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:

Text

Description automatically generated with medium confidence

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), Keysize = 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

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

## **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.

Diagram

Description automatically generated

**Πριν το άνοιγμα του αρχείου**

Diagram

Description automatically generated

**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:

Diagram

Description automatically generated

**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:

Diagram

Description automatically generated

**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.¹HÄ¹[Ëm±MÔÞÄ(uã¶.

00014 73 47 CE 12 0E 36 7C F1 49 2C 8A 18 04 71 17 6F 14 7F 38 0D sGÎ..6|ñI,...q.o..8.

00028 A5 84 F5 39 59 ED 3A A1 C4 2E 8C AA F4 23 6B 29 6A 37 8D 2B ¥.õ9Yí:¡Ä..ªô#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.ÂÃ.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[¢Ìæ..<¶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 .¹J..f+üE¥KLK?ü{ðå.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 .eÄ­P.´.ðr&.{.Ñ?

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

Graphical user interface, text, application, email

Description automatically generated

**Επιλογή του RSA public key**

Graphical user interface, text, application, email

Description automatically generated

**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:

Diagram

Description automatically generated

Κρυπτογραφημένο session key με το public RSA key:



Πατώντας save εμφανίζεται το εξής παράθυρο:

Text

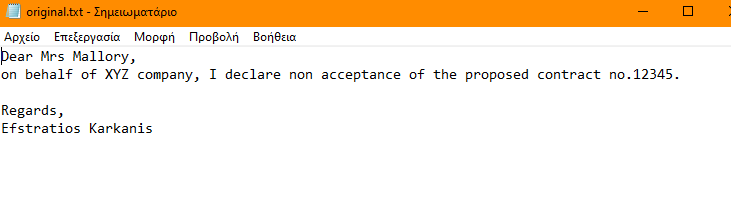
Description automatically generated

Αποτέλεσμα κρυπτογράφησης κειμένου και session key

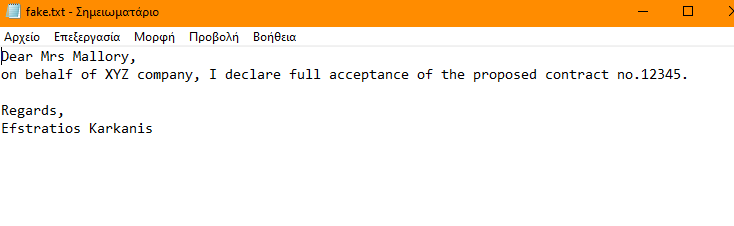
Να σημειωθεί ξανά, πως το session key κρυπτογραφείται με τον αλγόριθμο RSA χρησιμοποιώντας το public key που δημιουργήσαμε στο βήμα 2 και το μήνυμα κρυπτογραφείται με τον αλγόριθμο AES χρησιμοποιώντας το τυχαίο session key που δημιουργήσαμε σε αυτό το βήμα.

## **Δημιουργία αρχείων txt**

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



Αρχείο original.txt



Αρχείο fake.txt

## **Εύρεση επικίνδυνου μηνύματος**

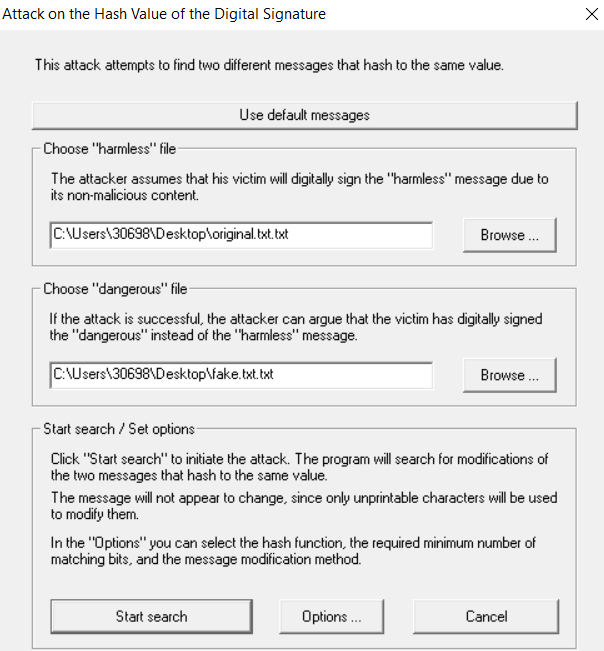
Στο συγκεκριμένο ερώτημα καλούμαστε, με βάση τα κείμενα που δημιουργήσαμε στο παραπάνω βήμα, να βρούμε ένα «επικίνδυνο» μήνυμα το οποίο μοιάζει στο fake.txt και να έχει την ίδια τιμή hash με το original.txt για τις παρακάτω περιπτώσεις:

