ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη-Υλοποίηση Εργαστηριακών Ασκήσεων σε θέματα Κρυπτογραφίας και Ασφάλειας Υπολογιστών

Γιάννης Γεωργακόπουλος ΑΜ: 1346

Επιβλέπων Καθηγητής : Παρασκευάς Κίτσος

Αντίρριο Απρίλιος 2019

Ευχαριστίες

Αρχικά θέλω να ευχαριστήσω όλους τους καθηγητές μου για τις πολύτιμη συμβολή τους στην ολοκλήρωση των προπτυχιακών σπουδών μου που ολοκληρώνονται με την παρούσα εργασία. Πολλές ευχαριστίες οφείλω στους γονείς μου που χωρίς την αμέριστη υλική και ηθική τους υποστήριξη θα ήταν αδύνατον να καταφέρω τον σκοπό μου. Επίσης θέλω να ευχαριστήσω τον καθηγητή μου κ. Παρασκευά Κίτσο για την καθοδήγηση και την επίβλεψη της παρούσας πτυχιακής εργασίας

Περιεχόμενα

ΕΙΣΑΓΩΓΗ

Η προσπάθεια ανάπτυξης μεθόδου ώστε να εγγυάται την μη διαρροή μηνυμάτων δεν είναι κάτι νεό. Οι αρχαίοι Έλληνες χρησιμοποίησαν κρυπτογραφημένα μηνύματα για να στείλουν οδηγίες στους στρατούς τους στο πεδίο της μάχης. Οι αλγόρυθμοι όμως, τέθηκαν σε πραγματική λειτουργία και οι δυνατότητές του μελετήθηκαν διεξοδικά κατά τη διάρκεια των δύο παγκοσμίων πολέμων καθώς εκτός από την αποστολή των κρυπτογραφημένων μηνυμάτων, έπρεπε επίσης να εποκρυπτογραφηθούν τα μηνύματα του εχθρού. Η εξάπλωση των υπολογιστών και των συστημάτων επικοινωνίας τη δεκαετία του ’60 έφερε μαζί της την απαίτηση από τον ιδιωτικό τομέα για την ύπαρξη μέσων προσπασίας των πληροφοριών σε ψηφιακή μορφή και για την παροχή υπηρεσιών ασφαλείας.

Η λέξη κρυπρογραφία προέρχεται από τα συνθετικά “κρύπτος” + “γράφω” και είναι ο επιστημονικός κλάδος, ο οποίος ασχολείται με την μελέτη, τη νανάπτυξη και την χρήση τεχνικών προκειμένου να επιτευχθεί η κωδικοποίηση και αποκωδικοποίηση ενός μηνύματος πληροφορίας, ούτως ώστε να υπάρξει ασφαλή διακίνησή τους μεταξύ εξουσιοδοτημένων προσώπων και μόνο.

Κρυπτογράφηση ονομάζεται η διαδικασία με την οποία επιτυγχάνεται η παραπάνω μετατροπή του μηνύματος με τέτοιο τρόπο ώστε το περιεχόμενο του να παραμείνει μυστικό. Η αντίστροφη διαδικασία όπου από το μετασχηματισμένο μήνυμα πράγεται πάλι το αρχικό ονομάζεται αποκρυπτογράφηση. Η αρχική πληροφορία αποτελεί το αρχικό κείμενο ή ακρυπτογράφητο κείμενο, ενώ το αποτέλεσμα της κρυπτογράφησης ονομάζεται κρυπτογραφημένο ή κρυπτοκείμενο. Μια τεχνική ή ένας αλγόριθμος κρυπτογράφησης ενός μηνύματος, λέγεται κρυπτοσύστημα ή κρυπτογραφικό σύστημα. Κρυπτογραφικός αλγόριθμος είναι η μέθοδος που χρησιμοποιείται για τον μετασχηματισμό των δεδομένων σε τέτοια μορφή που να μην επιτρέπει την αποκάλυψη περιεχομένων τους σε μη εξουσιοδοτημένα μέρη. Κατά κανόνα ο κρυπτογραφικός αλγόριθμος είναι μια πολύπλοκη μαθημτική ή λογική συνάρτηση. Κλειδί είναι μια σειρά ψηφίων που χρησιμοποιείται ως είσοδος στην συνάρτηση κρυπτογράφησης και διαδραματίζει καθοριστικό ρόλο στην όλη διαδικασία. Καθορίζει τις ακριβείς αντικαταστάσεις και τα αποτελέσματα των μετασχηματισμών που

εκτελούνται από τον αλγόριθμο κρυπτογράφησης, άρα διαφορετικά κλειδιά παράγουν διαφορετικά κρυπτοκείμενα. Η κρυπτογράφηση και αποκρυπτογράφηση ενός μηνύματος γίνεται με την βοήθεια του αλγορίθμου κρυπτογράφησης και του κλειδιού κρυπτογράφησης. Συνήθως ο αλγόριθμος κρυπτογράφησης είναι γνωστός, άρα η ασφάλεια του περιεχομένου του κρυπτογραφημένου μηνύματος που μεταδίδεται βασίζεται ως επί το πλείστον στην μυστικότητα του κλειδιού κρυπτογράφησης. Το κλεδί έχει συγκεκριμένο μήκος και το μέγεθός του καθορίζεται από τον αριθμό των ψηφίων από τα οποία αποτελείται. Γενικά ισχύει ότι όσο μεγαλύτερο είναι το κλειδί κρυπτογράφησης, τόσο δυσκολότερα μπορεί να αποκρυπτογραφηθεί το κρυπτογραφημένο μήνυμα από μη εξουσιοδοτημένα άτομα.

Διαφορετικοί αλγόριθμοι κρυπτογράφησης απαιτούν διαφορετικά μήκη κλειδιών για να πετύχουν τοίδιο επίπεδο ανθεκτικότητας κρυπτογράφησης. Ο αλγόριθμος είναι συνήθως δημοσιοποιημένος, ενλω το κλειδί παραμένει μυστικό. Είναι γνωστό μόνο στον αποστολέα και στους αποδέκτες του το κρυπτοκείμενο, ώστε να είναι σε θέση να το μετατρέψουν σε μη κρυπτογραφημένο κείμενο.

Κρυπτανάλυση είναι ο κλάδος της επιστήμης που ασχολείται με την μελέτη και την επινόηση μεθόδων που εξασφαλίζουν την κατανόηση του νοήματος της κρυπτογραφημένης πληροφορίας, έχοντας ως άγνωστες ποσότητες τον κρυφό μετασχηματισμό, το κλειδί με βάση το οποίο αυτός πραγματοποιήθηκε και το κρυπτογραφημένο μήνυμα. Η κρυπτανάλυση και η κρυπτογραφία απαρτίζουν την επιστήμη της κρυπτολογίας.

Η ασφάλεια των υπολογιστικών συστημάτων και γενικότερα της πληροφορίας χρησιμοποιεί τα διάφορα κρυπτογραφικά συστήματα ώστε να εξασφαλίσει τους εξής στόχους: Εμπιστευτικότητα, Ακεραιότητα, Πιστοποίηση αυθεντικότητας και Μη αποποίηση.

Η ανθεκτικότητα ενός κρυπτογραφικού συστήματος, αποτελεί ένα από τα βασικά ζητήματα που πρέπει να πιστοποιηθεί πριν τη χρήσιμοποίησή του σε πρακτικές εφαρμογές. Τα κρυπτογραφικά συστήματα εμφανίζουν διάφορα επίπεδα ασφαλείας, ανάλογα με το πόσο δύσκολα παραβιάζονται. Για να θεωρηθεί ασφαλής ένας αλγόριθμος πρέπει να μην μπορεί να παραβιαστεί, δηλαδή κάποιος που δεν έχει εξουσιοδότηση να μπορεί να αποκαλύψει το κλειδί. Ένας αλγόριθμος είναι απόλυτα ασφαλής αν δεν

υπάρχει δυνατότητα να παραβιαστεί ανεξαρτήτως του μεγέθους του κρυπτογραφημένου μηνύματος, των υπολογιστικών πόρων και του χρόνου που μπορεί να διαθέτει ο κρυπταναλυτής. Ένας αλγόριθμος ονομάζεται υπολογιστικά ασφαλής αν είναι αδύνατη η παραβιασή του με τους διαθέσιμους πόρους.

Επιπροσθέτως, όπως θα δείτε στις παρακάτω σελίδες υπάρχουν δύο κεφάλαια, όπου στον ένα θα μελετήσουμε τις εκφωνήσεις των ασκήσεων και μετέπειτα στο άλλο κεφάλαιο θα δούμε την υλοποίηση αυτών. Όλες οι ασκήσεις δημιουργήθηκαν με γνώμονα την βοήθεια στην εκμάθηση των φοιτητών να διευρύνουν την γνώση πάνω στη κρυπτογραφία και στην ασφάλεια των υπολογιστών. Τέλος, τα προγράμματα που χρησιμοποιήσαμε για την υλοποίηση αυτών είναι τα εξής: CrypTool, Matlab, CrackStation και η πλατφόρμα OPENSSL.

ΜΕΡΟΣ ΠΡΩΤΟ : ΕΚΦΩΝΗΣΕΙΣ ΑΣΚΗΣΕΩΝ

2.1 ΑΣΚΗΣΗ 1

Ερώτημα 1

Στα πλαίσια του εργαστηρίου πρόκειται να κρυπτογραφηθεί και αποκρυπτογραφηθεί ένα κείμενο με τον αλγόριθμο του Καίσαρα.

England is a country that is part of the United Kingdom. It shares land borders with Scotland to the north and Wales to the west. The Irish Sea lies northwest of England and the Celtic Sea lies to the southwest. England is separated from continental Europe by the North Sea to the east and the English Channel to the south. The country covers five-eighths of the island of Great Britain (which lies in the North Atlantic) in its centre and south, and includes over 100 smaller named islands such as the Isles of Scilly and the Isle of Wight.

Α. Παρατηρήστε την αντιστοίχηση (Mapping) που παρουσιάζει το εργαλείο. Ποιά είναι αυτή;

Β. Πατώντας το κουμπί encrypt, παρατηρήστε το κρυπτογραφημένο κείμενο που προκύπτει.

Γ. Αποκρυπτογραφήστε το κρυπτογραφημένο κείμενο χρησιμοποιώντας το ίδιο κλειδί.

Παρατήρηση: Μπορείτε να αποβάλετε τα κενά αλλά και τα πεζά γράμματα από το αρχικό κείμενο ώστε να γίνει πιο ομαλό προς κρυπτογράφηση από το menuoptions/TextOptions και απενεργοποιώντας το **Keepcharactersnotpresentinthealphabetunchanged**και **Distinguishbetweenuppercaseandlowercase**.

2.2 ΑΣΚΗΣΗ 2

Ανάλυση συχνότητας εμφάνισης των γραμμάτων

Για το παρακάτω καταγράψτε τις συχνότητες εμφάνισης των 26 χαρακτήρων του αγγλικού αλφαβήτου που εμφανίζονται σε αυτό (με το εργαλείο CrypTool – επιλογή «Analysis –>ToolsforAnalysis ->Histogram» ή «Analysis –>ToolsforAnalysis ->N- gram»). Συγκρίνετε το αποτέλεσμα με τις τυπικές συχνότητες εμφάνισης των γραμμάτων του αγγλικού αλφαβήτου και καταγράψτε τα συμπεράσματά σας.

Κείμενο

1 Introduction

The "Internet of things" is a revolution for the ICT world. Devices, system components and networks are becoming autonomous, ubiquitous and interconnected. When this technological advancement applies to the healthcare sectors, one of the most traditional critical sectors1 , the results are remarkable. Connected medical devices transform the way the healthcare industry works, both within hospitals and between different actors of the healthcare industry. Could you imagine an electronic device collecting information on patients' vital signs becoming "smart"? Or one that monitors life supporting machines to be able to react on any change of status? Connected medical devices can bring increased patient safety and efficiency, particularly if connected to Clinical information systems. When this applies to the whole healthcare organisation ecosystem, it becomes a "Smart Hospital".

However, the increased flow of information within and between hospitals brings risks that C-level professionals in the hospital (CIO, CISO etc.) need to address. The risks include possible harm to patient safety or loss of personal health information and may not only be caused by malicious actions but also by human errors, system or third-party failures and natural phenomena. As the attack surface increases with the introduction of connected devices, the attack potential grows exponentially.

1.1 Objective and scope

The objective of this study is to improve information security and resilience of hospitals to prevent disruptions to smart components that can cause

greater impact to patients' safety. The ultimate goal is to offer enhanced patient safety.

This study investigates the current status of Smart Hospitals and related information security issues, focusing on deployments in the EU. This involves determining the objectives achieved through "smart" devices and systems, the assets that make up a Smart Hospital, the information security threats as well as the security measures available to address them. Through gap identification between current threats and existing measures, this study makes concrete recommendations to improve information security in smart hospitals.

The focus of the study is the hospital itself and specifically on all the smart components that are offering value when built on top of already existing traditional systems

1.2 Methodology

This report was developed using a combination of desktop research as well as information from interviews with key stakeholders. The document analysis focuses on scientific, as well as industry and policy material, related to information security in smart hospitals. The interviews and the survey were conducted to validate and extend the findings of the document analysis.

The approach taken follows the ENISA methodology2 developed over the last three years based on the ENISA threat landscape approach, and involved:

- Mapping assets and developing a threat taxonomy that covers possible attacks via desktop research, and validating or identifying further gaps through interviews with security experts working in the field of healthcare information security, focusing on Hospitals.
- The assets are categorised based on their criticality, meaning the impact an incident in one of these could cause.
- Enumerating possible attacks that target or affect smart components in hospitals.
- Developing three attack scenarios with mitigation actions to provide information on practical examples of implementation, and validating these with security experts working in Hospitals.

- Developing good practices and performing a gap analysis based on desktop research and interviews.
- Proposing recommendations for future steps in information security for Smart Hospitals in Europe.

The term "asset" has two slightly different meanings in the information security context. In some cases the term is used to refer to the mostly technical components of an organisational information system. Such components allow organisations to meet their objectives but differ from each other with regard to their criticality. In other cases, the term is used more broadly to refer to organisational values that need to be protected. The protection of such values is sometimes an objective in itself.

Thirty experts participated in the interviews and the survey. Participants were hospital representatives, industry representatives and policy makers. Figure 2 depicts the distribution of participants across the three groups. All were able to draw on several years of experience with Information and Communication Technology (ICT) in healthcare and held senior positions.

1.3 Target Audience

The target audience of this study is executives and C-level professionals from hospitals. The aim is to help them to understand which are the steps they need to take to ensure information security when choosing "smart" solutions. IT and security professionals are of particular relevance (e.g. Chief Medical Information Officers, Chief Information Security Officers (CISOs)).

As a secure "Smart hospital" design has extensions to devices and systems security, this document could be useful also (but not only) for:

- Industry representatives: Executives and professionals of manufacturers of connected devices for healthcare are relevant with respect to industry representatives as well as technology and consulting companies focused on information security.
- Policy makers: Policy makers from Member States and the European Union (EU) are relevant if they are in charge of policies dealing with healthcare, critical infrastructures or information security.

1.4 Structure

The study is structured as follows:

Section 2 describes the smart hospital environment, paying particular attention to the definition of the term, the regulatory framework and guidelines related to information security, the objectives hospitals pursue and the effect of being "smart" on these objectives, and the key assets to be protected.

Section 3 pursues an asset-centric approach to threat and risk analysis. Based on the key assets and a vulnerabilities, potential attack points and threat types are discussed.

Section 4 describes five attack scenarios ranging from social engineering attacks on hospital staff to distributed denial-of-service attacks on hospital servers.

Section 5 describes the control and recovery measures available to protect the smart hospital from the threats faced. A differentiation is made between measures to be implemented by hospitals and the industry, respectively.

Section 6 makes concrete and actionable recommendations aimed at hospital executives, industry representatives and policy makers. Additionally, examples of good practice are described.

1.5 Smart Hospitals

This section is split in two parts. The first part describes the smart hospital environment, placing emphasis on the definition of the term "smart hospital", the objectives of introducing "smartness" in a hospital environment, the guidelines related to information security and the respective regulatory framework. The second part focuses on the assets that introduce "smartness" in the hospital environment and need to be protected due to their criticality for the operation of smart hospitals.

1.6 The Smart Hospital Environment

The overarching goal of smart hospitals is to deliver optimal patient care by making the most of advanced ICT. The availability of all relevant information when required; access to internal and external expertise when needed; and efficient and effective surgical/diagnosis processes that facilitates achieving this goal with low error rate and cost effectively.

A definition of the term "smart hospitals" may thus be:

"A smart hospital is a hospital that relies on optimised and automated processes built on an ICT environment of interconnected assets, particularly based on Internet of things (IoT), to improve existing patient care procedures and introduce new capabilities".....»

Πηγή : ENISA (European Union Agency for Network and Information Security) ,
Smart Hospitals Security and Resilience for Smart Health Service and Infrastructures , November 2016 , www.enisa.europa.eu

2.3 ΑΣΚΗΣΗ 3

Κρυπτανάλυση μονοαλφαβητικής αντικατάστασης

Θεωρείστε το κρυπτογραφημένο κείμενο που δίνεται παρακάτω, το οποίο έχει κρυπτογραφηθεί με αλγόριθμο μονοαλφαβητικής αντικατάστασης.

Α) Προσπαθήστε, μέσω του CrypTool, να επιχειρήσετε κρυπτανάλυση βάσει της συχνότητας εμφάνισης των γραμμάτων, με τους αυτοματοποιημένους τρόπους που διαθέτει το Cryptool (από την επιλογή "Analysis ->SymmetricEncryption (classic) ->Ciphertextonly ->Substitution"). Μπορεί το πρόγραμμα να ανακαλύψει το αρχικό μήνυμα, με βάση αυτό που παρατηρείτε; Δικαιολογείστε την απάντησή σας.

Κρυπτοκείμενο :

GrgnXyhpnefgjguhprrpgnxbgxambgxpopgn. UchnpnRcnvcnpn 1912, amjxyvpmvgxucxaOgbuhpvemgnvShpnomxcnynpimhjpxpmj. Amfgjgrhmgvlfchwpnesghx-xpbmzchxamUhpxpjaEcimhnbmnx'jOcvmgnvOlsamhJoaccrumzchmxamJmocnvFchrvFghuhcwmcyx. Pn 1939, Xyhpnexccwysgzyrr-xpbmhcrmgxUrmxoarmlSghw – famhmxcsjmohmxfchwfgjoghhpmvcyxxcvmopsamhxambprpxghlocvmjyjmvulEmhbgnlgnvpxjgrrpmj. XambgpnzcoyjczXyhpne'jfchwgxUrmxoarmlfgjpnohgowpnexamMnpebgocvm. Xyhpnesrglmvgwmlhcrmpnxapj, pnimnxpne – grcnefpxazmrrcfocvm-uhmgwmhEchvcnFmroabgn – gbgoapnmwncfngjxamUcbum. Xapjvmipomamrsmvxcjpenpzpognxrlhmvyomxamfchwczxamocvm-uhmgwmhj. Zhcbbpv-1940, EmhbgnGphZchomjpengrjfmhmumpnehmgvgxUrmxoarmlgnvxampnxmrrpemnomegpnmvzhcbxambfgjamrspnexamfghmzzchx.

Β) Προσπαθήστε εκ νέου να ανακτήσετε το μήνυμα, αν ξέρετε ότι είναι ένα κείμενο στα αγγλικά που αφορά τον AlanTuring, οπότε και η λέξη TURING αναμένεται να υπάρχει εντός του κειμένου (και, ενδεχομένως, περισσότερες από μία φορές). Φυσικά, αν κρίνετε ότι βοηθάει, μπορείτε να εφαρμόσετε και όλες τις άλλες υποθέσεις (περί του πιο συχνού γράμματος κτλ.) Για την βήμα-βήμα ανάλυσή σας, χρησιμοποιείστε την επιλογή "Analysis ->SymmetricEncryption (classic) ->Manualanalysis ->Substitution". Για την εύρεση ομάδων γραμμάτων που εμφανίζονται πολλές φορές στο κείμενο, μπορείτε να χρησιμοποιήσετε την επιλογή «Analysis ->ToolsforAnalysis ->N-gram) και να επιλέξετε όποια τιμή

του Ν επιθυμείτε. Να καταγράψετε στην απάντησή σας αναλυτικά τη διαδικασία που ακολουθήσατε.

2.4 ΑΣΚΗΣΗ 4

Στο ερώτημα αυτό θα καταδείξετε ότι και στον αλγόριθμο Vigenere μία επίθεση γνωστού μηνύματος (knownplaintextattack) μπορεί να είναι αποτελεσματική. Συγκεκριμένα, για το κρυπτογραφημένο με τον αλγόριθμο Vigenere μήνυμα «wvagfrrnfveifvailzx», γνωρίζετε ότι πρόκειται για μία επιστολή προς τη Mary. Συνεπώς, κάνετε την υπόθεση ότι το αρχικό μήνυμα ξεκινά με τους χαρακτήρες

«dearmary». Μήπως μπορείτε, αξιοποιώντας την πληροφορία αυτή, να ανακαλύψετε το μυστικό κλειδί και, ακολούθως, ολόκληρο το αρχικό μήνυμα;

(Υπόδειξη: Μπορείτε να αξιοποιήσετε το Cryptool για τους υπολογισμούς σας. Για κρυπτογραφήσεις/αποκρυπτογραφήσεις με Vigenere, επιλέγετε: "Encrypt/Decrypt ->Symmetric (classic) ->Vigenere"). Δημιουργήστε ένα αρχείο με το κρυπτογραφημένο μήνυμα μορφής .txt για διευκόλυνσή σας.

ΚΕΦΑΛΑΙΟ 3ο : ΜΟΝΤΕΡΝΑ ΚΡΥΠΤΟΓΡΑΦΙΑ ΣΥΜΜΕΤΡΙΚΟΥ ΚΛΕΙΔΙΟΥ

3.1 ΑΣΚΗΣΗ 1

Να καταδείξετε το λεγόμενο «avalancheeffect» του DES, ως εξής: Αφού επιλέξετε δικό σας μήνυμα μεγέθους 8 χαρακτήρων (όσο και ένα

blocktou DES) και το κρυπτογραφήσετε με κλειδί της επιλογής σας, στη συνέχεια να κάνετε τα εξής: i) Αλλάξτε μόνο ένα bit του κλειδιού (όποιο επιθυμείτε) και κρυπτογραφείστε εκ νέου το μήνυμα, ii) Αλλάξτε μόνο ένα bit του μηνύματος (όποιο επιθυμείτε) και κρυπτογραφείστε εκ νέου το τροποποιημένο αυτό μήνυμα, με το αρχικό κλειδί. Σχολιάστε τα αποτελέσματα.

(*Υπόδειξη: Για να συγκρίνετε δύο κρυπτοκείμενα, δενθα πρέπει να σταθείτε στην «οπτική» σύγκρισή τους, αλλά να δείτε συγκεκριμένα σε πόσεςθέσεις bit, από τις συνολικά 64, διαφέρουν. Από την επιλογή «View» του Cryptool 1.4.41, μπορείτε να επιλέξετε να δείτε ένα μήνυμα ASCII χαρακτήρων σε 16-δική αναπαράστασητων bytes αυτού και αντίστροφα, σε περίπτωση που το χρειαστείτε).*

3.2 ΑΣΚΗΣΗ 2

Α) Κρυπτογραφείστε το ονοματεπώνυμό σας με τον αλγόριθμο DES μέσω του Cryptool 1.4.30 ("Encrypt/Decrypt ->Symmetric(modern) ->DES(ECB)") με κλειδί της επιλογής σας, δύο διαδοχικές φορές - δηλαδή θα καλέσετε δύο διαδοχικές φορές τη συνάρτηση κρυπτογράφησης του DES, έτσι ώστε για μήνυμα m και κλειδί k να υπολογίσετε το $DES_k(DES_k(m))$, όπου $DES_k()$ η κρυπτογράφηση με αλγόριθμο DES και κλειδί k.

Β) Επαναλάβετε το ερώτημα 3α, όπου όμως ως κλειδί k επιλέξτε ένα (οποιοδήποτε) εκ των τεσσάρων «αδύναμων» (weak) κλειδιών :
0101 01 01 01 01 01 01
FE FE FE FE FE FE FE FE
E0 E0 E0 E0 F1 F1 F1 F1
1F 1F 1F 1F 0E 0E 0E 0E

Τι παρατηρείτε;

3.3 ΑΣΚΗΣΗ 3

Επιλέξτε κατάλληλα δικά σας μηνύματα κειμένου και, χρησιμοποιώντας το λογισμικό Cryptool 1.4.41, αναδείξτε τη βασική διαφορά μεταξύ του ECB και του CBC τρόπου λειτουργίας στον DES αλλά και στον 3DES, δηλαδή να αναδείξετε ότι όμοια blocks στο αρχικό μήνυμα κρυπτογραφούνται σε όμοια blocks στην περίπτωση ECB, αλλά σε διαφορετικά blocks στην περίπτωση CBC).

3.4 ΑΣΚΗΣΗ 4

Να καταδείξετε, με κατάλληλο δικό σας παράδειγμα στο Cryptool, ότι αν
μεταβληθεί έστω και ένα bit σε ένα μήνυμα σε οποιαδήποτε θέση και συγκρίνουμε, για αυτά τα δύο ελαφρώς διαφορετικά μηνύματα, τα αντίστοιχα
κρυπτοκείμενα που παράγονται με τον DES σε CBC τρόπο λειτουργίας, τότε
το τελευταίο block των δύο κρυπτοκειμένων είναι τελείως διαφορετικό (δηλαδή να αναδείξετε την ιδιότητα που περιγράφεται στη διαφάνεια 24 του
μαθήματος). Να πράξετε το ίδιο, με κατάλληλο παράδειγμα, και για τον 3DES
αλλά και για τον AES.

(Και στα δύο ανωτέρω ερωτήματα, ως κλειδί μπορείτε να επιλέξετε όποιο
θέλετε).

ΚΕΦΑΛΑΙΟ 5 : ΣΥΝΑΡΤΗΣΕΙΣ ΚΑΤΑΚΕΡΜΑΤΙΣΜΟΥ ΚΑΙ HMAC.

5.1 ΑΣΚΗΣΗ 1

Ο συνεργάτης σας προτείνει, ως συνάρτηση κατακερματισμού h, μία συνάρτηση η οποία δέχεται ως είσοδο ένα οποιοδήποτε μήνυμα M=m0m1m2…. αυθαίρετου μεγέθους και το αποτύπωμά του h(M) αποτελείται από 128 bits, υπολογιζόμενα ως εξής: m0, m2, m4, m6 , ….. , m254 (σε περίπτωση που το M αποτελείται από λιγότερα από 255 ψηφία, «συμπληρώνεται» κατάλληλα με μηδενικά). Περιγράψτε αν είναι μία καλή συνάρτηση κατακερματισμού, δηλαδή αν πληροί όλες τις επιθυμητές ιδιότητες.

5.2 ΑΣΚΗΣΗ 2

Για τα δύο παρακάτω μηνύματα :

message1.txt :

Dear Tom,

I hope this e-mail finds you well.

I would like to ask you a favor: Please give John 10000 $ from my account. I think that he deserves this.

Kind Regards

Και **message2.txt :**

Dear Tom,

I hope this e-mail finds you well.

I would like to ask you a favor: Please give Bob 100000 $ from my account. I

thinkthathedeservesthis.

KindRegards

Μέσω του Cryptool 1.4.41, να διαπιστώσετε ότι μπορούν να τροποποιηθούν
κατάλληλα τα αρχεία ώστε αφενός να μην αλλάζει το περιεχόμενό τους (αυτό
σημαίνει ότι η τροποποίηση των μηνυμάτων σημαίνει εισαγωγή κενών ή μη
εκτυπώσιμων χαρακτήρων) και αφετέρου τα αποτυπώματά τους να
ταυτίζονται σε ένα πλήθος bits που ορίζει ο χρήστης (8,16,32,64 bits κ.ο.κ.).
Για να το δείτε αυτό, αξιοποιήστε την επιλογή «Analysis» -> «Hash» ->
«Attacks on the hash values of the digital signatures», όπου στη συνέχεια, με
την επιλογή «options", μπορείτε να επιλέξετε SHA-1 και πλήθος bits ταύτισης
στα αποτυπώματα 8, 16 και 32 bit αντίστοιχα. Αναμένεται να δείτε ότι σε
μικρό χρονικό διάστημα μπορούν εύκολα να τροποποιηθούν τα μηνύματα
ώστε τα αποτύπωματά τους να ταυτίζονται στον εκάστοτε αριθμό θέσεων που
έχει κάθε φορά επιλεγεί (τον υπολογισμό θα τον εκτελέσει αυτόματα το Cryptool, επιλέγοντας «start search»).

5.3 ΑΣΚΗΣΗ 3

Ποιες από τις ακόλουθες προτάσεις θεωρείτε ότι αποτελούν καλές ιδέες για
την κατασκευή ενός MAC και ποιες όχι; Εξηγείστε την απάντησή σας.

Α) Κρυπτογράφηση του μηνύματος εισόδου Μ με τον AES σε ECB τρόπο
λειτουργίας, και ο MAC θα προκύπτει λαμβάνοντας το τελευταίο block του
παραγόμενου κρυπτοκειμένου.
Β) Κρυπτογράφηση του μηνύματος εισόδου Μ με τον RC4 και ο MAC θα
προκύπτει λαμβάνοντας τα τελευταία 256 bit του παραγόμενου
κρυπτοκειμένου.
Γ) Κρυπτογράφηση του μηνύματος εισόδου Μ με τον AES σε CBC τρόπο
λειτουργίας, και ο MAC θα προκύπτει λαμβάνοντας το δεύτερο block του
παραγόμενου κρυπτοκειμένου.

ΚΕΦΑΛΑΙΟ 6 : ΘΕΩΡΙΑ ΑΡΙΘΜΩΝ ΓΙΑ ΚΡΥΠΤΟΓΡΑΦΙΑ

6.1 ΑΣΚΗΣΗ 1

Να κάνετε τους ακόλουθους υπολογισμούς (για κάθε πράξη modn, το αποτέλεσμα θα πρέπει να είναι ένας αριθμός μεταξύ 0 και n-1).
(21+ 12) mod 30,
(6-10) mod 13,
(10-2*12) mod 21

6.2 ΑΣΚΗΣΗ 2

Το σύνολο {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16}, εφοδιασμένο με τις πράξεις της πρόσθεσης και του πολλαπλασιασμού mod 17, αποτελεί πεπερασμένο σώμα; Αν ναι, επιλέξτε δύο οποιαδήποτε διαφορετικά στοιχεία a,b του σώματος (μεγαλύτερα από το 10) και εκτελέστε τις πράξεις a+b, a-b, a*b, a/b (όπου όλοι οι τελεστές είναι mod 13).
(Υπόδειξη: Εφόσον χρειαστείτε κάποιον modular αντίστροφο, μπορείτε να
χρησιμοποιήσετε τον διαδικτυακό τόπο http://www.dcode.fr/modularinverse)

ΚΕΦΑΛΑΙΟ 7 : ΚΡΥΠΤΟΓΡΑΦΙΑ ΔΗΜΟΣΙΟΥ ΚΛΕΙΔΙΟΥ

7.1 ΑΣΚΗΣΗ 1

Α) Η ασφάλεια του Diffie-Hellman έγκειται στο πρόβλημα διακριτού λογαρίθμου
(DLP), το οποίο για πολύ μεγάλες τιμές του p δεν μπορεί να επιλυθεί με αποδοτικό αλγόριθμο: αν ωστόσο ο p είναι μικρός αριθμός, τότε δεν παρέχεται καμία ασφάλεια. Για να το καταδείξετε αυτό, θεωρείστε ένα σχήμα Diffie- Hellman με p=23, g=20, όπου ο Α στέλνει στον Β τον αριθμό x'=10 και ο Β στον Α τον αριθμό y'=16. Βρείτε το μυστικό κλειδί που αντάλλαξαν οι δύο χρήστες.

Β) Επιβεβαιώστε τα ανωτέρω μέσω του κατάλληλου εργαλείου προσομοίωσης του Cryptool (επιλογή IndividualProcedures ->Protocols – Diffie – Hellmandemonstration). Θα πρέπει, στο βήμα α) ανωτέρω, να

έχετε βρει και τα δύο ιδιωτικά κλειδιά – τόσο του Α όσο και του Β – για να τα θέσετε ως είσοδο, μαζί με τα g, p, στο εργαλείο προσομοίωσης, για να καταδείξετε ότι θα αποσταλούν πράγματι οι τιμές x'=10 και y'=16, καθώς επίσης και να επιβεβαιώσετε ότι το μυστικό κλειδί που τελικά αντήλλαξαν είναι αυτό που βρήκατε στο ερώτημα Α.

Γ) Επιχειρήστε να δημιουργήσετε, μέσω του Cryptool, μία νέα επιλογή παραμέτρων για την υλοποίηση του πρωτοκόλλου Diffie-Hellman, διατηρώντας
το ίδιο p=23 αλλά επιλέγοντας g=8. Τι παρατηρείτε; Μπορείτε να το εξηγήσετε;

7.2 ΑΣΚΗΣΗ 2

Σε ένα κρυπτοσύστημα RSA, ο χρήστης Α έχει δημόσιο κλειδί e=7, N=5371, ενώ ο χρήστης Β έχει δημόσιο κλειδί e=5, N=2581.

Α) Δείξτε ότι μπορείτε να ανακτήσετε τα ιδιωτικά κλειδιά και των δύο χρηστών,

ακριβώς λόγω των μικρών δημόσιων κλειδιών N που έχουν επιλεγεί.

(*Υπόδειξη: Για την παραγοντοποίηση ακέραιου αριθμού, μπορείτε να χρησιμοποιήσετε το Cryptool 1.4.41, επιλογή «IndividualProcedures ->RSADemonstration ->Factorizationofanumber»).*

Β) Παρακολουθείτε το κανάλι μετάδοσης και παρατηρείτε ότι ο Β στέλνει στον Α το κρυπτοκείμενο c=1867. Να ανακτήσετε το αρχικό μήνυμα m.

(***Σημείωση: Για την εύρεση Modular αντιστρόφου αριθμού, μπορείτε να χρησιμοποιήσετε κάποιο έτοιμο εργαλείο. Σε κάθε περίπτωση, για επιβεβαίωση των υπολογισμών σας μπορείτε να αξιοποιήσετε το εργαλείο Cryptool από την επιλογή Encrypt ->Asymmetric ->RSADemonstration: θα πρέπει όμως να καταδείξετε τις πράξεις (δηλαδή τους τύπους και το συλλογισμό σας) και όχι να παραθέσετε απλά τα τελικά αποτελέσματα που θα σας δώσει το πρόγραμμα).***

7.3 ΑΣΚΗΣΗ 3

Δείξτε ότι αν κάποιος γνωρίζει το φ(N) και το N, τότε μπορεί να υπολογίσει την παραγοντοποίηση του N (δηλαδή τα p,q τέτοια ώστε N=pq). Προς επικύρωση της διαδικασίας που θα περιγράψετε, υπολογίστε την παραγοντοποίηση του N=171950281, γνωρίζοντας ότι φ(N)=171923092.

(Υπόδειξη: Δείξτε αρχικά ότι p+q=N-φ(N)+1. Μην παραγοντοποιήσετε το N με αυτόματο εργαλείο λογισμικού).

ΚΕΦΑΛΑΙΟ 8 : ΨΗΦΙΑΚΕΣ ΥΠΟΓΡΑΦΕΣ ΚΑΙ ΨΗΦΙΑΚΑ ΠΙΣΤΟΠΟΙΗΤΙΚΑ

8.1 ΑΣΚΗΣΗ 1

Ας ανακαλέσουμε την Άσκηση 7 και συγκεκριμένα τους χρήστες Α και Β του Ερωτήματος Β) αυτής, οι οποίοι επικοινωνούν με τον αλγόριθμο RSA. Όπως είδαμε στο ερώτημα αυτό, ο Β έστειλε στον Α το μήνυμα m=12, το οποίο κρυπτογραφήθηκε σε c=1867.

Α) Αν ο Β ήθελε ταυτόχρονα να υπογράψει αυτό το μήνυμα m, με απλή RSA
ψηφιακή υπογραφή, ποια θα ήταν η υπογραφή που θα δημιουργούσε;
(Θεωρείστε, για λόγους απλότητας, ότι H(m)=m).
Β) Περιγράψτε τις ενέργειες που θα κάνει ο Α προκειμένου να ελέγξει τη
γνησιότητα της ψηφιακής αυτής υπογραφής.
Γ) Αν ο Β επιθυμούσε το μήνυμά του m=12 να υπογραφεί όχι από τον ίδιο
αλλά από τον Α, εν είδει RSA «τυφλής υπογραφής» (blindsignature), ποιες
ενέργειες θα έκανε ο Β και ποιες ο Α; Καταγράψτε τα βήματα αλλά και τις
σχετικές πράξεις που απαιτούνται.

8.2 ΑΣΚΗΣΗ 2

Εγκαταστήστε το πρόγραμμα GPG και δημιουργήστε ζευγάρι δημόσιου/ιδιωτικού κλειδιού. Χρησιμοποιώντας το δημόσιο κλειδί του διδάσκοντα (αρχείο .asc) κρυπτογραφήστε ένα αρχείο με ό,τι περιεχόμενο επιθυμείτε, έτσι ώστε να μπορεί να το διαβάσει μόνο ο διδάσκοντας και κανείς άλλος. Υπογράψτε ταυτόχρονα το μήνυμά σας, έτσι ώστε να είναι ο διδάσκων σε θέση να επαληθεύσει την υπογραφή σας (θα πρέπει να υποβάλετε και το δημόσιό σας κλειδί). Για το πρόγραμμα GPG μπορείτε να ανατρέξετε στη σελίδα: http://www.gpg4win.org/about.html

8.3 ΑΣΚΗΣΗ 3

Έστω ότι ένας χρήστης V θέλει να εξακριβώσει την ταυτότητα του χρήστη C σε ένα σύστημα Fiat-Shamir. Το δημόσιο κλειδί του C αποτελείται από τους αριθμούς N=55, u=16. Ας υποθέσουμε ότι ο C στέλνει στον V τον αριθμό x=4 (witness). Κατόπιν ο V στέλνει στον C το e=1 (challenge) και η απάντηση που λαμβάνει ο V είναι ο αριθμός y=5. Θα εκληφθεί ως έγκυρη (σωστή) αυτή η απάντηση από τον Verifier; (μην προσπαθήσετε να «ανακαλύψετε» το ιδιωτικό κλειδί του C, θεωρείστε ότι είναι υπολογιστικά ανέφικτο).

ΚΕΦΑΛΑΙΟ 13 : ΑΣΦΑΛΙΖΟΝΤΑΣ ΤΟ ΔΙΑΔΙΚΤΥΟ: ΤΟ ΠΡΩΤΟΚΟΛΛΟ ΑΣΦΑΛΕΙΑΣ SSL/TLS. Η ΠΛΑΤΦΟΡΜΑ OPENSSL

Ακολουθούν Συνδυαστικές Ασκήσεις για πλήθος θεμάτων που αφορούν τα ανωτέρω Κεφάλαια.

Υποδείξεις:

1) Σε όλους τους συμμετρικούς αλγορίθμους, τόσο για την κρυπτογράφηση όσο και για την αποκρυπτογράφηση, να καλείτε και τις παραμέτρους –nopad –nosalt (επειδή το openssl ως προκαθορισμένη ρύθμιση εισάγει «padding" και "salting" σε κάθε κρυπτογράφηση, ανεξαρτήτως μεγέθους της εισόδου. Με τις ως άνω παραμέτρους, δεν θα παρατηρείτε «περίεργα» block στην έξοδο).

2) Για τη δημιουργία αρχείου κειμένου ακριβώς 16 χαρακτήρων (π.χ. για όνομα ClaudeShannon), μπορείτε να εργαστείτε ως εξής: echo –n "claudeshannoncla" >name.txt

3) Για να επεξεργαστείτε αρχεία με περιεχόμενο σε 16δική μορφή, μπορείτε να χρησιμοποιήσετε έναν οποιονδήποτε hexeditor. Ένας καλός editor που

μπορείτε να εγκαταστήσετε δωρεάν στο Ubuntu είναι ο Bless, αλλά έχετε την επιλογή να χρησιμοποιήσετε όποιον επιθυμείτε).

13.1 ΑΣΚΗΣΗ 1 : Συμμετρικοί και ασύμμετροι αλγόριθμοι κρυπτογράφησης – Openssl

Α. Δημιουργήστε ένα αρχείο με το όνομα name.txt το οποίο να περιέχει ακριβώς 16 χαρακτήρες, οι οποίοι να είναι οι πρώτοι 16 χαρακτήρες του ονοματεπωνύμου σας (αν τυχόν το ονοματεπώνυμό σας αποτελείται από λιγότερους από 16 χαρακτήρες, προσθέστε χαρακτήρες της επιλογής σας). Το μέγεθος του αρχείου θα πρέπει να είναι ακριβώς 16 byte. Κρυπτογραφείστε αυτό το αρχείο με τον αλγόριθμο AES (128 bit μεγέθους κλειδιού, ενώ το διάνυσμα IV – όπου χρειάζεται – να είναι το μηδενικό), σε τρόπους λειτουργίας ECB, CBC και OFB αντίστοιχα, με κλειδί της επιλογής σας – δημιουργώντας με αυτόν τον τρόπο 3 κρυπτοκείμενα encrypted_ecb.bin, encrypted_cbc.bin και encrypted_ofb.bin. Θα πρέπει να περιγράψετε τις εντολές που θα χρησιμοποιήσετε, καθώς επίσης και να υποβάλετε και τα ως άνω κρυπτοκείμενα που θα έχετε δημιουργήσει.

Β. Δημιουργήστε ένα αρχείο με το όνομα repeated_name.txt, το οποίο να περιέχει το περιεχόμενο του name.txt αλλά επαναλαμβανόμενο 5 φορές, έτσι ώστε να έχει μέγεθος 80 byte. Να επαναλάβετε το ερώτημα i για το ίδιο κλειδί κρυπτογράφησης και IV που χρησιμοποιήσατε ανωτέρω, δημιουργώντας 3 νέα κρυπτοκείμενα new_encrypted_ecb.bin, new_encrypted_cbc.bin και new_encrypted_ofb.bin. Θα πρέπει να περιγράψετε τις εντολές που θα χρησιμοποιήσετε. Παρατηρείστε τα κρυπτοκείμενα τόσο του ερωτήματος ii όσο και του ερωτήματος i και

σχολιάστε τα, βάσει των όσων είναι γνωστά από τη θεωρία του μαθήματος για τους τρόπους λειτουργίας των κρυπταλγορίθμων τμήματος.

Γ. Στα τρία κρυπτοκείμενα του Ερωτήματος 2, τροποποιήστε το πρώτο bit αυτών και αποκρυπτογραφήστε τα, με το ίδιο κλειδί και το ίδιο IV, δημιουργώντας τα αρχεία new_decrypted_ecb.bin, new_decrypted_cbc.bin και new_decrypted_ofb.bin. Σχολιάστε τα αποτελέσματα.

13.2 ΑΣΚΗΣΗ 2

Το αρχείο ciphertext1.bin έχει δημιουργηθεί με κρυπτογράφηση με AES σε τρόπο λειτουργίας CTR, με άγνωστο για εσάς κλειδί (δεν γνωρίζετε ούτε έστω το μέγεθος του κλειδιού). Γνωρίζετε όμως ότι πρόκειται για κρυπτογράφηση του μηνύματος «message.txt» (μεγέθους 32 χαρακτήρων) το οποίο στέλνει η Alice στον Bob με στοιχεία τραπεζικού λογαριασμού της Alice, για να της μεταφέρει ο Bob χρήματα – δηλαδή, ο λογαριασμός της Alice είναι 1586120871445081. Με άλλα λόγια, το αρχείο ciphertext1.bin αποτελεί κρυπτογράφηση του αρχείου message.txt.

Α)ΗEve θέλει να παραβιάσει την ασφάλεια της επικοινωνίας, τροποποιώντας το

κρυπτοκείμενο κατά τέτοιο τρόπο ώστε ο Bob, όταν το αποκρυπτογραφήσει με το μυστικό κλειδί, να διαβάσει, αντί για το λογαριασμό 1586120871445081, τον λογαριασμό 1586120871445198 (ο οποίος είναι ο λογαριασμός της Eve). Μπορεί η Eve να τροποποιήσει

κατάλληλα το κρυπτογραφημένο αυτό μήνυμα έτσι ώστε να επιτύχει το σκοπό της; Να εξηγήσετε αναλυτικά το συλλογισμό σας.

Β) Τι πρέπει να κάνει η Alice προκειμένου να είναι σε θέση ο Bob να «καταλαβαίνει» αν το μήνυμα που λαμβάνει είναι γνήσιο και δεν αλλοιώθηκε κατά τη μετάδοση - έτσι ώστε να καθίσταται τελικά μη αποτελεσματική η τεχνική της Eve που θα (πρέπει να) περιγράψετε στο ερώτημα Α).

13.3 ΑΣΚΗΣΗ 3

Να δημιουργήσετε ένα ζεύγος δημόσιου-ιδιωτικού κλειδιού για τον RSA, με μέγεθος 3072 bits για το N. Να υποβάλετε το δημόσιο κλειδί σας (όχι το ιδιωτικό), περιγράφοντας και τις εντολές με τις οποίες το δημιουργήσατε.

13.4 ΑΣΚΗΣΗ 4

Α) Να δημιουργήσετε ένα τυχαίο κλειδί συμμετρικής κρυπτογράφησης μεγέθους 128 bit και να το αποθηκεύσετε σε ένα αρχείο key.bin. Με το κλειδί αυτό να κρυπτογραφήσετε με AES, σε τρόπο λειτουργίας CTR, με μηδενικό IV, το αρχείο

«name.txt» της Άσκησης 13.1 Α). Να υπολογίσετε επίσης το αποτύπωμα του αρχείου κρυπτογραφημένα αρχεία – χωρίς το αρχείο key.bin.

αυτού με τη συνάρτηση SHA-256, το οποίο επίσης ακολούθως να κρυπτογραφήσετε με

τον ίδιο αλγόριθμο και το ίδιο κλειδί. Να περιγράψετε τις εντολές που θα

χρησιμοποιήσετε («αποκρύπτοντας» το κλειδί) και να υποβάλετε τα δύο αυτά

Β) Για να αποκρυπτογραφήσει ο διδάσκων και μόνο αυτός τα αρχεία του ερωτήματος[ii] (δηλαδή το μήνυμα και το αποτύπωμα αυτού), χρειάζεται να έχει το συμμετρικό κλειδί που δημιουργήσατε (το αρχείο key.bin). Πώς θα το στείλετε το κλειδί αυτό στον διδάσκοντα του Τμήματός σας με ασφαλή τρόπο; Περιγράψτε τη διαδικασία και υποβάλετε τα σχετικά αρχεία (ένα ή περισσότερα, ό,τι κρίνετε ότι πρέπει) που θα δημιουργήσετε. Προσοχή: θα πρέπει να υποβάλατε μόνο τα απολύτως απαραίτητα αρχεία έτσι ώστε μόνο ο διδάσκων να μπορεί να αποκρυπτογραφήσει το συμμετρικό κλειδί. Σας δίνεται το δημόσιο κλειδί publickey-tutor.pem του διδάσκοντα

13.5 ΑΣΚΗΣΗ 5

Μελετήστε τη συνάρτηση crypt του Ubuntu (με την εντολή mancrypt) και περιγράψτε πώς χρησιμοποιείται για τη δημιουργία ψηφιακών αποτυπωμάτων. Εξηγείστε ειδικότερα το ρόλο του salt, πώς αυτό χρησιμοποιείται στην crypt και τι υποδηλώνουν οι πρώτοι 3 (από τους συνολικά 12) χαρακτήρες του πεδίο του salt.