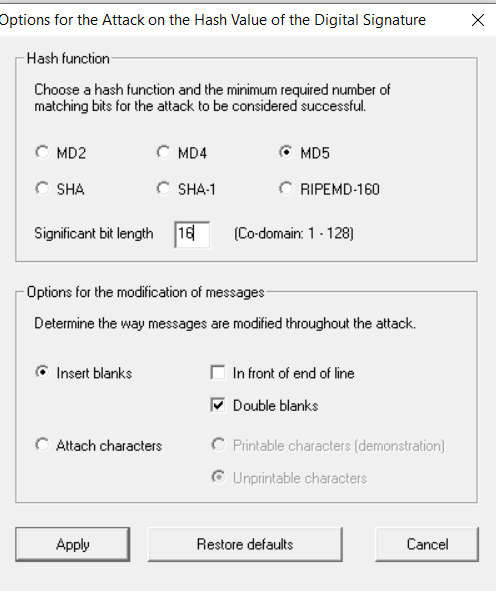
* Για τον αλγόριθμο MD5 και τα πρώτα 16 bit της τιμής hash.
* Για τον αλγόριθμο MD5 και τα πρώτα 50 bit της τιμής hash.
* Για τον αλγόριθμο SHA1 και τα πρώτα 80 bit της τιμής hash.
* Για τον αλγόριθμο SHA1 και όλα (160) τα bit της τιμής hash.

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 περίπτωσή μας πάντα τα original.txt και fake.txt).

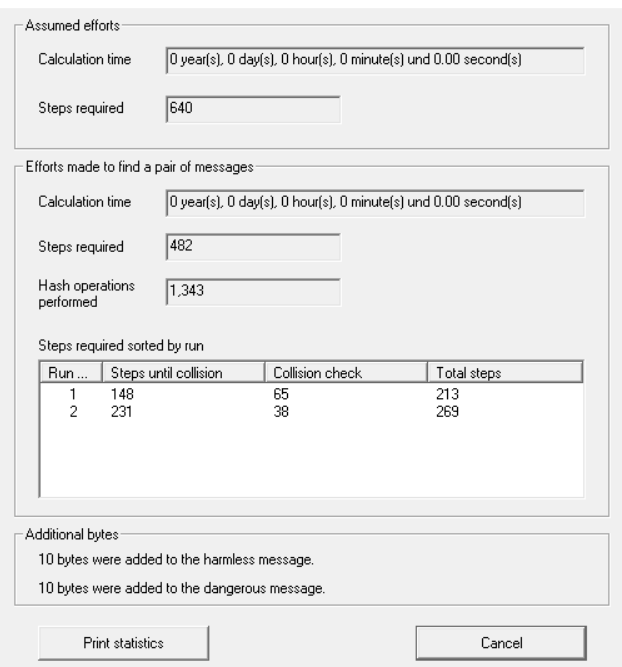
1. **Για τον αλγόριθμο MD5 και τα πρώτα 16 bit της τιμής hash.**

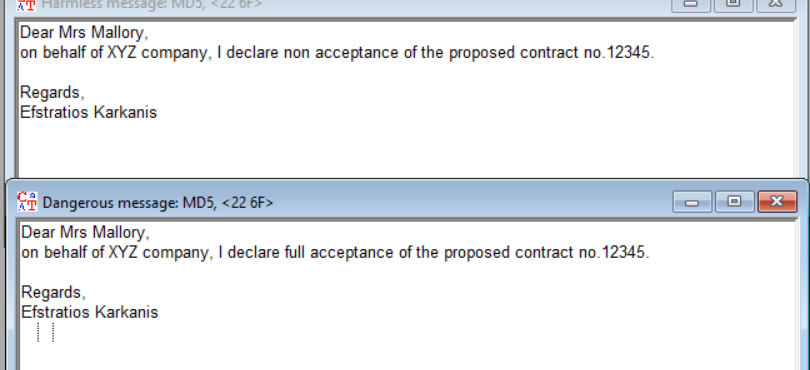
Αφού βάλουμε τις απαραίτητες παραμέτρους.