13.6 ΑΣΚΗΣΗ 6

(Υπόδειξη: Η συνάρτηση crypt θα πρέπει να κληθεί μέσα από ένα πρόγραμμα C (ή C++). Στον πηγαίο κώδικα του προγράμματος θα πρέπει να αναγράψετε στην αρχή την εντολή

#include<crypt.h>

Ενώ για τη μεταγλώττισή του θα πρέπει να χρησιμοποιήσετε και την παράμετρο –lcrypt:
gcccrack_passwd.c –lcrypt –ocrack_passwd όπου crack_passwd.c (ως παράδειγμα) το όνομα του αρχείου με τον πηγαίο κώδικα και crack_passwd το όνομα του εκτελέσιμου αρχείου που θα δημιουργήσετε. Εφόσον δημιουργηθεί σωστά το εκτελέσιμο αρχείο, θα το καλέσετε ως εξής: ./crack_passwd. Πιθανώς να σας ζητηθεί, την πρώτη φορά, να εγκαταστήσετε το μεταγλωττιστή gcc.)

Α) Έχετε καταφέρει να αποκτήσει πρόσβαση στο αρχείο shadow ενός UbuntuLinux συστήματος (βλ. αρχείο shadow στο e-class). Το σύστημα αυτό έχει δύο χρήστες, με ονόματα bill και helen. Σκοπός σας είναι να ανακαλύψετε τα συνθηματικά τους. Για να το καταφέρετε αυτό, έχετε κάποιες πληροφορίες για την προσωπική ζωή των bill και helen:

Για το χρήστη bill, έχετε κάποιες πληροφορίες περί της προσωπικής του ζωής, όπως ότι μένει στη Λευκωσία, η σύζυγός του ονομάζεται Μαρία, έχει γεννηθεί στις 25/8/1977 και του αρέσουν οι PinkFloyd.

Για τη helen, έχει γεννηθεί στις 3/8/1984, το τηλέφωνό της είναι 6955345671 και η πινακίδα του αυτοκινήτου της είναι ZKA5231.

Και οι δύο είναι νέοι χρήστες των υπολογιστών και πιθανότατα επιλέγουν «εύκολα» στο να απομνημονευτούν συνθηματικά, οπότε η λίστα με τα πιο συχνά συνθηματικά για το 2017 συγκεντρώνει καλές πιθανότητες να περιέχουν τα συνθηματικά που επέλεξαν. Περιγράψτε αναλυτικά τις ενέργειες που θα κάνετε για να μαντέψετε τα συνθηματικά τους, κάνοντας τις δοκιμές σας πάνω στο αρχείο shadow. Ποια συνάρτηση κατακερματισμού χρησιμοποιήθηκε για την παραγωγή των αποτυπωμάτων των συνθηματικών;

Β) Ποια από τα συνθηματικά που βρήκατε στο ερώτημα Α) θα μπορούσαν να βρεθούν και με επίθεση τύπου rainbowattack, εφόσον βέβαια δεν είχε χρησιμοποιηθεί το salt? Να καταδείξετε την απάντησή σας με βάση το διαδικτυακό εργαλείο https://crackstation.net/. (θεωρείστε ότι και στην περίπτωση που δεν χρησιμοποιούνταν salt, χρησιμοποιείται η ίδια συνάρτηση κατακερματισμού με αυτή που χρησιμοποιήθηκε από το λειτουργικό σύστημα Ubuntu).

13.7 ΑΣΚΗΣΗ 7 (Συνδυασμός συμμετρικής και ασύμμετρης κρυπτογράφησης)

Η Alice στέλνει σε κάποιον από τους χρήστες της λίστας users το κρυπτογραφημένο αρχείο encrypted_message.hex, το οποίο έχει κρυπτογραφηθεί με τον AES, σε CBC τρόπο λειτουργίας με κλειδί k μεγέθους 128 bit (16 bytes). Συγκεκριμένα, το στέλνει στο χρήστη με ψευδώνυμο «B7 70 0B 37»: τα ψευδώνυμα των χρηστών προκύπτουν με εφαρμογή του SHA-512 στην ηλεκτρονική διεύθυνση (email) του χρήστη, λαμβάνοντας τους τελευταίους 4 χαρακτήρες του αποτυπώματος. Για την ασφαλή ανταλλαγή του μυστικού αυτού κλειδιού k, η Alice το κρυπτογράφησε με τον αλγόριθμο RSA, χρησιμοποιώντας το δημόσιο RSA κλειδί του παραλήπτη, όπως περιέχεται στο αρχείο users. Το κρυπτογραφημένο αυτό κλειδί k είναι το αρχείο encrypted_key.txt. HRSA κρυπτογράφηση γίνεται με κρυπτογράφηση κάθε χαρακτήρα (byte) ξεχωριστά, δηλαδή το κάθε byte του κλειδιού αντιστοιχεί σε έναν αριθμό βάσει της κατά ASCII κωδικοποίησής του (δηλαδή το byte 01000001 – που είναι ο χαρακτήρας 'A' - αντιστοιχεί στον αριθμό 65 κ.ο.κ., βάσει της ASCII κωδικοποίησης

που είναι διαθέσιμη στο http://www.ascii-code.com/). Η κρυπτογράφηση έγινε με εφαρμογή της σχέσης $c=m^e \bmod N$ χαρακτήρα-χαρακτήρα (όπου mo αριθμός της Ascii κωδικοποίησης του κάθε byte, όπως περιγράφηκε ανωτέρω) – για αυτό λοιπόν και προκύπτουν 16 αριθμοί στο κρυπτοκείμενο, χωρισμένοι μεταξύ τους με το σύμβολο #.

Α) Να βρείτε σε ποιον χρήστη απευθύνεται το μήνυμα της Alice: αν πιστεύετε ότι δεν είναι εφικτό να βρεθεί ο χρήστης, να εξηγήσετε την απάντησή σας.

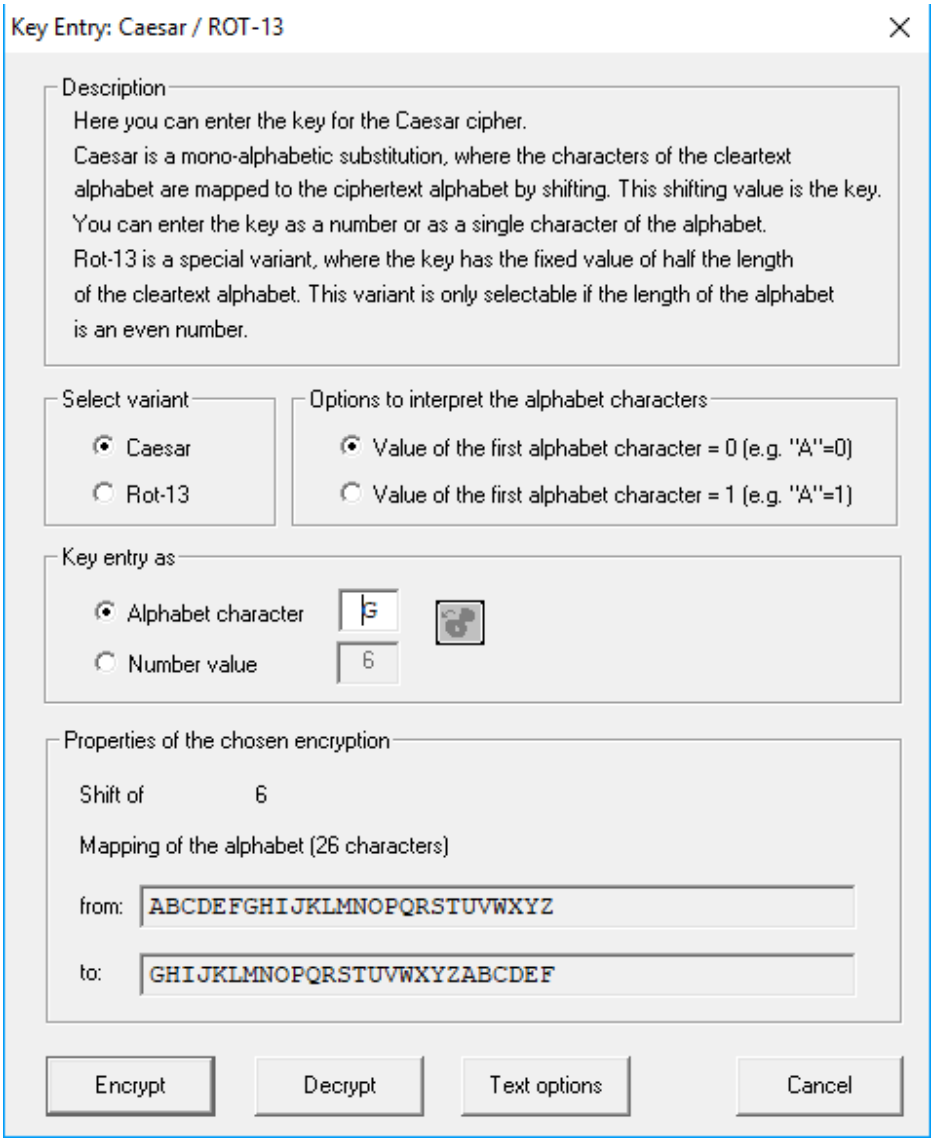
ΜΕΡΟΣ ΔΕΥΤΕΡΟ : ΑΠΑΝΤΗΣΕΙΣ ΑΣΚΗΣΕΩΝ

2.1 ΑΣΚΗΣΗ 1 :

Ερώτημα 1

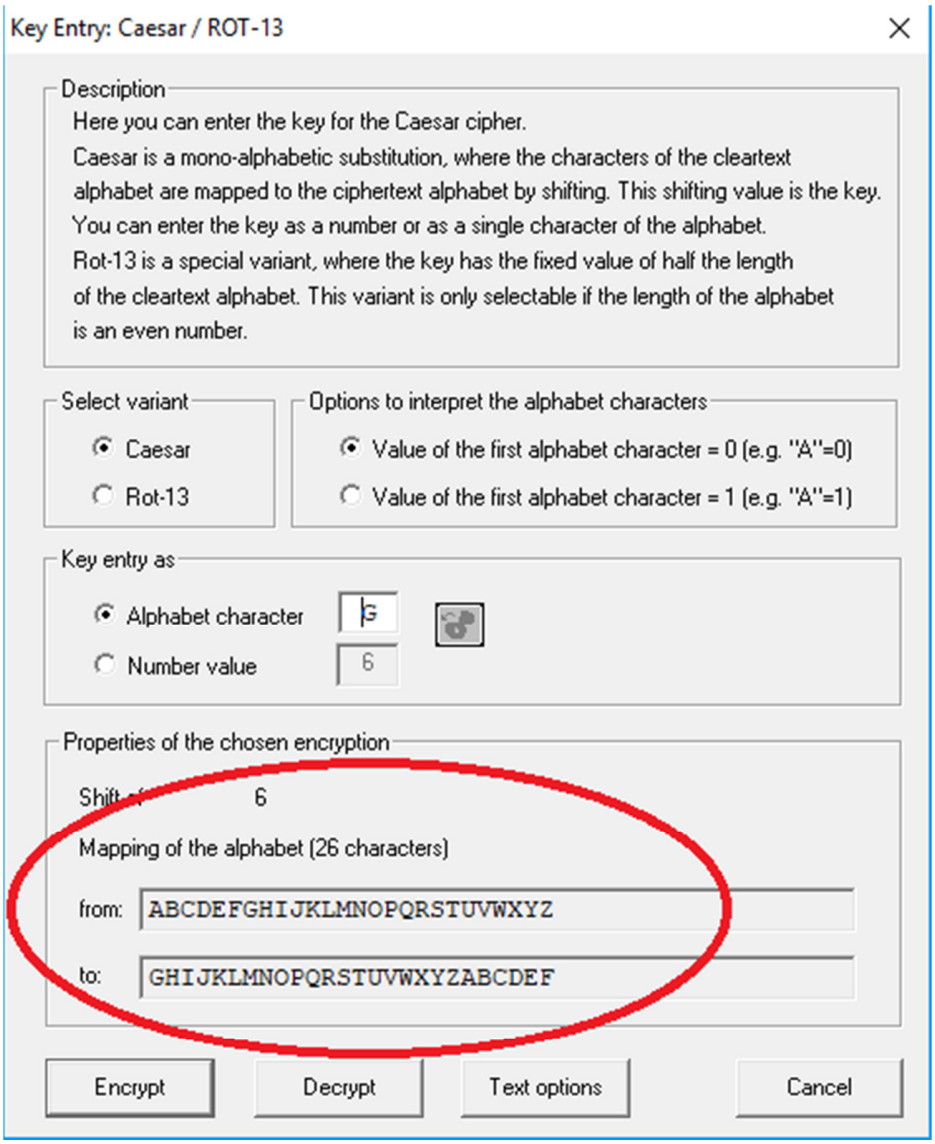
Εκτελέστε το πρόγραμμα CrypTool και ανοίξτε τοαρχείο CrypTool-en.txt στο directoryCrypTool\examples μέσω του μενού File \ Open.

Επιλέγουμε από το μενού **Encrypt/Decrypt \ Symmetric (classic) \ Caesar/Rot-13.**

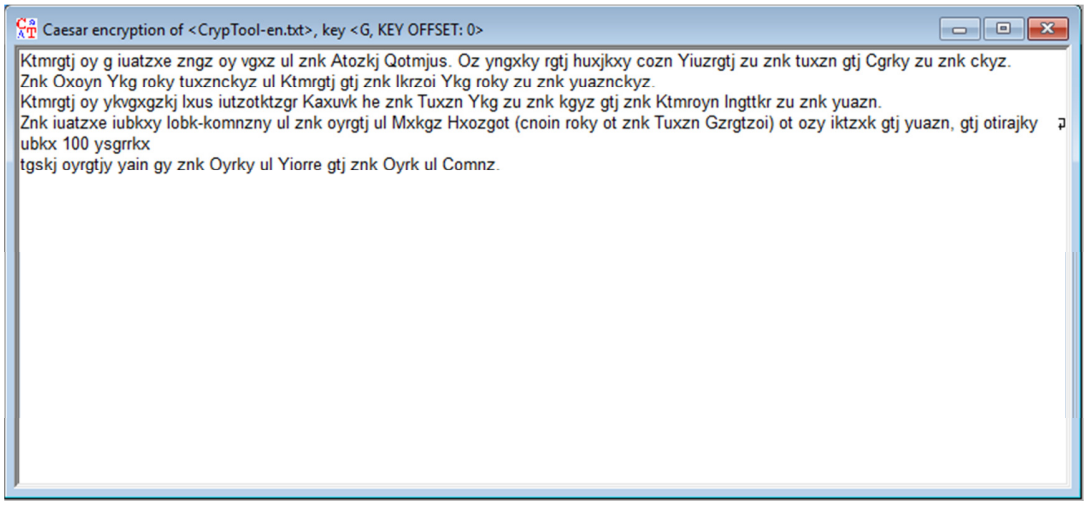


Επιλέξτε να ξεκινάει το αλφάβητο από το 0 (ο πρώτος χαρακτήρας δηλ. το Α να αντιστοιχεί στο 0) και βάλτε σαν κλειδί το πρώτο γράμμα του ονόματός σας.

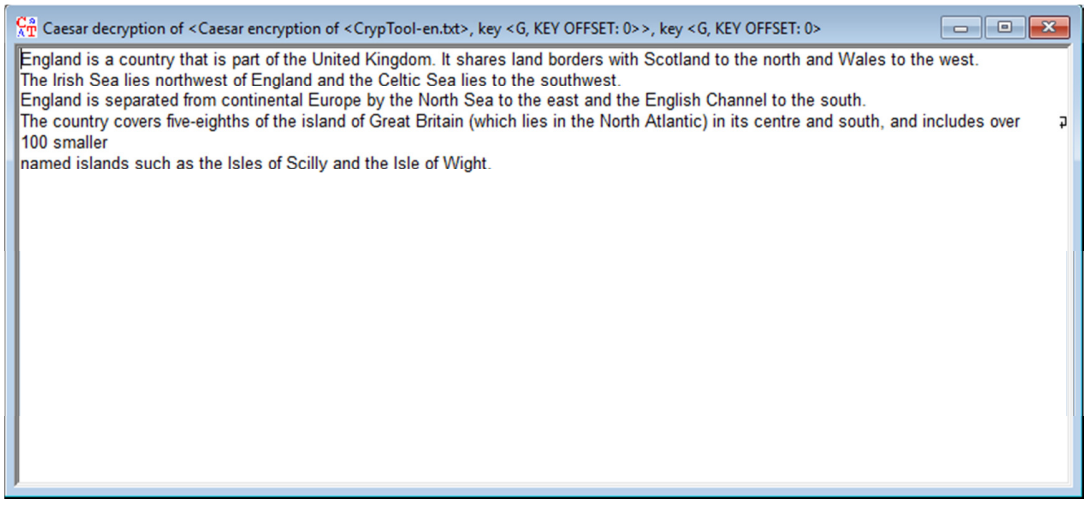
Α. Παρατηρήστε την αντιστοίχηση (Mapping) που παρουσιάζει το εργαλείο. Ποιά είναι αυτή;



Β. Πατώντας το κουμπί encrypt, παρατηρήστε το κρυπτογραφημένο κείμενο που προκύπτει.

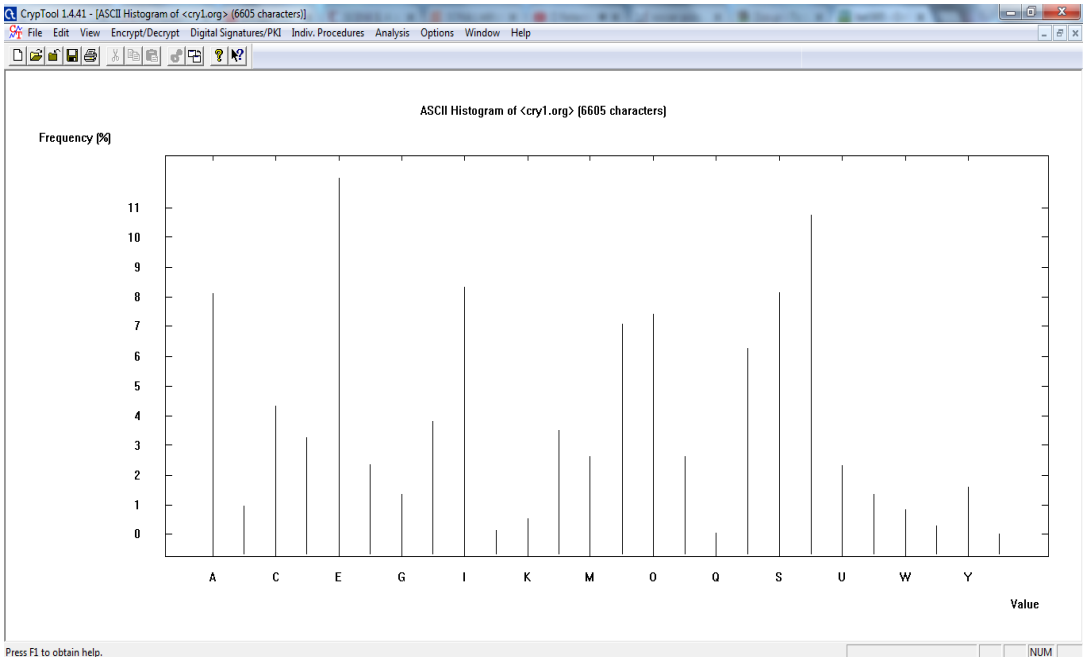


Γ. Αποκρυπτογραφήστε το κρυπτογραφημένο κείμενο χρησιμοποιώντας το ίδιο κλειδί.

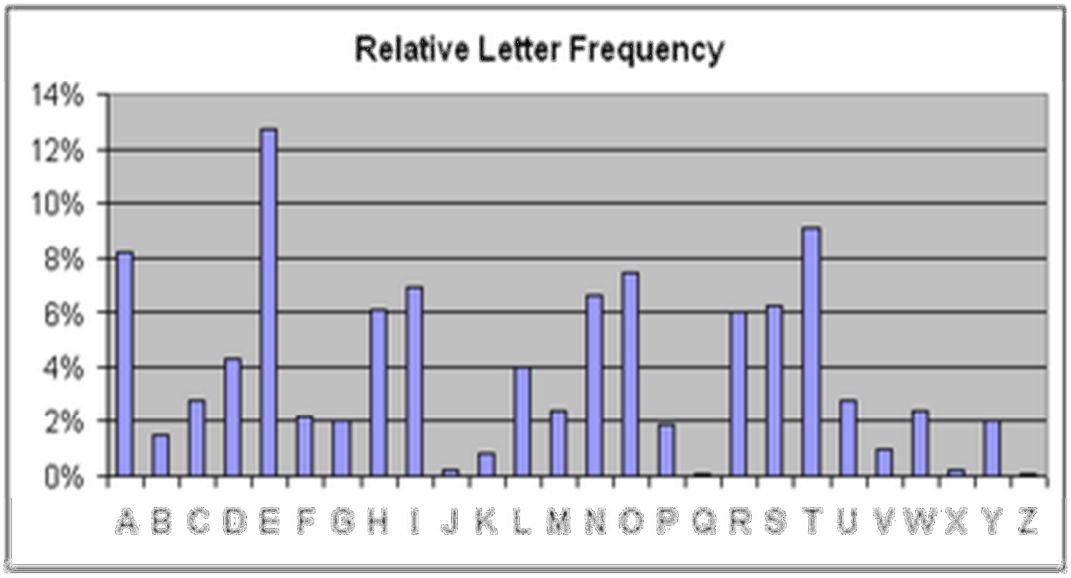


2.2 ΑΣΚΗΣΗ 2

Χρησιμοποιώντας το λογισμικό εργαλείο CrypTool βρήκαμε το ιστόγραμμα συχνοτήτων εμφάνισης των γραμμάτων που εμφανίζονται στο κείμενο ,το οπόιο και παραθέτουμε :



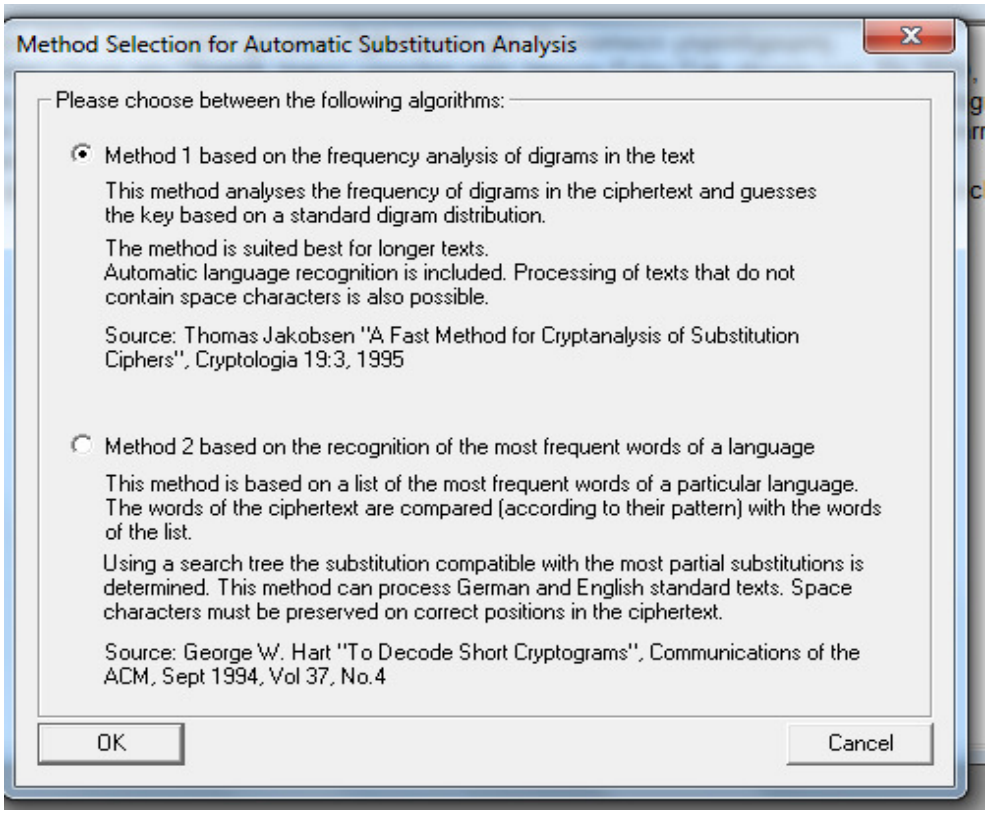
Παραθέτουμε και το ιστόγραμμα τυπικής εμφάνισης των γραμμάτων του αγγλικού αλφαβήτου :



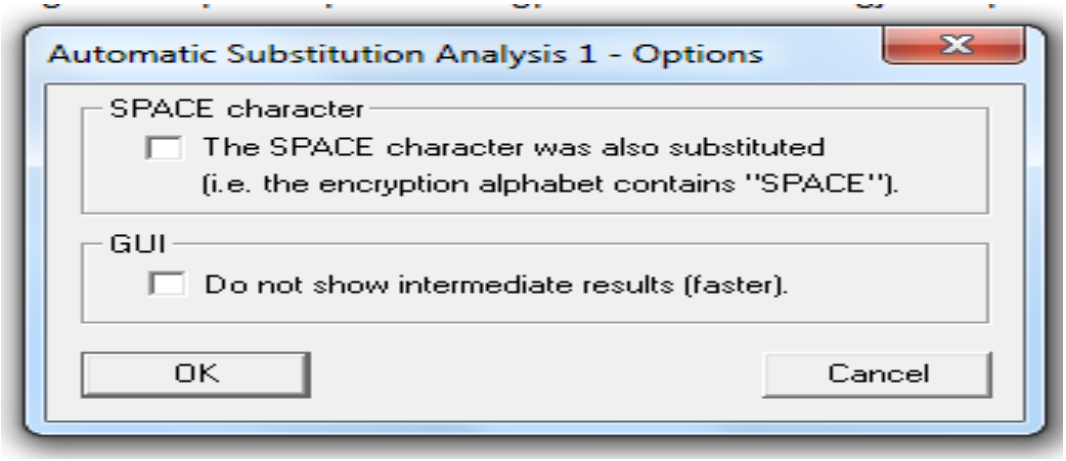
Παρατηρώντας τα δύο ιστογράμματα συχνοτήτων επισημαίνουμε την πολύ μεγάλη ομοιότητα που εμφανίζουν

2.3 ΑΣΚΗΣΗ 3

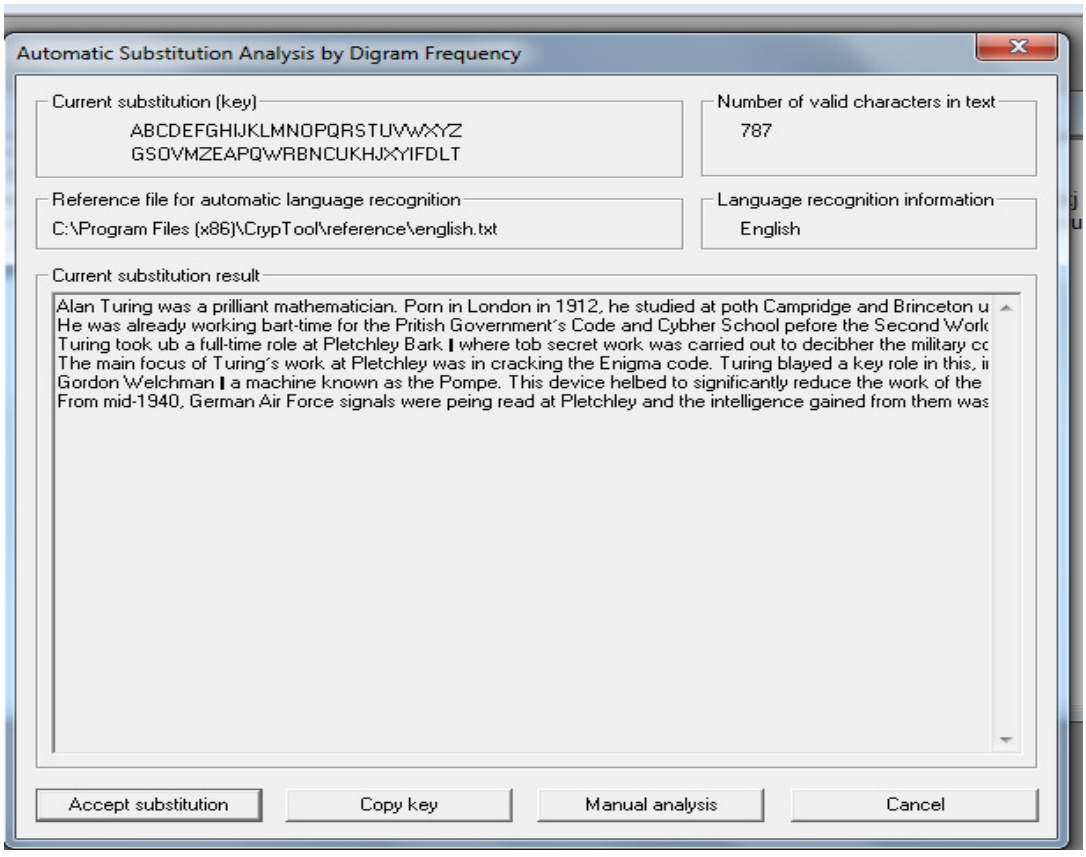
Α. Με την επιλογή "Analysis -> Symmetric Encryption (classic) -> Ciphertext only -> Substitution" εμφανίζεται το εξής παράθυρο επιλογών :



Επιλέγοντας την πρώτη μέθοδο (Method 1) , εμφανίζεται το παράθυρο επιλογής :



Επιλέγουμε την πρώτη option : SPACEcharacter , πατάμε OK και το αποτέλεσμα είναι :

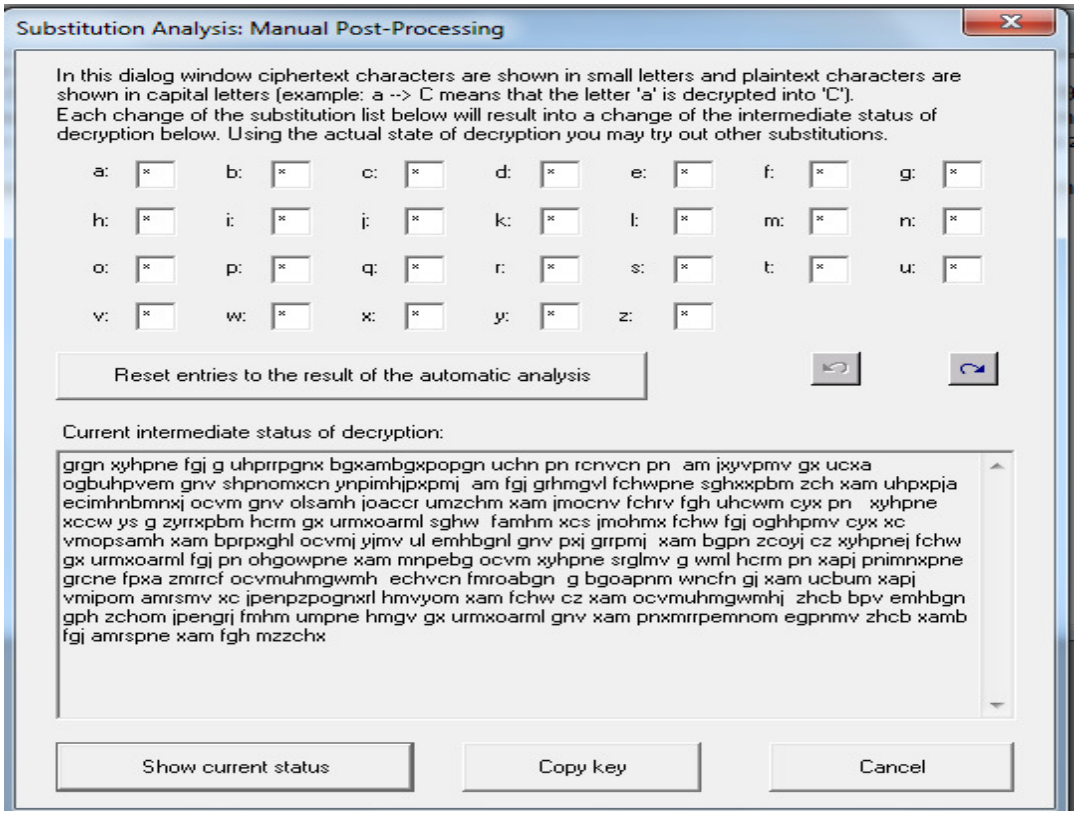


“Alan Turing was a prilliant mathematician. Porn in London in 1912, he studied at poth Campridge and Brinceton universities. He was already working bart-time for the Pritish Government’s Code and Cybher School pefore the Second World War proke out. In 1939, Turing took ub a full-time role at Pletchley Bark – where tob secret work was carried out to decibher the military codes used py Germany and its allies. The main focus of Turing’s work at Pletchley was in cracking the Enigma code. Turing blayed a key role in this, inventing – along with fellow code-preaker Gordon Welchman – a machine known as the Pompe. This device helbed to significantly reduce the work of the code-preakers. From mid-1940, German Air Force signals were peing read at Pletchley and the intelligence gained from them was helbing the war effort.”

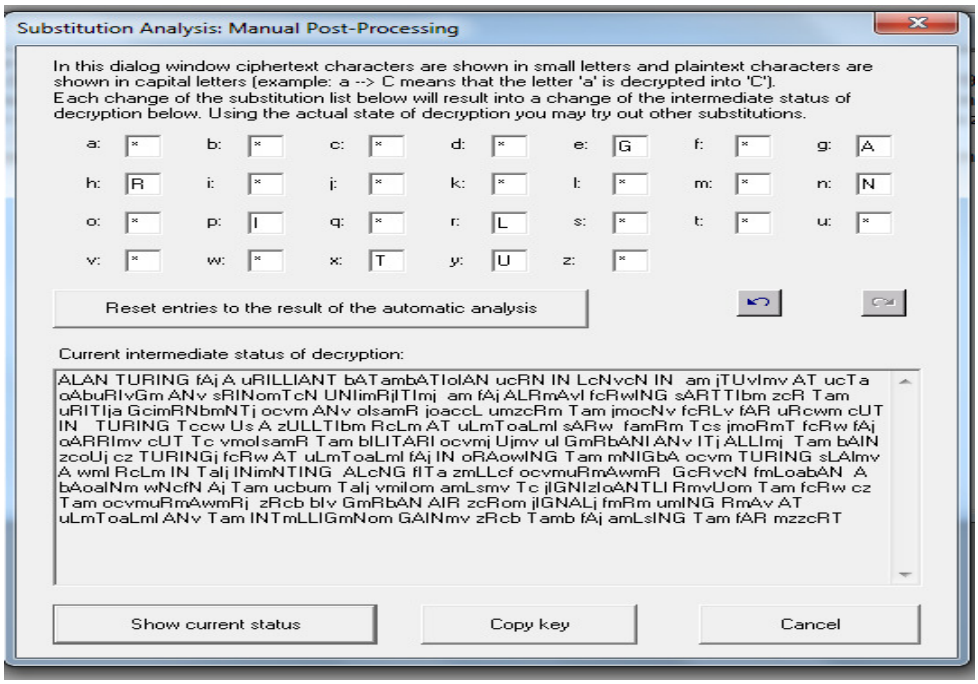
Παρατηρούμε ότι πρόγραμμα αποκρυπτογράφησε όλο το κείμενο με μόνο σφάλμα την αντικατάσταση του γράμματος Β με το γράμμα Ρ.

Β)

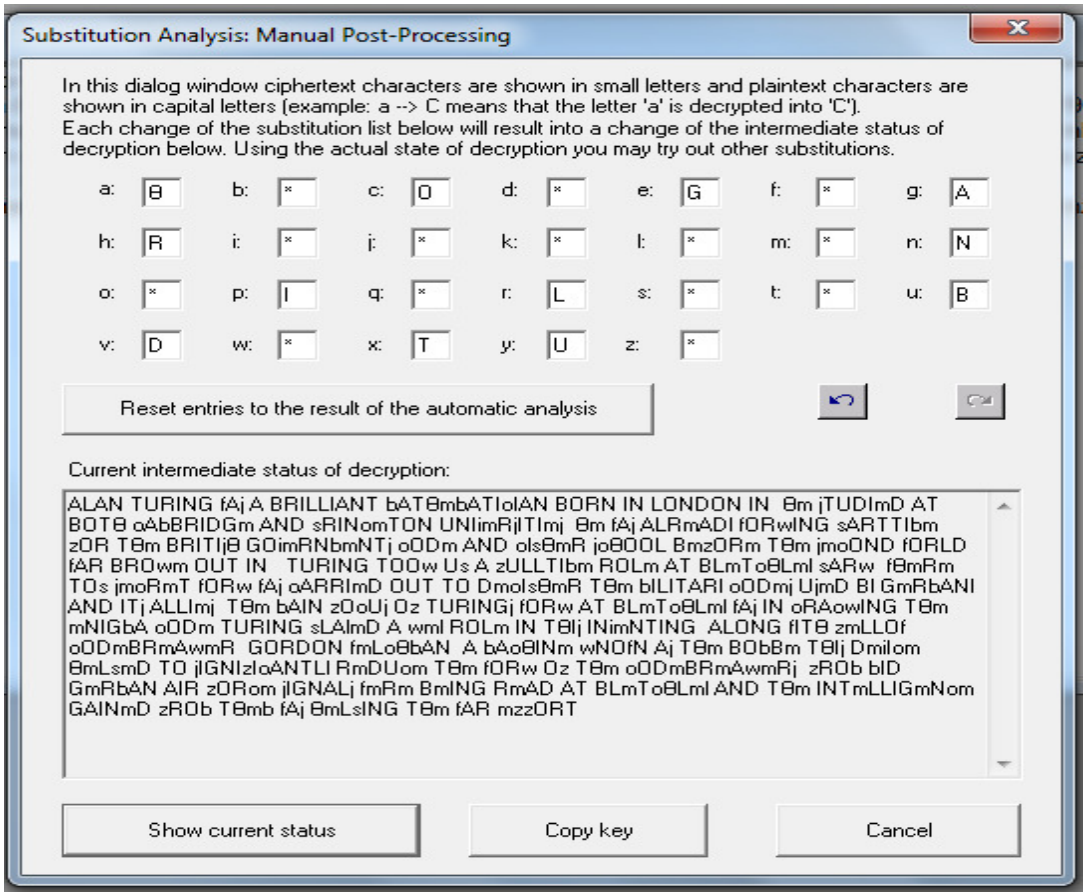
Χρησιμοποιώντας την επιλογή **“Analysis ->SymmetricEncryption (classic) ->Manualanalysis ->Substitution”** εμφανίζεται το παράθυρο :



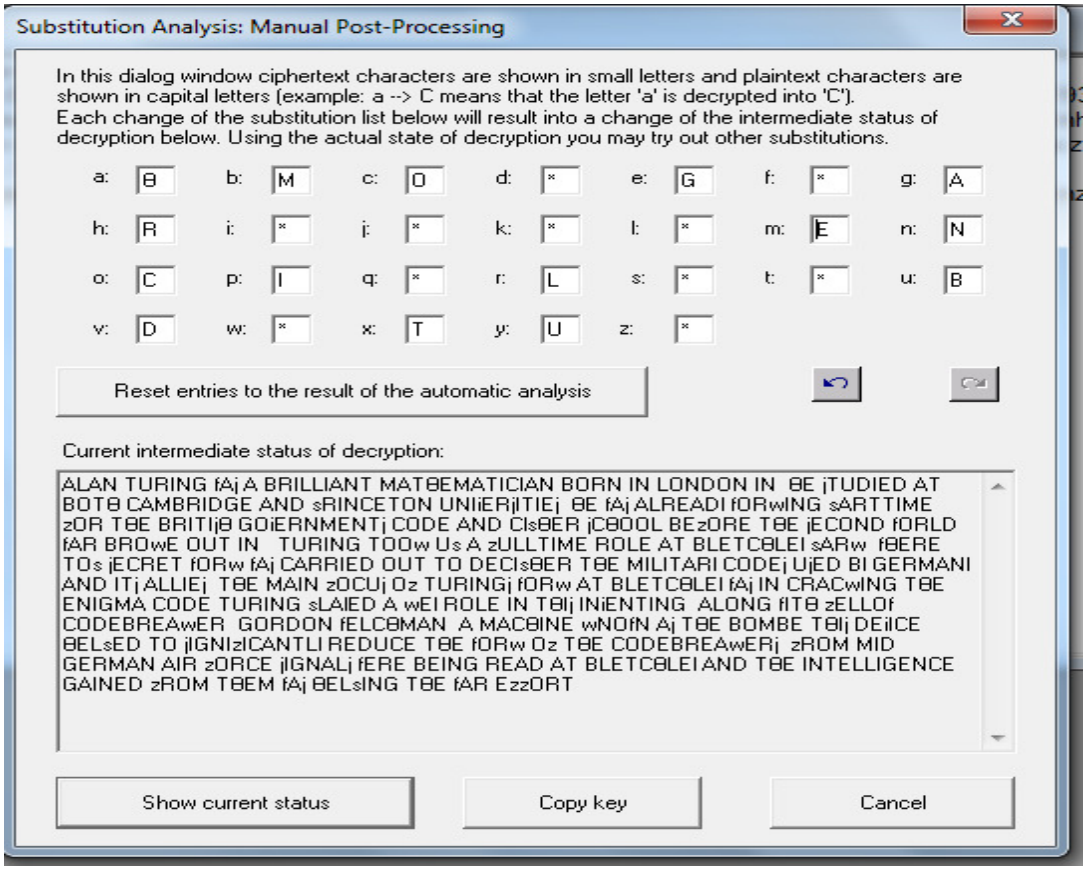
Παρατηρώντας ότι το κρυπτογράφημα : GrgnXyhpne αναφέρεται στο όνομα του AllanTuring εκτελούμε τις αντικαταστάσεις έχουμε :



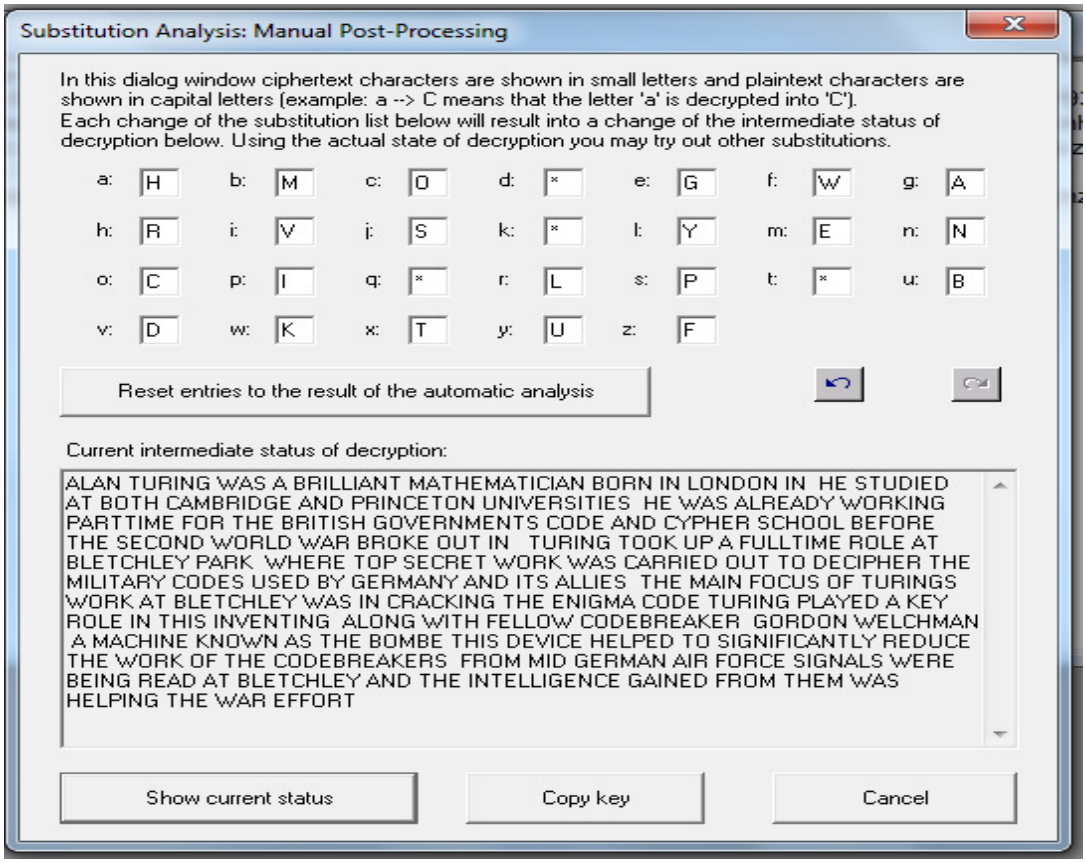
Επίσης παρατηρούμε ότι η λέξη : uRILLIANT που εμφανίζεται σημαίνει προφανώς ότι θα πρέπει να γίνει η αντικατάσταση u➔B , επίσης BcRN σημαινει BORN άρα c➔O , LONvON σημαίνει LONDON άρα v➔D , οπότε το κείμενο γίνεται :



Η λέξη oAbRIDGm πιθανότατα σημαίνει CAMBRIDGE άρα o➔C ,b➔M και m➔E με αυτές τις αντικαταστάσεις το κείμενο γίνεται αρκετά ευανάγνωστο :



Απομένουν οι προφανείς αντικαταστάσεις z➔F , w➔K , i➔V κ.ο.κ και προκύπτει το αποκρυπτογραφημένο κείμενο :



2.4 ΑΣΚΗΣΗ 4

Καταρχάς παρατηρούμε στο κρυπτοκείμενο ότι η ακολουθία χαρακτήρων **fv** επαναλαμβάνεται με απόσταση 4. Συμπεραίνουμε λοιπόν ότι τι κλειδί πρέπει να έχει μέγεθος 4 (ή 2). Έστω L=4. Γνωρίζωντας ότι το μήνυμα αρχίζει με τη φράση dearmary , συμπεραίνουμε ότι :