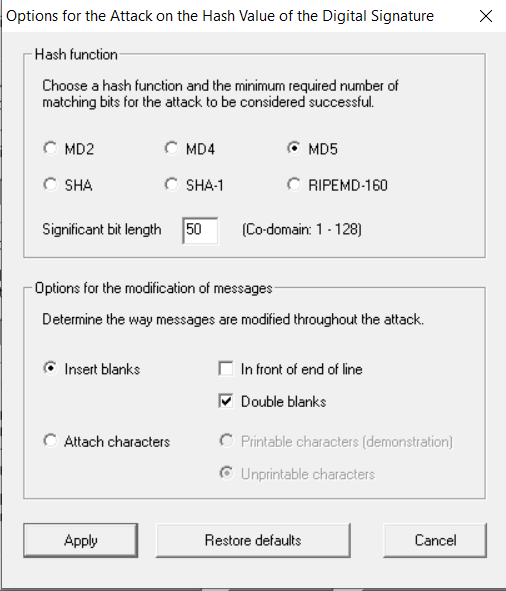
Πραγματοποιούμε την επίθεση. Ο χρόνος είναι σχεδόν μηδενικός.

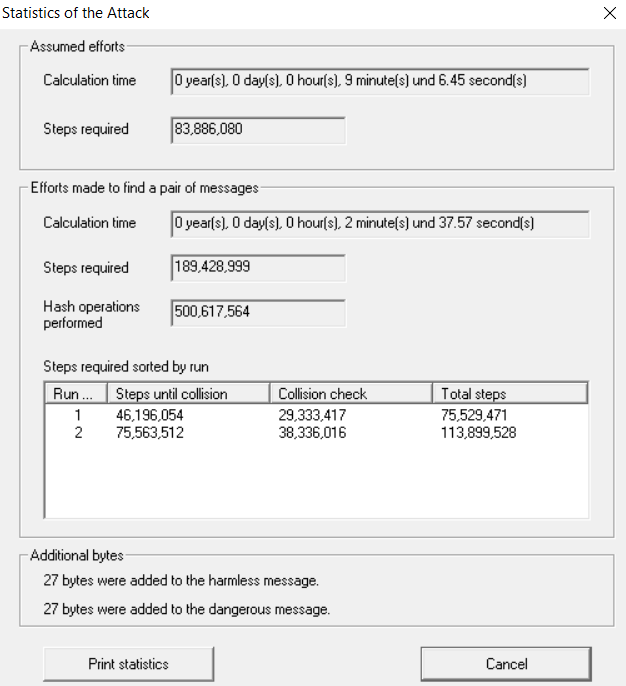


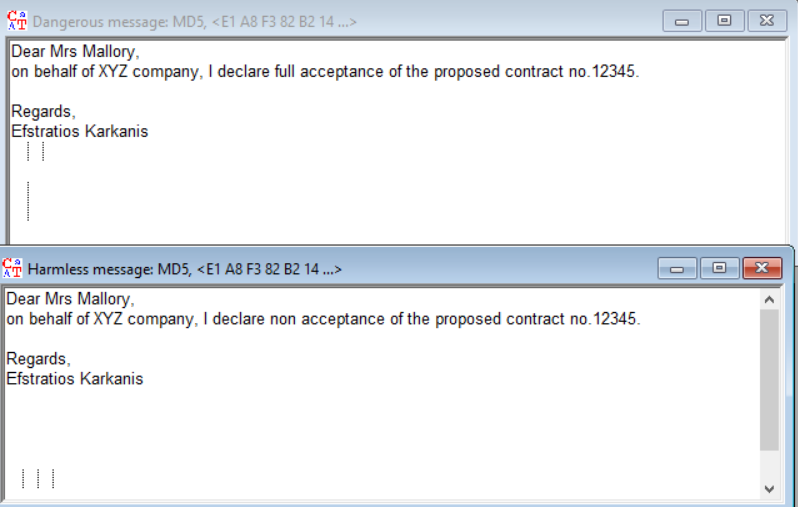


1. **Για τον αλγόριθμο MD5 και τα πρώτα 50 bit της τιμής hash.**

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

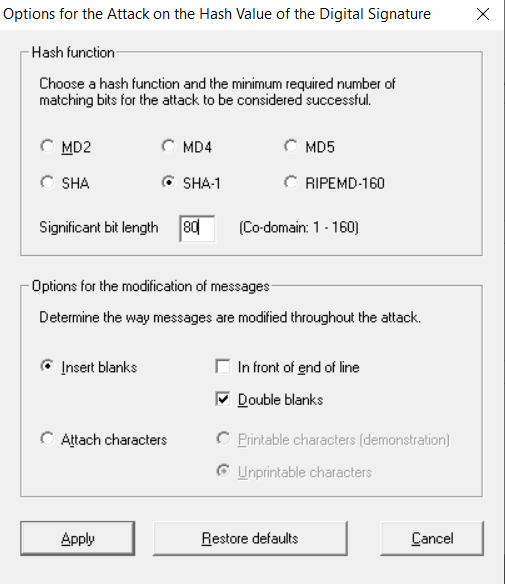


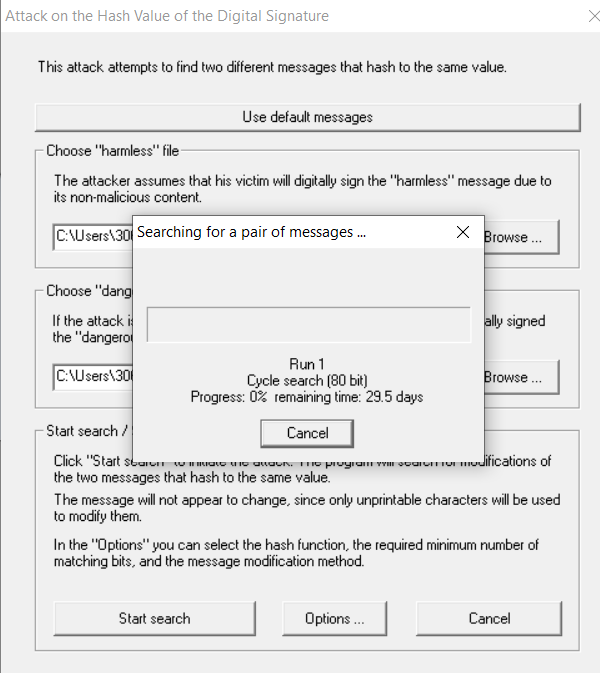




1. **Για τον αλγόριθμο SHA1 και τα πρώτα 80 bit της τιμής hash.**

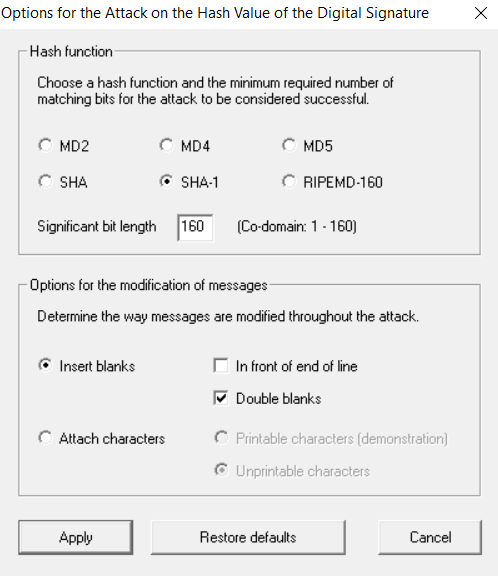
Φορτώνουμε και πάλι τα δεδομένα μας και αφού κάνουμε τις απαραίτητες αλλαγές εκτελούμε την επίθεσή μας, ο χρόνος ολοκλήρωσης ήταν περίπου 30 ημέρες.

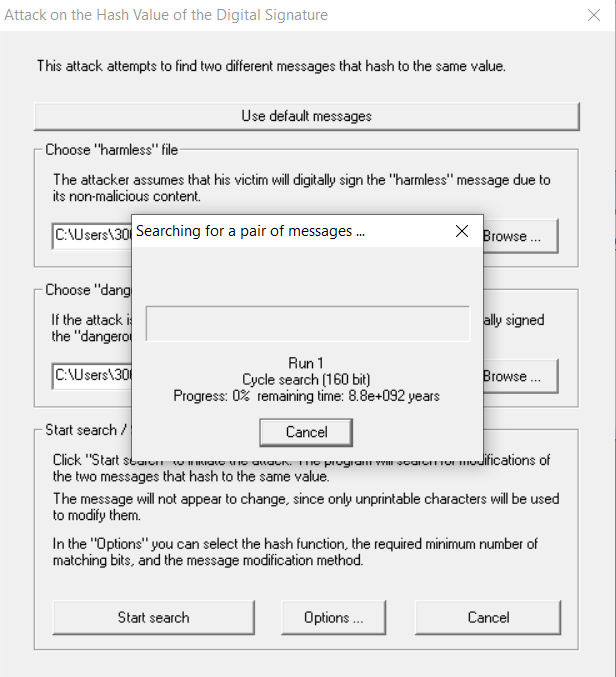




1. **Για τον αλγόριθμο SHA1 και όλα (160) τα bit της τιμής hash.**

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





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.

## **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 υπογράφει 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.

## **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 πραγοντοποιείται 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

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

* **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

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

* **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

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

* **n= 46447630664227253828834624577737400936236373765414123721785766493529824756761 (256 bit)**
* **n = 34949428219027603669916737263191942467425261103383711036964777686816647287093628783841146029685645943051340117620310565676227110109274458253713189806502779 (512 bit)**

1. **Μέρος (β)** – **Χρήση GPG**:

## 

## 

## 