Tod κρυπτογραφήθηκε ως w➔m_1=3 και c_1=22

Toe κρυπτογραφήθηκε ως v➔m_2=4 και c_2=21

Toa κρυπτογραφήθηκε ως a➔m_3=0 και c_3=0

Tor κρυπτογραφήθηκε ως g➔m_4= 17 και c_4=6

Η σχέση που δίνει την αποκρυπτογράφηση στον αλγόριθμο Vigenereείναι :

$m_j = c_j - k_{jmodL}$ (mod26)

Την εφαρμόζουμε για κάθε χαρακτήρα και έχουμε :

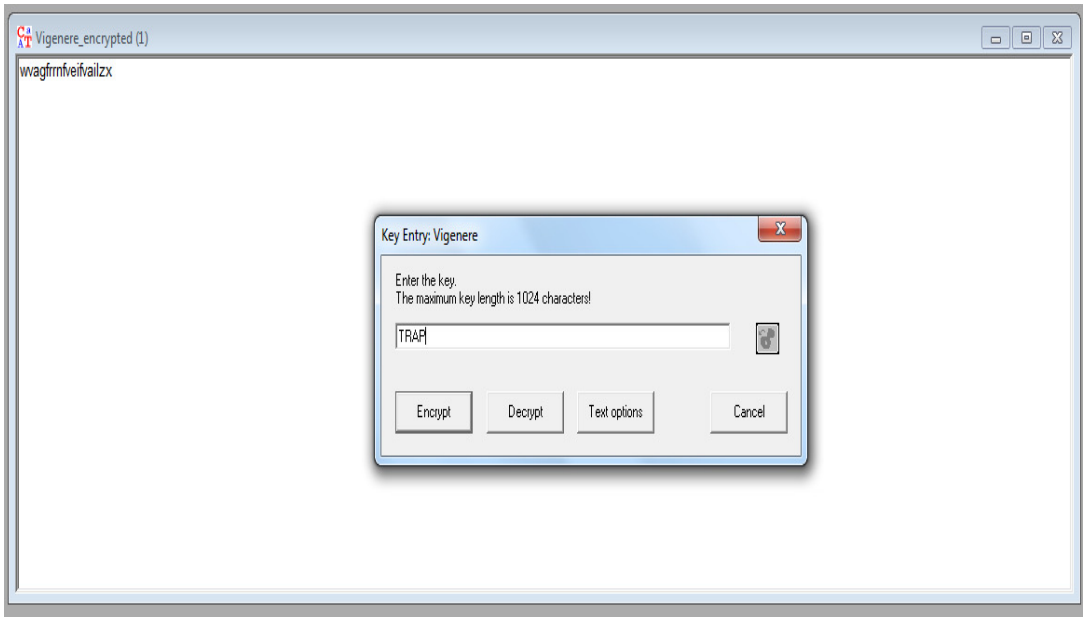
$m_1 = c_1 - k_1$ (mod26) ➔ $3=22 - k_1$ (mod26) ➔ $-19 = - k_1$ (mod26) ➔ $k_1= 19$ ➔**k_1=T**

$m_2 = c_2 - k_2$ (mod26) ➔ $4=21 - k_2$ (mod26) ➔$-17 = - k_2$ (mod26) ➔ $k_2= 17$ ➔**k_2=R**

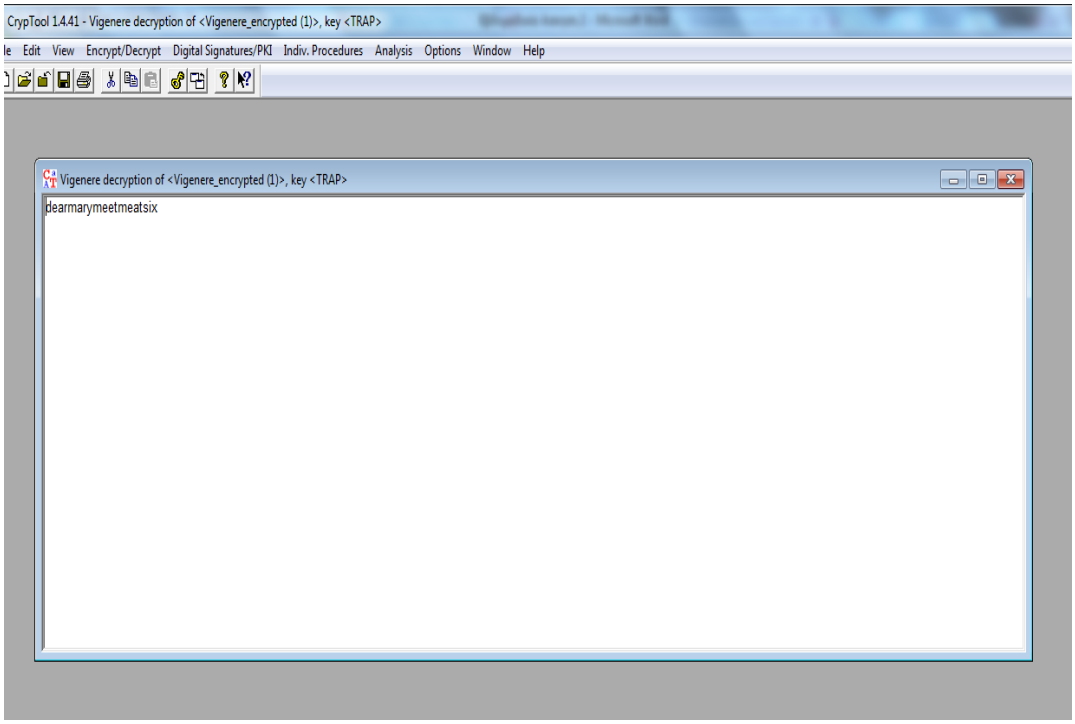
$m_3 = c_3 - k_3$ (mod26) ➔ $0=0 - k_3$ (mod26) ➔k2= 0 ➔**k_3=A**

$m_4 = c_4 - k_4$ (mod26) ➔ $17=6 - k_4$ (mod26)➔$11 = - k_4$ (mod26) ➔ $-11=k_4$(mod26) ➔$15=k_4$(mod26) ➔ $k_4=15$ ➔**k_4= P**

Άρα η λέξη κλειδί είναι : TRAP . Χρησιμοποιούμε τώρα το CrypTool με την ακολουθία εντολών : “Encrypt/Decrypt ->Symmetric (classic) ->Vigenere” και στο παράθυρο που εμφανίζεται τοποθετουμε το κλειδί :



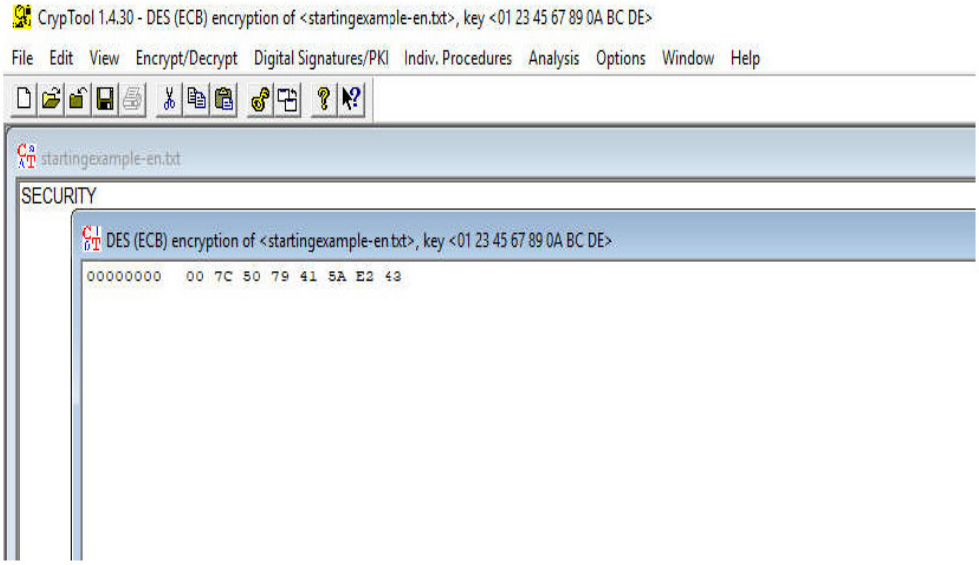
Επιλέγοντας Decrypt παίρνουμε το αρχικό μήνυμα :



“DEAR MARY MEET ME AT SIX”

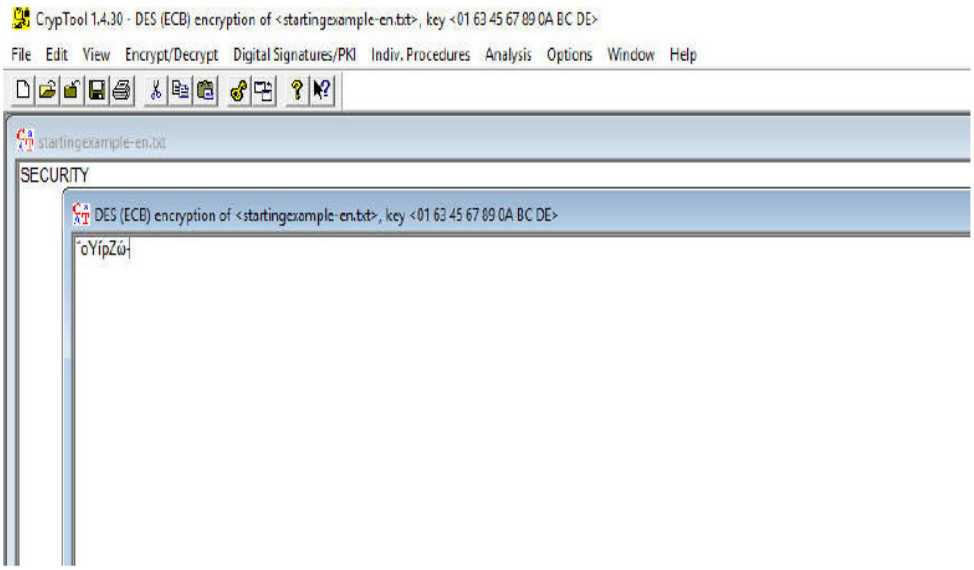
3.1 ΑΣΚΗΣΗ 1

Κρυπτογραφούμε το μήνυμα SECURITY (8 χαρακτήρες) με κλειδί 01234567890ABCDE και λαμβάνουμε το κρυπτοκείμενο (σε 16δική μορφή) 00 7C 50 79 41 5AE2 43.



i) Αλλάζουμε ένα bit του κλειδιού – π.χ. το 10ο bit, οπότε το κλειδί μετατρέπεται σε

01634567890ABCDE (το 2 προηγουμένως, δηλ. το 0010, που αντιστοιχούσε στα bit 9-12, μετατράπηκε σε 6, δηλαδή 0110). Κρυπτογραφούμε το μήνυμα με το νέο αυτό κλειδί και λαμβάνουμε το κρυπτοκείμενο ΄οΥίρΖώ- (οι χαρακτήρες είναι εκτυπώσιμοι)



Τα δύο κρυπτοκείμενα είναι όντως διαφορετικά – αλλά για να δούμε ακριβώς πόσο διαφέρουν, τα προσθέτουμε με XOR για να δούμε σε πόσα bit διαφέρουν. Η πρόσθεση με XOR μπορεί να γίνει με κρυπτογράφηση one-time-pad, οπότε πραγματοποιώντας την λαμβάνουμε το αποτέλεσμα A1 13 09 A6 31 9C 1C 6E ή 10100001 00010011 00001001 10100110 00110001 10011100 00011100 01101110 (δηλαδή τα δύο κρυπτοκείμενα διαφέρουν σε 27 από τις συνολικά 64 θέσεις, ήτοι σε ποσοστό περίπου 42%.)

ii) Αλλάζουμε ένα bit του μηνύματος – π.χ. το 8ο bit, οπότε το μήνυμα μετατρέπεται σε RECURITY (η αλλαγή φαίνεται με το κίτρινο – το byte του χαρακτήρα S προηγουμένως, δηλ. το 01010011, που αντιστοιχούσε στα bit 1-8, μετατράπηκε στο byte του χαρακτήρα R, δηλαδή 01010010). Κρυπτογραφούμε με το αρχικό μας κλειδί και λαμβάνουμε το κρυπτοκείμενο 25 B7 7F 11 DABE 67 7F

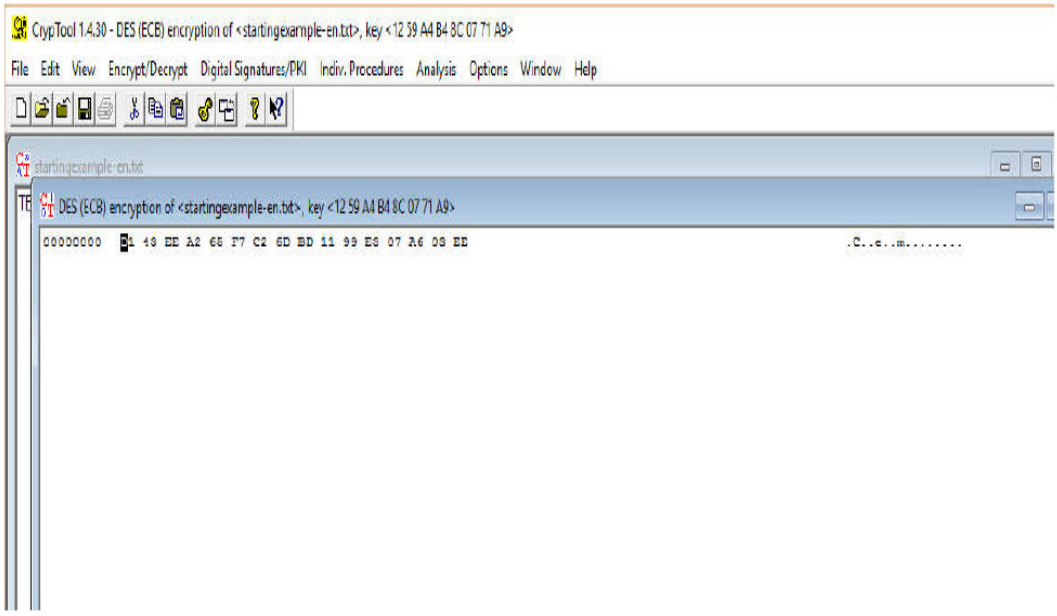
Όπως και νωρίτερα, συγκρίνουμε τα δύο κρυπτοκείμενα προσθέτοντάς τα με XOR και λαμβάνουμε 25 CB 2F 68 9BE4 85 3C, δηλαδή 00100101 11001011 00101111 01101000 10011011 11100100 1000010100111100 (δηλαδή τα δύο κρυπτοκείμενα διαφέρουν σε 32 από τις συνολικά 64 θέσεις, ήτοι σε ποσοστό 50%.)

Άρα, επιβεβαιώνεται το avalancheeffect (αλλαγή σε 1 bit σημαίνει πολλές αλλαγές στο κρυπτοκείμενο, περίπου ίσες με το μισό πλήθος ψηφίων).

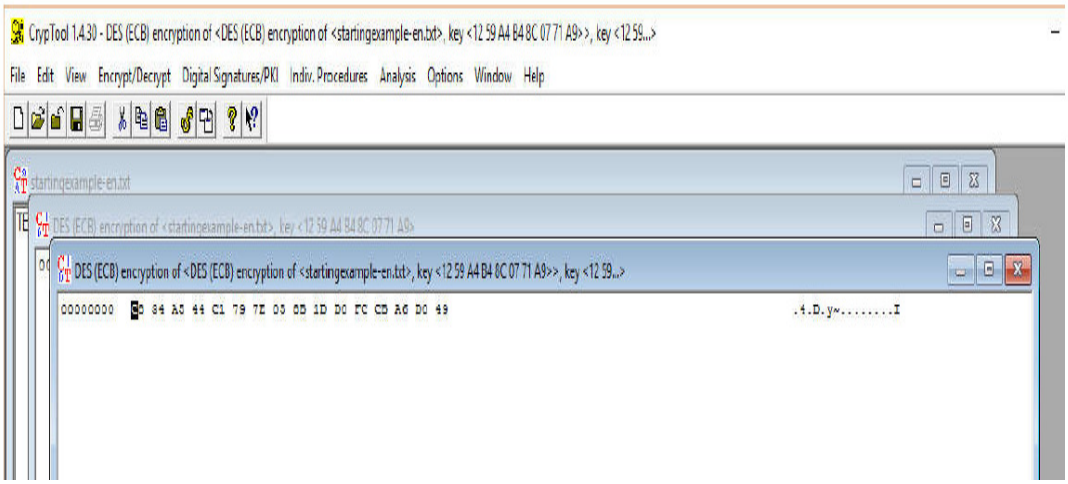
3.2 ΑΣΚΗΣΗ 2

Α) Κρυπτογραφούμε το μήνυμα TESTMESSAGE διαδοχικά δύο φορές (αντίστοιχα για οποιοδήποτε άλλο μήνυμα, όπως το ονοματεπώνυμο που

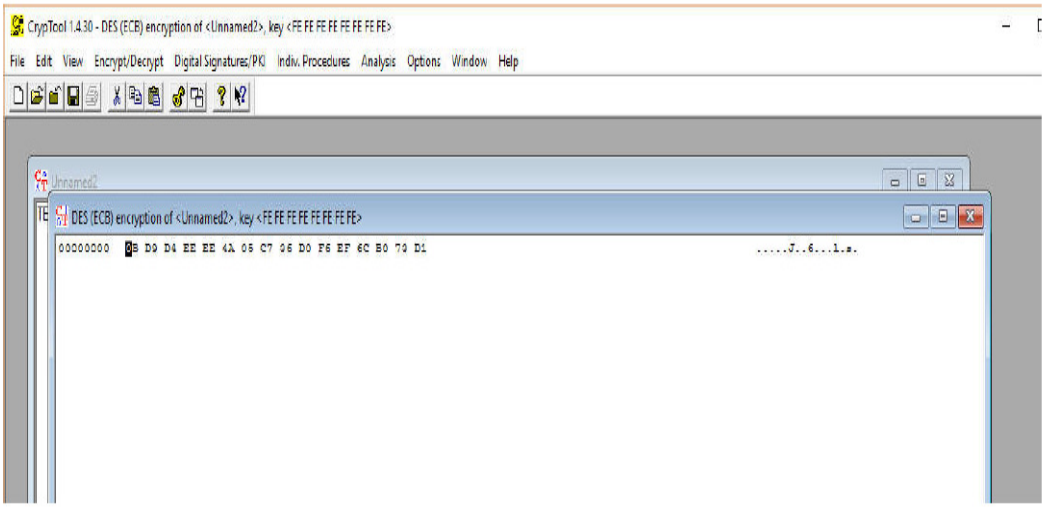
ζητείται), με κλειδί κρυπτογράφησης το 12 59 A4 B4 8C 07 71 A9 (τυχαία επιλογή). Η πρώτη κρυπτογράφηση μας δίνει το εξής κρυπτοκείμενο:



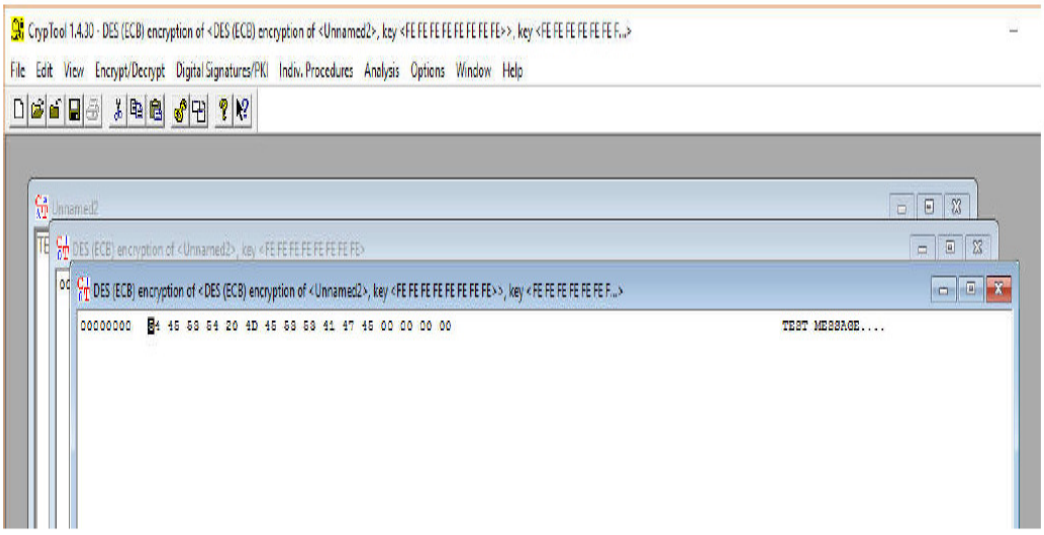
Και κρυπτογραφώντας το εκ νέου, θα έχουμε:



Β) Επιλέγουμε ως κλειδί το FEFEFEFEFEFEFEFE (ένα εκ των αδύναμων κλειδιών). Κρυπτογραφώντας το μήνυμά μας μία φορά με αυτό το κλειδί, θα έχουμε:



Κρυπτογραφώντας το εκ νέου, με το ίδιο κλειδί, θα έχουμε:



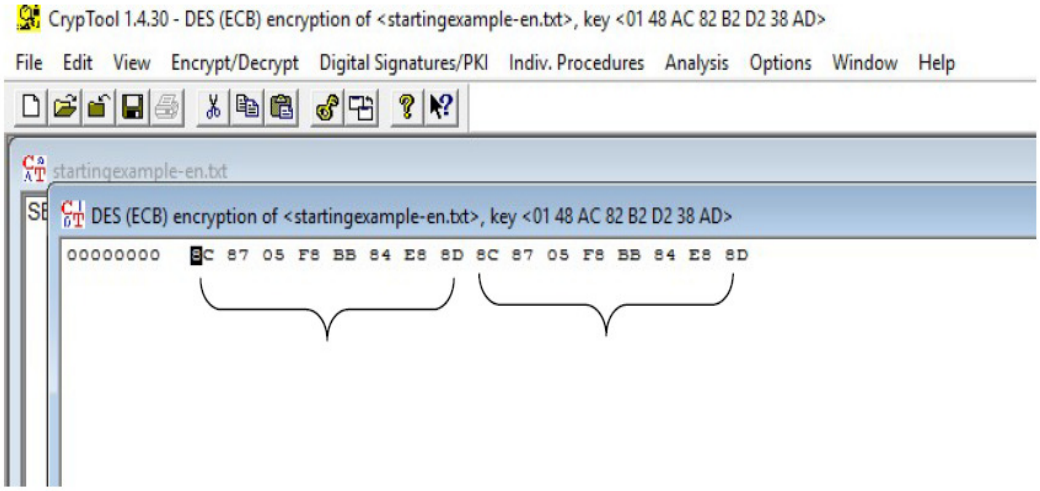
Δηλαδή ανακτούμε το ίδιο μήνυμα. Αυτό οφείλεται στο ότι κάθε weakkey παράγει ακριβώς τα ίδια υπο-κλειδιά και για τους 16 γύρους, οπότε η κρυπτογράφηση ταυτίζεται με την αποκρυπτογράφηση (λόγω της Feistel δομής του αλγορίθμου).

3.3 ΑΣΚΗΣΗ 3

Μπορούμε να χρησιμοποιήσουμε το μήνυμα SECURITYSECURITY, το οποίο αποτελείται από 8 χαρακτήρες (δηλ. 64 bits, όσο είναι το μέγεθος του block για τον DES και τον 3DES) που επαναλαμβάνονται. Αναμένεται το κρυπτοκείμενο στον ECB τρόπο λειτουργίας, είτε του DES είτε του 3DES, να αποτελείται επίσης από 8 χαρακτήρες που επαναλαμβάνονται, ενώ αυτό δεν αναμένεται να παρατηρηθεί στον CBC τρόπο λειτουργίας: αυτά θα ισχύουν για οποιοδήποτε κλειδί κρυπτογράφησης και αν επιλεγεί.

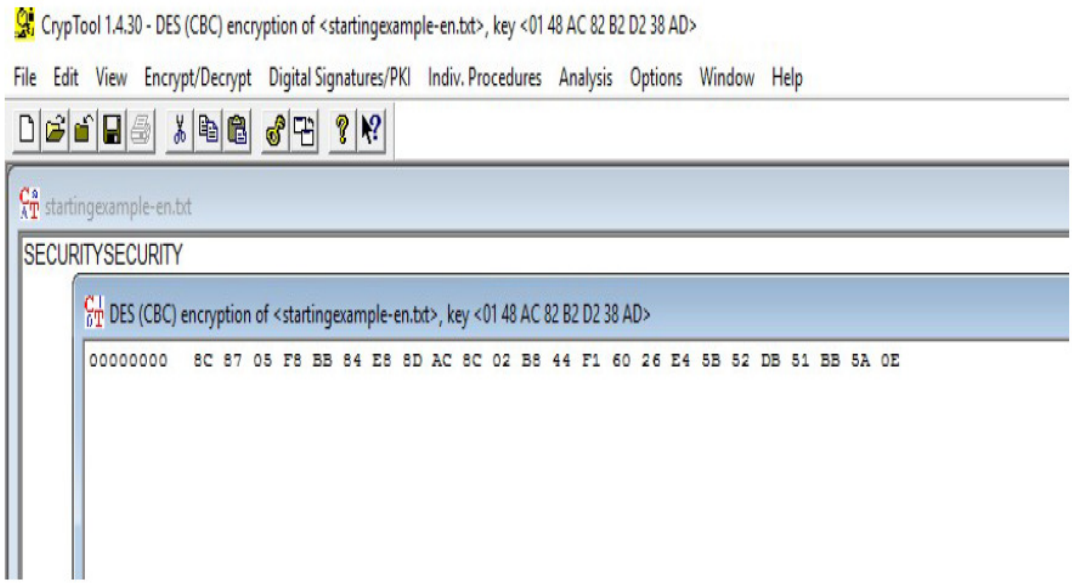
Πράγματι, έχουμε:

Κρυπτογράφηση DES-ECB (κλειδί (τυχαία επιλογή): 01 48 AC 82 B2 D2 38 AD:



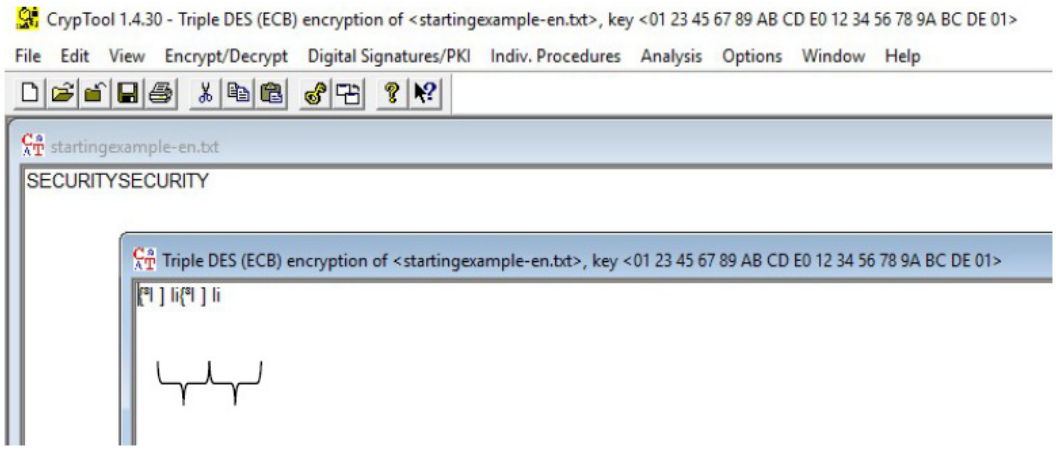
Εικόνα 1

Κρυπτογράφηση DES-CBC (κλειδί (τυχαία επιλογή): 01 48 AC 82 B2 D2 38 AD :



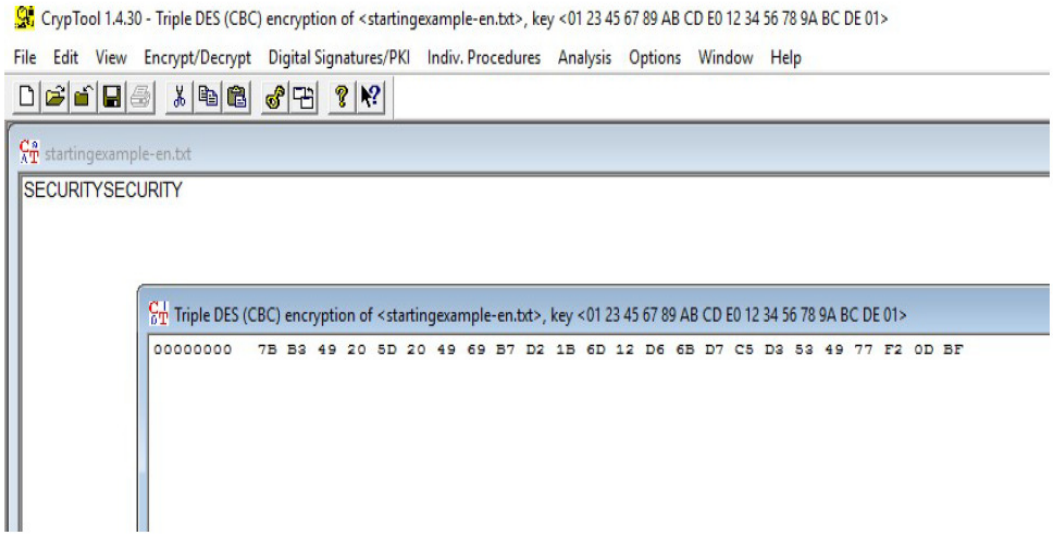
Εικόνα 2

Κρυπτογράφηση 3DES-ECB (κλειδί (τυχαία επιλογή): 01 23 45 67 89 ABCDE0 12 34 56 78 9ABCDE 01 :



Εικόνα 3

Κρυπτογράφηση 3DES-CBC (κλειδί (τυχαία επιλογή): 01 23 45 67 89 ABCD
E0 12 34 56 78 9A BC DE 01):



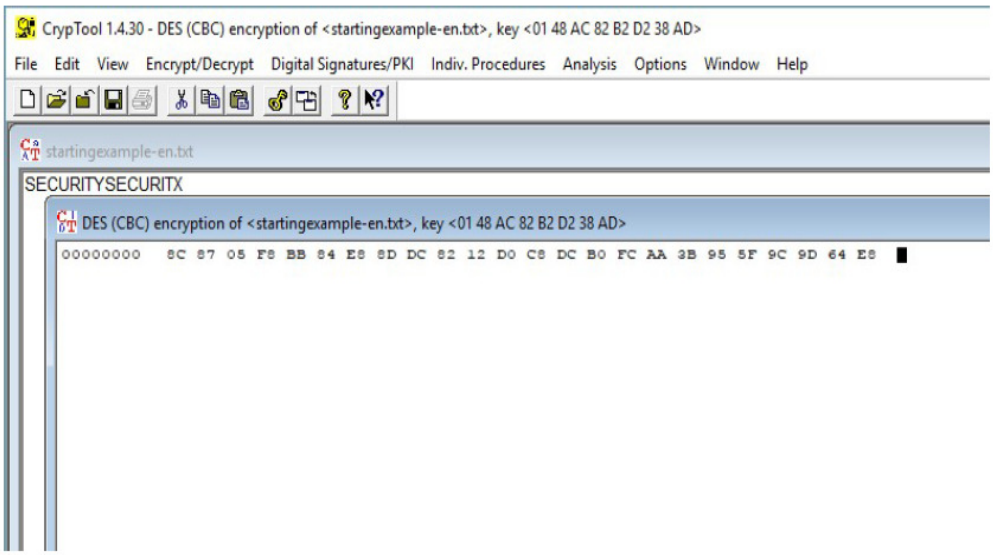
Εικόνα 4

3.4 ΑΣΚΗΣΗ 4

Αξιοποιώντας τα προηγούμενα παραδείγματα της Άσκησης 3, μπορούμε να κρυπτογραφήσουμε το μήνυμα SECURITYSECURITX (διαφέρει σε έναν μόνο χαρακτήρα από το SECURITYSECURITY και, μάλιστα, σε ένα μόνο bit, αφού η Ascii κωδικοποίηση του Y είναι 01011001 και του X είναι 01011000), με τα ίδια κλειδιά που χρησιμοποιήθηκαν ανωτέρω, σε CBC τρόπο λειτουργίας.

Έχουμε λοιπόν για την περίπτωση του DES-CBC (κλειδί: 01 48 AC 82 B2 D2

38 AD):

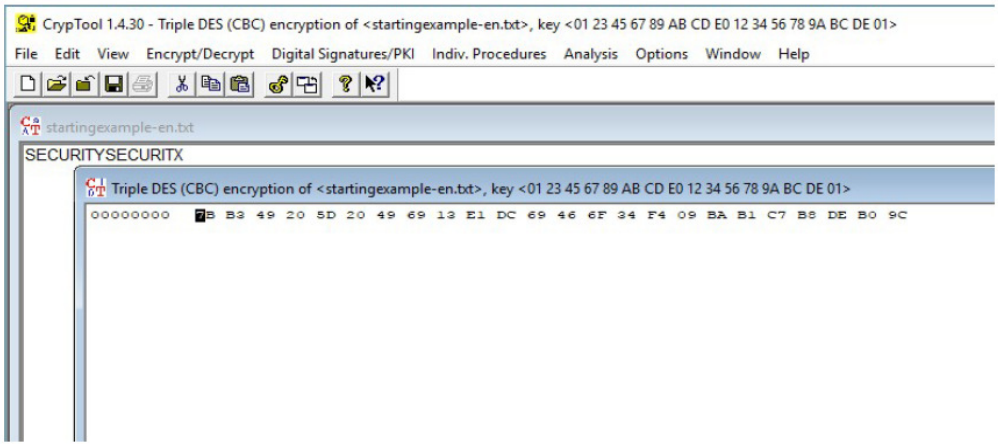


Εικόνα 5

Συγκρίνοντας λοιπόν την αυτην την εικόνα με την δεύτερη εικόνα της προηγούμενης ασκησης , επιβεβαιώνουμε ότι μία πολύ μικρή αλλαγή στο μήνυμα επέφερε μεγάλες αλλαγές στο τελευταίο block του κρυπτοκειμένου (στην Εικόνα 2 το block αυτό είναι το E4 5B 52 DB 51 BB 5A 0E ενώ στην Εικόνα 5 το block αυτό είναι το AA 3B 95 5F 9C 9D 64 E8 – μπορούμε να επιβεβαιώσουμε, με τις δυαδικές αναπαραστάσεις των ανωτέρω, ότι διαφέρουν στα 35 από τα συνολικά 64 bit – δηλαδή περίπου στα μισά).

Αντίστοιχα, κρυπτογραφούμε το SECURITYSECURITX με τον 3DES-CBC, ξανά με

το κλειδί 01 23 45 67 89 AB CD E0 12 34 56 78 9A BC DE 01):



Εικόνα 6

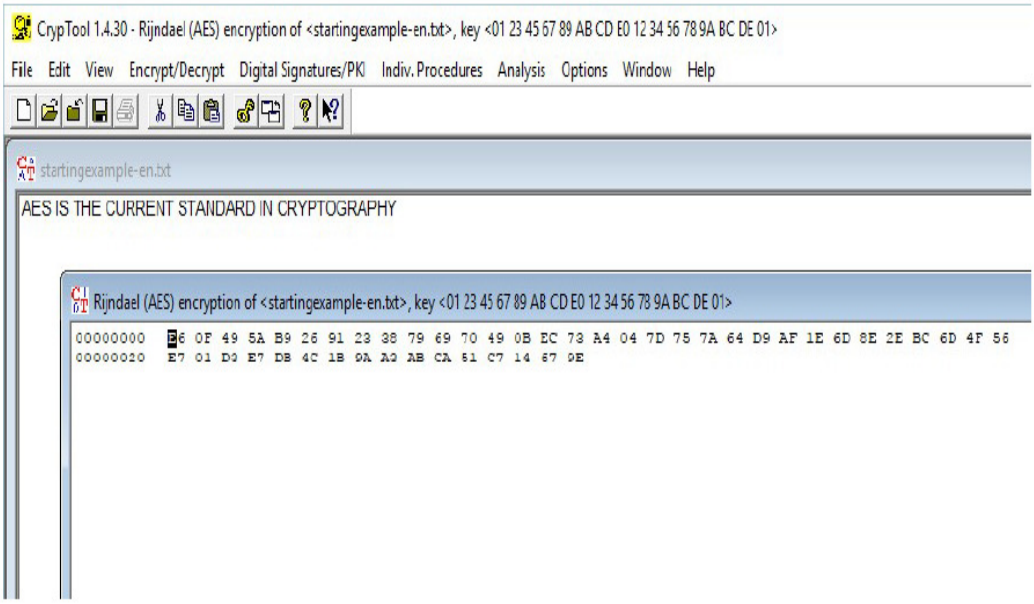
Συγκρίνοντας λοιπόν την Εικόνα 6 με την Εικόνα 4, επιβεβαιώνουμε ότι μία πολύ
μικρή αλλαγή στο μήνυμα επέφερε μεγάλες αλλαγές στο τελευταίο block του
κρυπτοκειμένου (στην Εικόνα 4 το block αυτό είναι το C5 D3 53 49 77 F2 0DBF
ενώ στην Εικόνα 6 το block αυτό είναι το 09 BAB1 C7 B8 DEB0 9C – μπορούμε να
επιβεβαιώσουμε, με τις δυαδικές αναπαραστάσεις των ανωτέρω, ότι διαφέρουν στα
34 από τα συνολικά 64 bit – δηλαδή περίπου στα μισά).

Τέλος, κρυπτογραφούμε αντίστοιχα και με τον AES. Το μήνυμα
SECURITYSECURITY έχει μέγεθος 16 χαρακτήρες, δηλαδή 128 bit – δηλαδή
αντιστοιχεί σε ένα μόνο block μηνύματος. Επιλέγουμε μεγαλύτερο μήνυμα, για να
έχουμε τουλάχιστον δύο block μηνύματος (άρα, θα παραχθεί και ένα block

κρυπτοκειμένου). Επιλέγουμε το μήνυμα

AES IS THE CURRENT STANDARD IN CRYPTOGRAPHY

μεγέθους 43 χαρακτήρων μαζί με τα κενά (οπότε θα γίνει και συμπλήρωση – padding). Χρησιμοποιούμε ως κλειδί κρυπτογράφηση το 01 23 45 67 89 AB CD E0 12 34 56 78 9ABCDE 01 και το κρυπτοκείμενο είναι το εξής:



Εικόνα 7

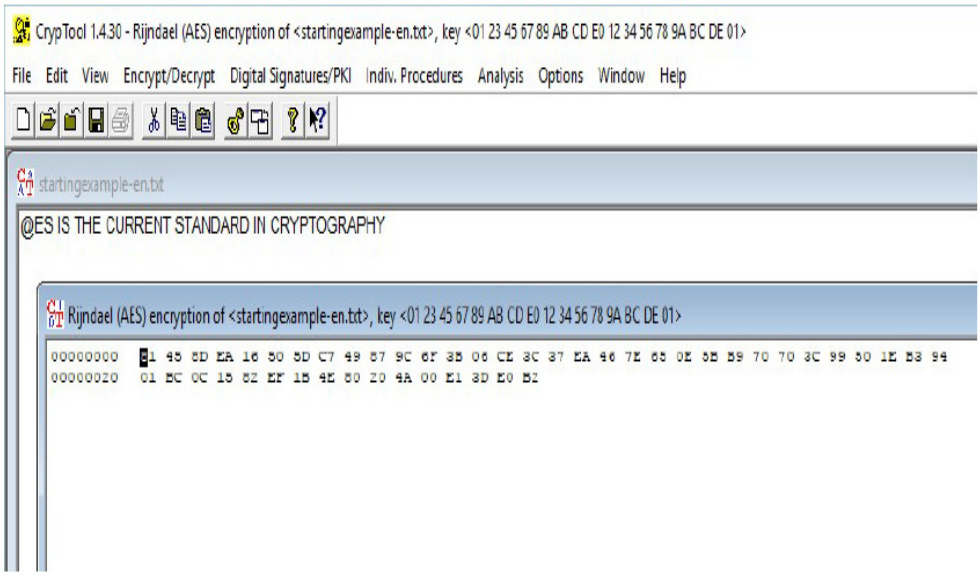
Το τελευταίο block του κρυπτοκειμένου είναι το

E7 01 D3 E7 DB 4C 1B 9A A3 AB CA 51 C7 14 67 9E

Αλλάζουμε το μήνυμά μας κατά 1 bit, σε

@ES IS THE CURRENT STANDARD IN CRYPTOGRAPHY

Και κρυπτογραφούμε το νέο μήνυμα με το ίδιο κλειδί:



Εικόνα 8

Το τελευταίο block του κρυπτοκειμένου είναι το

01 BC 0C 15 82 EF 1B 4E 80 20 4A 00 E1 3D E0 B2

Συγκρίνοντας αυτά τα δύο (των Εικόνων 8 και 7) παρατηρούμε ότι μία πολύ μικρή

αλλαγή στο μήνυμα επέφερε μεγάλες αλλαγές στο τελευταίο block του κρυπτοκειμένου (μπορούμε να επιβεβαιώσουμε, με τις δυαδικές αναπαραστάσεις των

ανωτέρω, ότι διαφέρουν στα 58 από τα συνολικά 128 bit – δηλαδή περίπου στα μισά).

5.1 ΑΣΚΗΣΗ 1

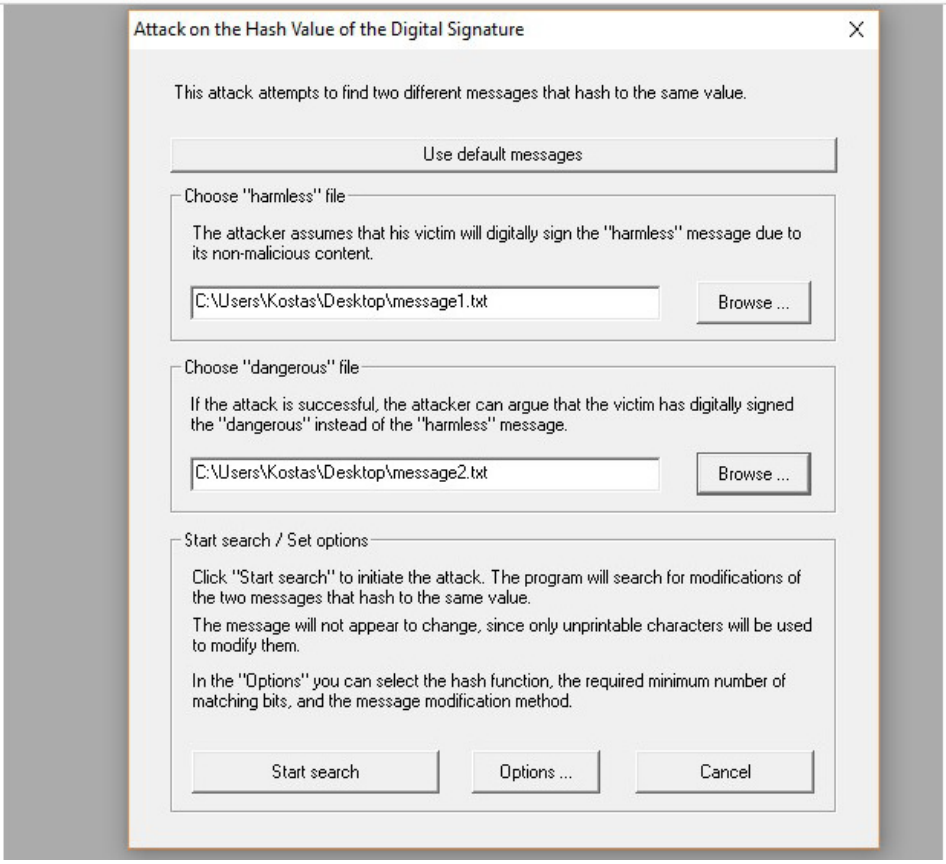
Η συνάρτηση αυτή δεν ικανοποιεί το 2nd-preimageresistance – και, ως εκ τούτου, ούτε το collisionresistance. Για οποιοδήποτε μήνυμα Μ, είναι πολύ εύκολο να βρούμε ένα άλλο μήνυμα Μ’ με το ίδιο αποτύπωμα με το Μ. Συγκεκριμένα, κάθε μήνυμα Μ’ που ταυτίζεται με το Μ στο 1ο, 3ο, 5ο, 7ο,

…, 255ο ψηφίο, ανεξαρτήτως στις τιμές που θα έχει στα άλλα ψηφία, θα έχει το ίδιο αποτύπωμα με το Μ («σύγκρουση»).

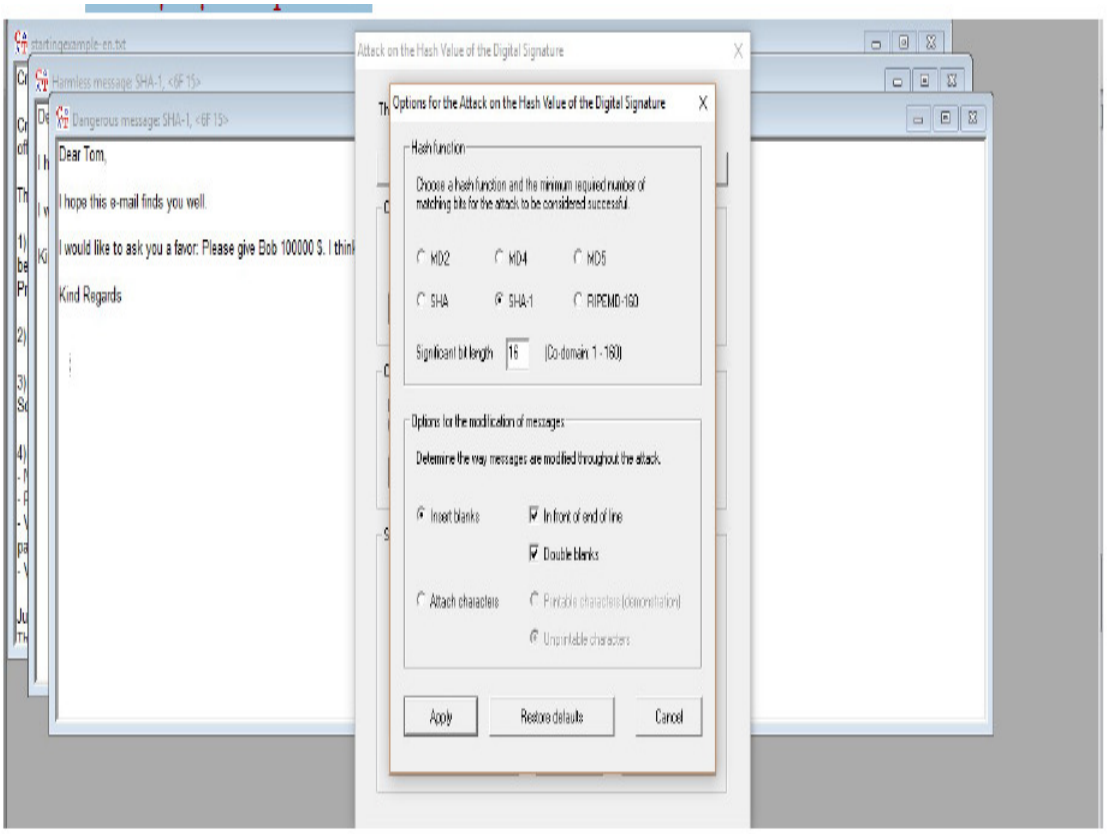
5.2 ΑΣΚΗΣΗ 2

Κατασκευάζουμε δύο μηνύματα, το ένα («message1.txt») αναφέρει:
Dear Tom,
I hope this e-mail finds you well.
I would like to ask you a favor: Please give John 10000 $ from my account. I think that he deserves this.
Kind Regards
και το άλλο («message2.txt») αναφέρει
Dear Tom,
I hope this e-mail finds you well.
I would like to ask you a favor: Please give Bob 100000 $ from my account. I
think that he deserves this.
Kind Regards

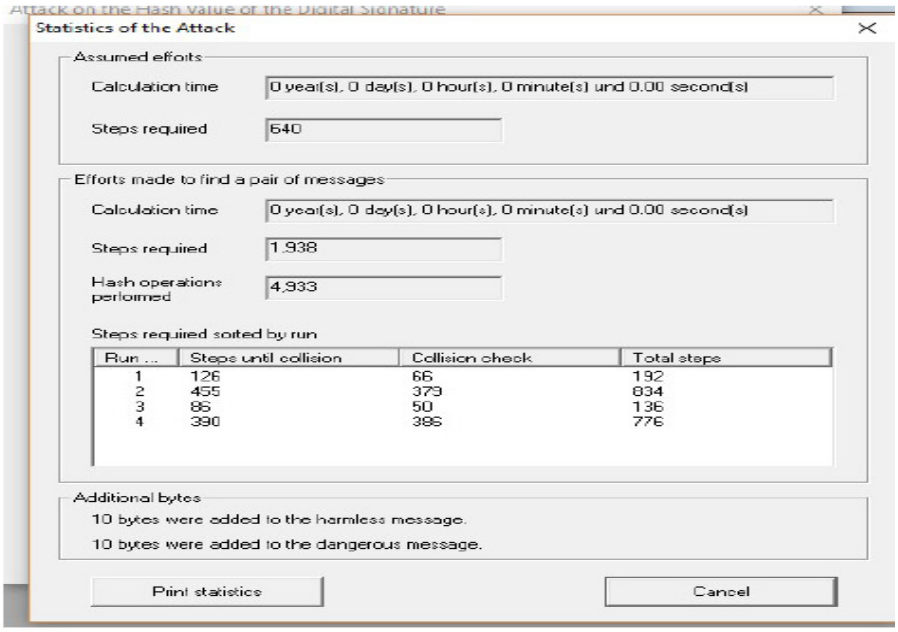
Έχουμε λοιπόν:



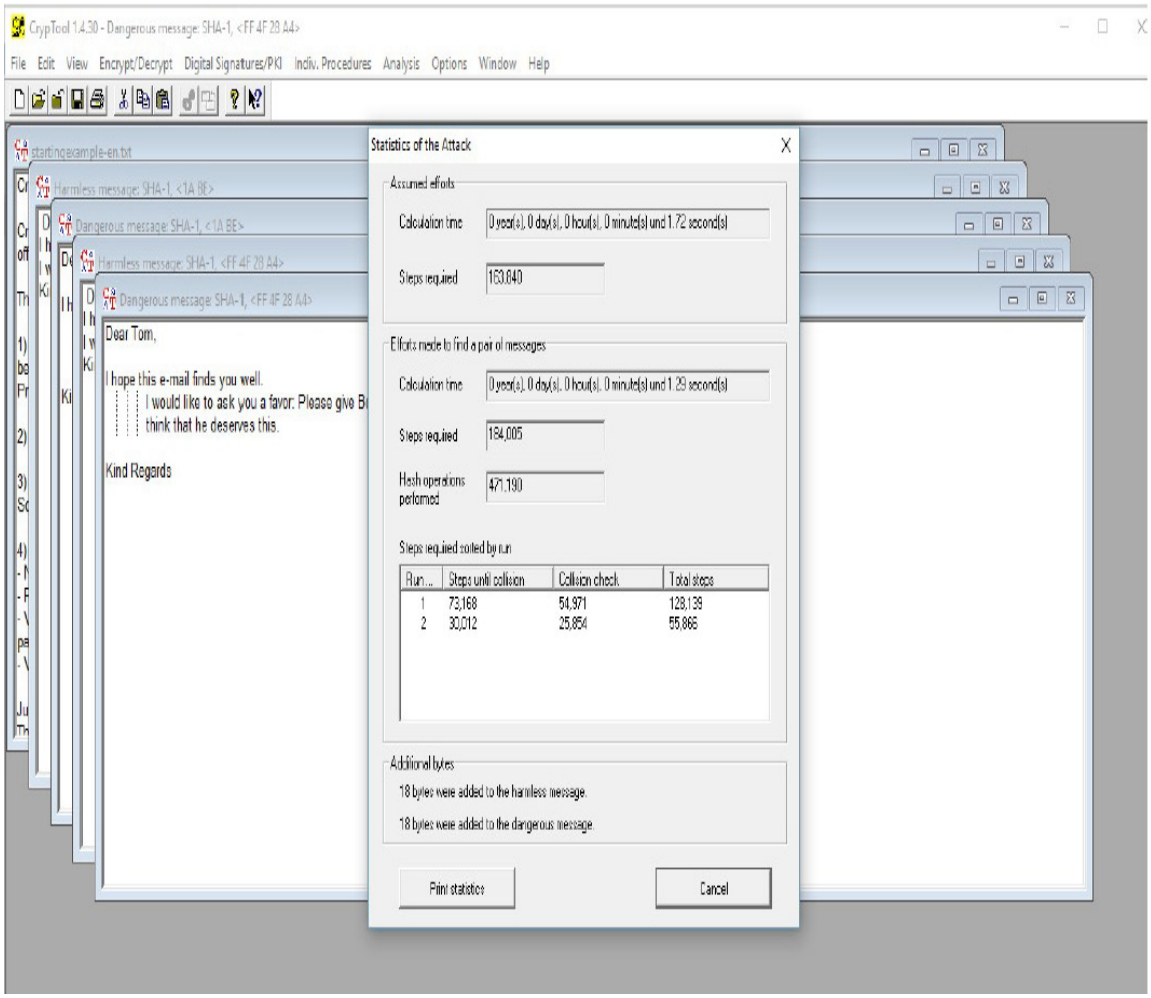
Επιλέγουμε «Options»:



Δηλώνουμε SHA-1 και πλήθος bits ταύτισης των αποτυπωμάτων: 16 bits
Στη συνέχεια, πατώντας «Startsearch» ακαριαία τροποποιούνται τα μηνύματα, με εισαγωγή κενών χαρακτήρων, ώστε τα αποτυπώματά τους να ταυτίζονται σε 16 bits.



Για μέγεθος ταύτισης 32 bit, πάλι η τροποποίηση των μηνυμάτων έγινε γρήγορα (σε 1,29 sec).



5.3 ΑΣΚΗΣΗ 3

Α) Δεν είναι καλή ιδέα, γιατί οποιαδήποτε δύο διαφορετικά μηνύματα συμφωνούν στο τελευταίο μπλοκ (δηλαδή στα τελευταία 128 bit αν το μέγεθος του Μ είναι πολλαπλάσιο των 128 bit) θα έχουν τον ίδιο MAC (και, προφανώς, για δοθέν μήνυμα μπορούμε να βρούμε πολλά άλλα με τον ίδιο MAC).

Β) Δεν είναι καλή ιδέα, γιατί – ανακαλώντας τη λειτουργία των streamciphers,
όπως είναι ο RC4 - οποιαδήποτε δύο διαφορετικά μηνύματα συμφωνούν στα
τελευταία 256 bit θα έχουν τον ίδιο MAC (και, προφανώς, για δοθέν μήνυμα μπορούμε να βρούμε πολλά άλλα με τον ίδιο MAC).
Γ) Δεν είναι καλή ιδέα, γιατί οποιαδήποτε δύο διαφορετικά μηνύματα συμφωνούν στα πρώτα δύο μπλοκ (δηλαδή στα πρώτα 256 bit αν το μέγεθος του Μ είναι μεγαλύτερο των 256 bit) θα έχουν τον ίδιο MAC (και, προφανώς,
για δοθέν μήνυμα μπορούμε να βρούμε πολλά άλλα με τον ίδιο MAC).

6.1 ΑΣΚΗΣΗ 1

(21 + 12) mod 30 = 33 mod 30 =3

(6-10) mod 13 = -4 mod 13 ≡ 9
(γιατί -4 mod 13 ≡ (-4+13) mod 13 = 9 mod 13 =9)

(10-2*12) mod 21 = (10-24) mod 21 ≡ (10-3) mod 21≡ 7
(*γιατί 24 mod 21 ≡3 mod 21 = 3. Στο ίδιο τελικό αποτέλεσμα θα καταλήγαμε ακόμα αν κάναμε την πράξη (10-24) mod 21 = -14 mod 21 και συνεχίζαμε αναλόγως).*

6.2 ΑΣΚΗΣΗ 2

Αφού ο 17 είναι πρώτος αριθμός (δεν διαιρείται από κανέναν ακέραιο πλην του εαυτού του και της μονάδας), το ανωτέρω σύνολο είναι πεπερασμένο σώμα, δηλαδή ορίζονται όλες οι πράξεις (συγκεκριμένα, πρόκειται για το πεπερασμένο σώμα GF(17)). Επιλέγουμε τυχαία τα στοιχεία a=15 και b=11.
Έχουμε:
Πρόσθεση: (15 + 11) mod 17 = 26 mod 17 = 9
Αφαίρεση: (15 – 11) mod 17 = 4 mod 17 = 4
Πολλαπλασιασμός: (15 x 11) mod 17 = 165 mod 17 = 12
Διαίρεση: 15/11 mod 17 = (15 x 11-1) mod 17.
Πρέπει να βρούμε τον αριθμό 11^{-1}mod 17
Από το διαδικτυακό τόπο http://www.dcode.fr/modular-inverse
μπορούμε να δούμε ότι 11^{-1}mod 17 = 14
(Πράγματι: (14 x 11) mod 17 = 154 mod 17 = 1).
Άρα, 15/11 mod 17 = (15 x 14) mod 17 = 210 mod 17 = 6.
Άρα, για τα ανωτέρω a, b στοιχεία του GF(17), έχουμε a+b=9, a-b=4,
a*b=12 και a/b=6.

7.1 ΑΣΚΗΣΗ 1

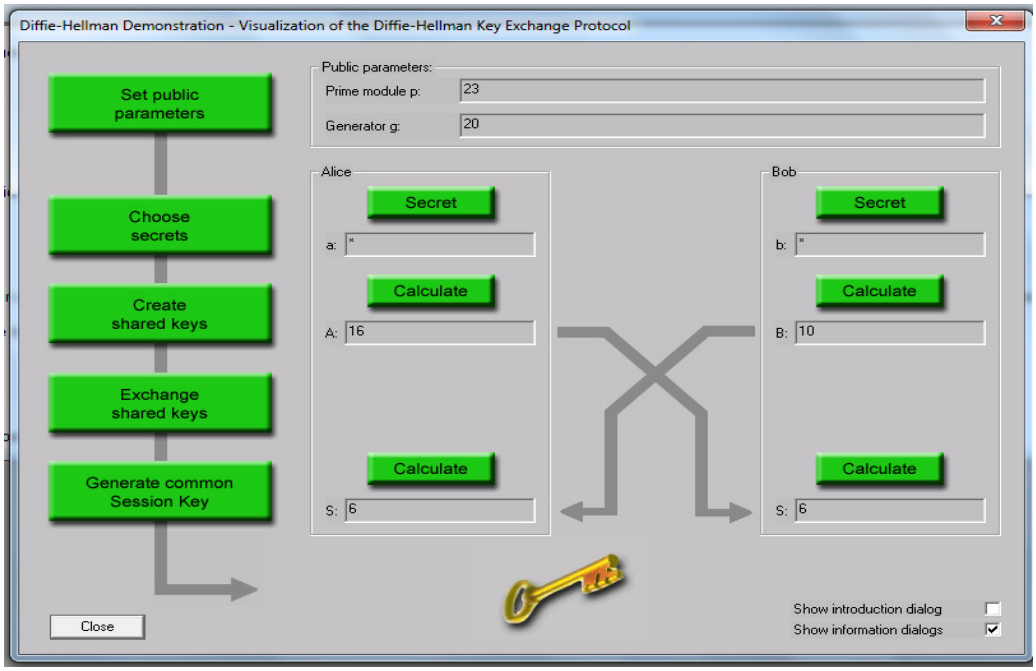
Α)Οι μικροί αριθμοί επιτρέπουν, με εξαντλητικούς ελέγχους, να υπολογίσουμε τα
άγνωστα x,y (ένα μόνο εξ αυτών να υπολογίσουμε, μας αρκεί για τον υπολογισμό του μυστικού κλειδιού που αντάλλαξαν). Για τον υπολογισμό του x, κάνουμε τις δοκιμές 20^{k}mod 23 για όλα τα k από 1 μέχρι 22, μέχρι να βρούμε αποτέλεσμα 10. Στην περίπτωση αυτή, θα βρούμε k=5, αφού

$20^5 \text{mod} 23 = 10$. Άρα, x=5. Το μυστικό κλειδί $g^{xy}\text{mod}p$ ισούται επίσης με $(y')^x \text{mod}p$, δηλ. εν προκειμένω ισούται με 165 mod 23 = 6. Άρα, το μυστικό κλειδί είναι 6.

Σε κάθε περίπτωση πάντως, με εξαντλητικές δοκιμές μπορούμε να βρούμε και το y με ανάλογο τρόπο. Κάνοντας τις δοκιμές θα βρούμε y=6 (αφού $20^6 \text{mod} 23 = 16$).

Μπορούμε να επιβεβαιώσουμε ότι και ο χρήστης B θα υπολογίσει το ίδιο μυστικό κλειδί $g^{xy}\text{mod}p = (x')^y \text{mod}p = 106 \text{ mod } 23 = 6$.

B) Χρησιμοποιώντας το Cryptool και εισάγωντας τα δεδομένα που μας δόθηκαν πήραμε :



At first, Alice and Bob agreed on the public parameters. So they chose a prime p and a generator g:

p: 23

g: 20

Alice chose her secret number 'a' while Bob chose his secret number 'b':
a: 6

b: 5

If the chosen secret values a and b are greater or equal the prime module p, then they need to be reduced modulo p. The actual values are given below:

a (reduced mod p):
6

b (reduced mod p):
5

On the basis of the previously chosen secret numbers, Alice and Bob created their respective shared keys. Alice computed her shared key A, while Bob computed his shared key B:
A: 16

B: 10

In order to calculate their secret and common Session Key, Alice and Bob exchanged their shared keys: Alice sent her shared key A to Bob and Bob sent his shared key B to Alice.

Alice and Bob were able to calculate the secret and common Session Key now. Alice computed the Session Key SA, Bob computed the Session Key SB:

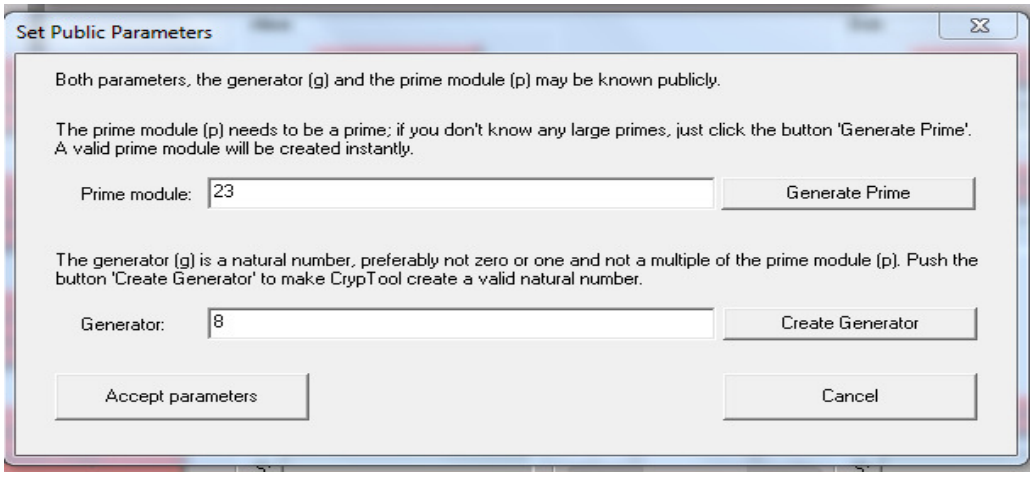
SA: 6

SB: 6

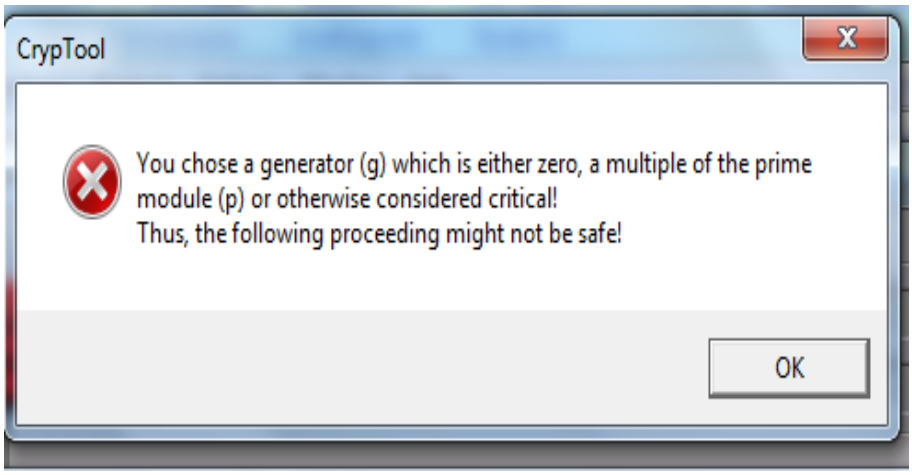
Theoretically it is now possible for Alice and Bob to use their Session Keys to encrypt documents they would like to exchange covertly.

Παρήχθησαν δηλαδή τα αποτελέσματα που βρήκαμε θεωρητικά.

Γ) Επιχειρούμε την υλοποίηση του πρωτοκόλλου Diffie –Hellman με παραμέτρους p=23 και g=8 :



Και παίρνουμε το μήνυμα :



Το οποίο μας προειδοποιεί ότι δεν είναι αποδεκτή η τιμή για τον γεννήτορα g=8 .

Η εξήγηση προκύπτει από το γεγονός ότι ο γεννήτορας g=8 δεν είναι κατάλληλος γιατι δεν ικανοποιεί την απαραίτητη προυπόθεση : $g^K \pmod{p} \neq 1$ για όλα τα k που είναι διαιρέτες του p-1. Οι διαιρέτες του p-1 = 22 είναι το 2 και το 11 . Όμως αν κάνουμε την πράξη, $8^{11} \bmod 23 = 1$, συνεπώς η επιλογή του g=8 ως γεννητορα αποκλείεται.

7.2 ΑΣΚΗΣΗ 2

Α) Για τους δύο χρήστες Α και Β γνωρίζουμε τα δημόσια κλειδια τους e_A= 7 και e_B = 5 καθώς και τα N_A = 5371 και N_B= 2581. Χρησιμοποιούμε το Cryptool (Individual Procedures -> RSA Demonstration -> Factorization of a number) και βρίσκουμε τους πρώτους παράγοντες :

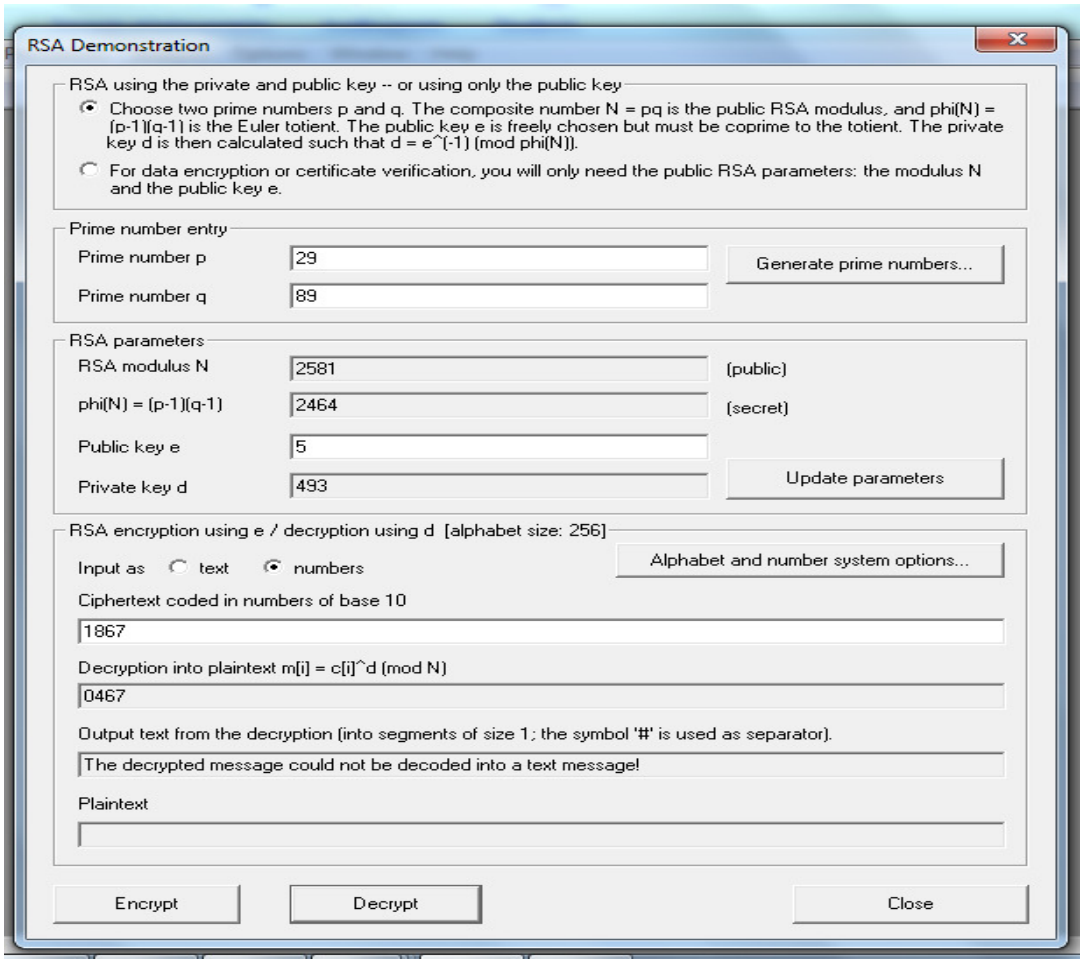
N_A= 41*131 και N_B=29*89 , άρα οι αντίστοιχες συναρτήσεις Φ(N) είναι :

Φ_A = 40 * 130 = 5200 και Φ_B= 28*88 = 2464 . Με αυτά τα δεδομένα υπολογίζουμε το ιδιωτικό κλειδί του κάθε χρήστη από τη σχέση **: Φ(N) = e*d -1**➔ $\Phi_A = 7 * d_A - 1$ ➔5200 = 7*d-1 ➔**d_A= 743** και $\Phi_B = 5*d_B - 1$ ➔ 2464 = $5*d_B$ -1 ➔**d_B = 493**

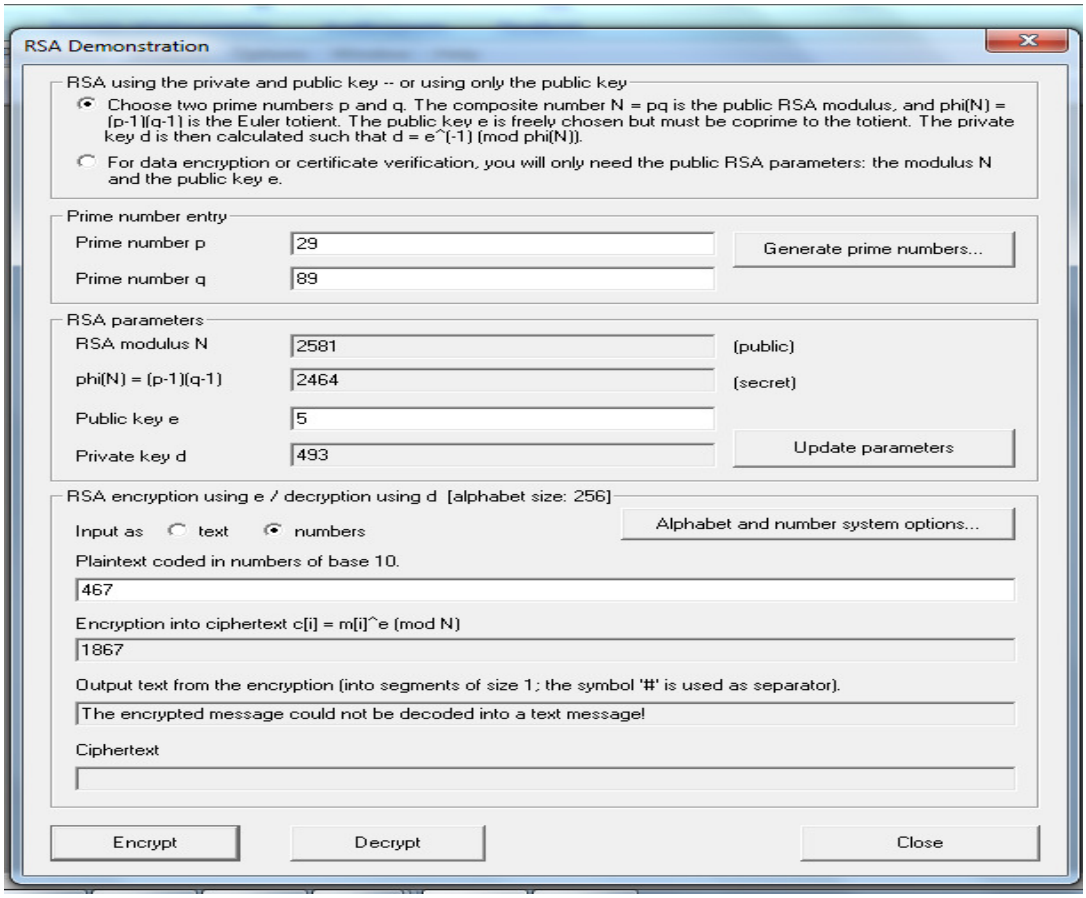
Β) Δεδομένου ότι έχουμε υποκλέψει τα κλειδιά του χρήστη Β μπορούμε με αυτά να διαβάσουμε όποιο μήνυμα στέλνει. Γνωρίζουμε λοιπόν ότι e_B = 5 , N_B= 2581 και d_B = 493 Γνωρίζουμε επίσης τους τύπους κρυπτογράφησης και αποκρυπτογράφησης μηνύματος στον RSA αλγόριθμο : $c = m^e \text{mod} N$ και $m = c^d \text{mod} N$ αντίστοιχα. Έχωντας υπολκλέψει το κρυπτοκείμενο c= 1867 μπορούμε να βρούμε το μήνυμα m : $m = 1867^{493} \text{mod } 2581 = 467$

m = 467

Για επαλήθευση χρησιμοποιούμε το Cryptool (Encrypt➔Asymmetric➔RSADemonstration) και έχουμε :



Αλλά και αντίστροφα :



7.3 ΑΣΚΗΣΗ 3

Αρχικά θα αποδείξουμε την ταυτότητα : p+q = N – φ(N) +1 (1)

Γνωρίζουμε ότι N= p*q και φ(N) = (p-1) * (q-1) . Με αντικατάσταση στην (1) έχουμε :

p+q= p*q –(p-1)*(q-1) +1 ➔p+q = p*q – (p*q –p –q +1) +1 ➔

p+q = p*q –p*q+p+q -1+1➔p+q =p+q (Q.E.D)

από την ταυτότητα που αποδείξαμε παίρνουμε : p + q = 171950281 – 171923092 +1 = 27190

άρα p +q = 27190 . Ξέρουμε λοιπόν ότι οι p , q είναι πρώτοι αριθμοί και έχουν άθροισμα 27190. Χρησιμοποιούμε το εργαλείο http://www.math.com/students/calculators/source/prime-number.htm (είναι υπολογιστής εύρεσης πρώτων αριθμών και όχι εργαλείο παραγοντοποίησης)

Και δοκιμάζουμε διαδοχικά¨

για p=3 ➔q= 27187 ➔notaprime

για p=5 ➔ q= 27185 ➔ not a prime

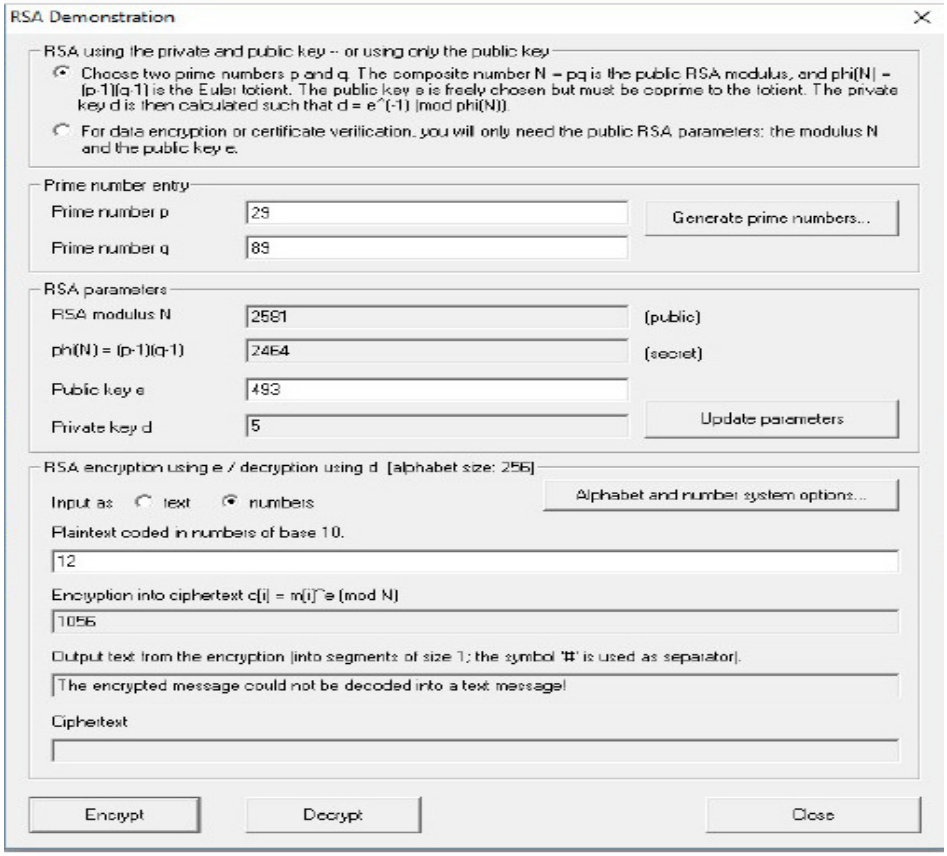
για p= 7 ➔ q= 27183 ➔not a prime

για p=11 ➔q = 27179 ➔ 27179 is prime!

Άρα έχουμε : p = 11 και q = 27179 ή p= 27179 και q = 11

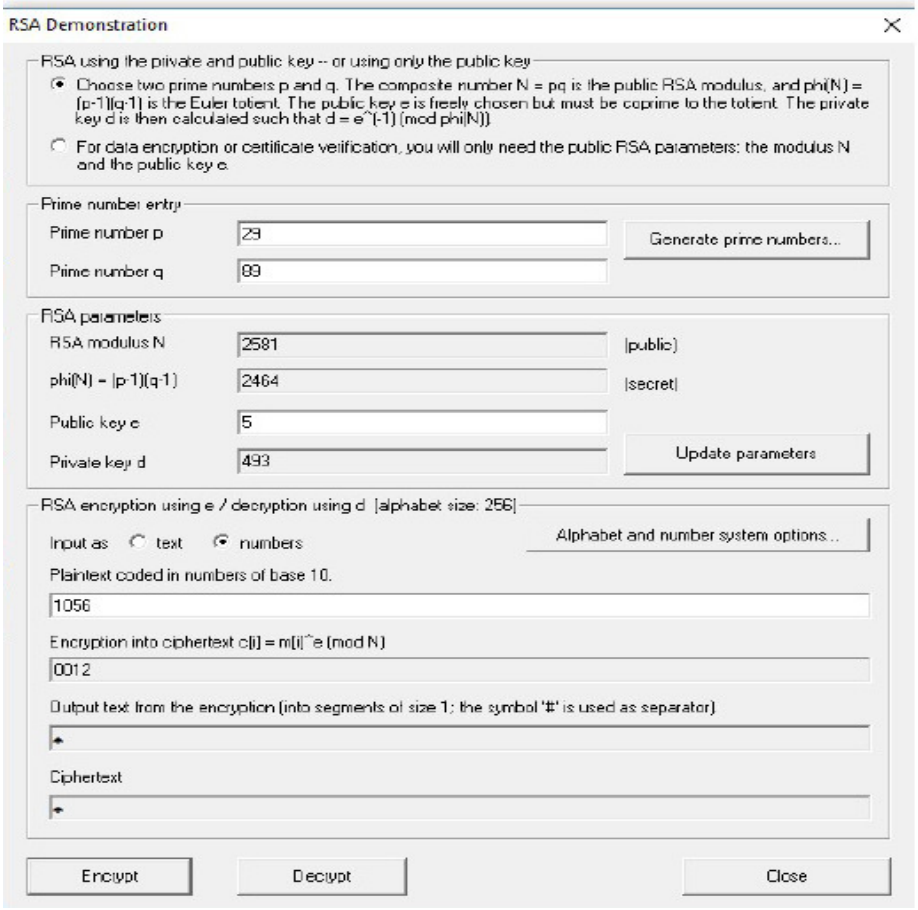
8.1 ΑΣΚΗΣΗ 1

Α) Η ψηφιακή υπογραφή στον RSA είναι κρυπτογράφηση με το ιδιωτικό μας κλειδί, δηλαδή (H(m))dmodN. Ο Β, όπως είδαμε στην Άσκηση 7, έχει ιδιωτικό κλειδί d=493, ενώ επίσης για το δημόσιο κλειδί του N ισχύει N=2581. Άρα, η υπογραφή s ισούται με s=12493 mod 2581. Αξιοποιούμε το Cryptool (δηλώνοντας ότι κρυπτογραφούμε το μήνυμα 12 με αυτό το κλειδί, προκειμένου να γίνει αυτή η πράξη):



Άρα, η υπογραφή είναι s=1056.

Β) Ο Α θα κάνει την πράξη $s^e \bmod N$, όπου e, N το δημόσιο κλειδί του αποστολέα Β. Άρα, θα κάνει την πράξη 10565 mod 2581.



Θα βρει το σωστό αποτύπωμα του μηνύματος 12.

Γ) Ο Β επιλέγει τυχαίο r. Έστω r=2.

Υπολογίζει το r^emodN, όπου (e,N) το δημόσιο κλειδί του υπογράφοντα Α.

Άρα (ανακαλούμε τα στοιχεία του Α από την Άσκηση 7.2):

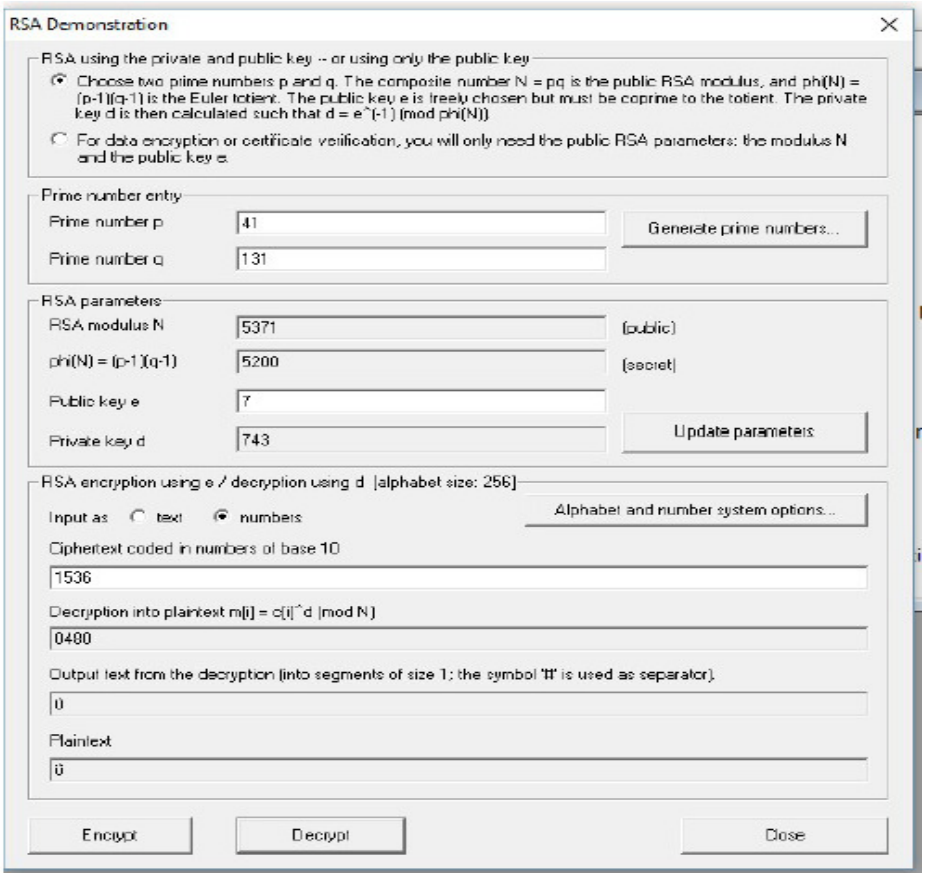
2^7mod 5371 = 128 mod 5371 = 128.

Ακολούθως, στέλνει στον Β το mr^e (mod N) = 12 x 128 (mod 5371) = 1536.

Ο Α λαμβάνει το «συσκοτεισμένο» μήνυμα 1536 – δεν μπορεί να ξέρει ποιο είναι το αρχικό μήνυμα (δηλ. το 12). Υπογράφει το 1536 με το ιδιωτικό του κλειδί d, το οποίο είναι το 743 (βλ. πάλι Άσκηση 7.2).

1536^{743} (mod 5371) = 480 (βλ. επόμενη εικόνα – επιλέξαμε «decrypt» για να

γίνει η επιθυμητή πράξη).



Ο Β λαμβάνει την τυφλή υπογραφή 480 και τη διαιρεί με το r=2, δηλ. πολλαπλασιάζει με τον αντίστροφο του r. 480 x 2^{-1} (mod 5371) = 480 x 2686 (mod 5371) = 240. Επαλήθευση: το 240 είναι πράγματι η υπογραφή που θα έβαζε ο Α στο μήνυμα 12, αν το γνώριζε. Πράγματι, η υπογραφή που θα έβαζε θα ήταν:

$12^{743} \text{mod } 5371 = 240$ (επαληθεύεται και αυτό με το cryptool).

8.2 ΑΣΚΗΣΗ 2

Το κρίσιμο σημείο εδώ είναι ότι κατά την κρυπτογράφηση πρέπει να δηλώσουμε τον παραλήπτη – δηλαδή το δημόσιο κλειδί αυτού και όχι το δικό μας. Μόνο αν θέλουμε να κρυπτογραφήσουμε κάτι το οποίο να μπορούμε να αποκρυπτογραφήσουμε μόνο εμείς, χρησιμοποιούμε το δικό μας δημόσιο κλειδί για την κρυπτογράφηση. Εφόσον θέλουμε να υπογράψουμε το μήνυμά μας, θα μας ζητηθεί το ιδιωτικό μας κλειδί (δηλαδή να εισάγουμε το passphrase μας).

8.3 ΑΣΚΗΣΗ 3

OVerifier θα υπολογίσει αρχικά το $y^2 \bmod N = 5^2 \bmod 55 = 25$ Ακολούθως, θα υπολογίσει το $xu^e \bmod N = 4 \times 161 \bmod 55 = 64 \bmod 55 = 9$. Άρα, η απάντηση δεν είναι σωστή.

13.1 ΑΣΚΗΣΗ 1

Α). Το αρχείο που δημιουργούμε έχει μέγεθος 16 byte = 128 bit, δηλαδή όσο και το μέγεθος ενός block στον AES. Το αποτέλεσμά μας θα είναι επίσης μεγέθους 16 byte (128 bit), αφού θα χρησιμοποιήσουμε τις παραμέτρους «nopad» και «nosalt». Τα κρυπτοκείμενα που προκύπτουν αναμένεται να είναι «τυχαίου» - χωρίς κάποια αναγνωρίσιμη δομή – περιεχομένου. Για παράδειγμα, αν το αρχείο "name.txt" περιέχει το ονοματεπώνυμο «kostaslimniotis» (μεγέθους ακριβώς 16 bytes), μπορούμε να χρησιμοποιήσουμε την εντολή

openssl enc –aes-128-cbc –K 0123456789abcdef0123456789abcdef –iv 00000000000000000000000000000000 –in name.txt –out encrypted_cbc.bin –nopad - nosalt

για κρυπτογράφησή του με τον CBC τρόπο λειτουργίας, με κλειδί 0123456789abcdef0123456789abcdef μεγέθους 128 bit και μηδενικό διάνυσμα αρχικοποίησης. Με αντίστοιχη εντολή προκύπτουν και οι κρυπτογραφήσεις με τους τρόπους λειτουργίας ECB και OFB (προσέξτε ότι στον ECB δεν χρειάζεται διάνυσμα αρχικοποίησης). Με hexeditors μπορούμε να δούμε, στη 16δική αναπαράσταση της Ascii κωδικοποίησης των χαρακτήρων, τα αποτελέσματα.

Κρυπτογράφηση με ECB: 31 F2 A2 B5 79 74 15 DF DF B2 59 88 CA 4C 1C 43

Κρυπτογράφηση με CBC: 31 F2 A2 B5 79 74 15 DF DF B2 59 88 CA 4C 1C 43

Κρυπτογράφηση με OFB: 12 C4 B6 B6 59 1B 8D E8 BA E5 A0 08 7E 7E 0B 07

Παρατηρείστε ότι το κρυπτοκείμενο με ECB και CBC τρόπο λειτουργίας είναι όμοια.

Αυτό είναι αναμενόμενο, αφού κρυπτογραφείται μόνο ένα block και στον CBC το διάνυσμα αρχικοποίησης είναι το μηδενικό (οπότε η είσοδος στον κρυπτογραφικό αλγόριθμο είναι αυτούσιο το αρχικό block, όπως και στον CBC. Στον OFB η κατάσταση είναι διαφορετική, ακριβώς γιατί η κρυπτογράφηση είναι με διαφορετικό τρόπο – πρώτα κρυπτογραφείται το μηδενικό διάνυσμα αρχικοποίησης και μετά προστίθεται το block του μηνύματος.

Β) Το αρχείο repeated_name.txt έχει μέγεθος 5x16 byte, δηλαδή αποτελείται από 5 blocks των 128 bits, τα οποία είναι και όμοια μεταξύ τους (το κάθε block είναι ουσιαστικά όμοιο με το αρχικό name.txt του

ερωτήματος Α). Εκτελούμε αντίστοιχη εντολή – π.χ., κατ' αναλογία με το ερώτημα i, στον CBC τρόπο λειτουργίας θα έχουμε:

openssl enc –aes-128-cbc –K 0123456789abcdef0123456789abcdef –iv 00000000000000000000000000000000 –in repeated_name.txt –out new_encrypted_cbc.bin –nopad –nosalt

ενώ, με ανάλογο τρόπο, κάνουμε και τις ECB και OFB κρυπτογραφήσεις. Τα αποτελέσματα είναι τα εξής (κάθε γραμμή αντιστοιχεί σε ένα block κρυπτοκειμένου, μεγέθους 128 bit):

Κρυπτογράφηση με ECB:

31 F2 A2 B5 79 74 15 DFDFB2 59 88 CA 4C 1C 43

31 F2 A2 B5 79 74 15 DF DF B2 59 88 CA 4C 1C 43

31 F2 A2 B5 79 74 15 DF DF B2 59 88 CA 4C 1C 43

31 F2 A2 B5 79 74 15 DF DF B2 59 88 CA 4C 1C 43

31 F2 A2 B5 79 74 15 DF DF B2 59 88 CA 4C 1C 43

Κρυπτογράφηση με CBC:

31 F2 A2 B5 79 74 15 DF DF B2 59 88 CA 4C 1C 43

27 7B F1 57 FD FB 28 01 BE 4D 9B C8 4D 72 F0 8C

79 A5 B8 1D B9 BE 87 CE AB ED 5A 73 ED 5A AB 73

FD DC BC E8 68 2C DF F8 55 BB 09 27 93 14 63 CC

90 AE 9B 0E CC F0 7F 7A 59 EC 70 C4 00 E9 6D 31

Κρυπτογράφηση με OFB:

12 C4 B6 B6 59 1B 8D E8 BA E5 A0 08 7E 7E 0B 07

62 B1 D3 EA 8C BE 9F 6B 4E E9 F0 68 F9 7E 5B 26

9D F7 DF EA 73 64 0E D4 CE C5 26 F8 52 17 F1 53

0E 41 4E 75 AF 19 EE 29 1E 3F D9 EF D8 37 2D A5

50 EB A2 D2 91 4E 8D 90 19 9C 5C 85 9E DE 37 E9

Παρατηρούμε τα εξής:

Στον ECB, όλα τα block του κρυπτοκειμένου είναι όμοια μεταξύ τους – αναμενόμενο, αφού και τα block του μηνύματος είναι όμοια μεταξύ τους. Επίσης, τα block του κρυπτοκειμένου είναι όμοια με το block κρυπτοκειμένου του ερωτήματος Α – επίσης λογικό και αναμενόμενο, βάσει του πώς λειτουργεί ο ECB . Στον CBC, το πρώτο block είναι – όπως και στο ερώτημα Α), για τους ίδιους λόγους – όμοιο με το πρώτο block του κρυπτοκειμένου του ECB, αλλά στη συνέχεια όλα τα block είναι διαφορετικά μεταξύ τους, παρά το γεγονός ότι τα block του μηνύματος είναι ίδια. Αυτό είναι λογικό και αναμενόμενο, βάσει του πώς λειτουργεί ο CBC.

Στον OFB, επίσης όλα τα block είναι διαφορετικά μεταξύ τους , παρά το γεγονός ότι τα block του μηνύματος είναι ίδια. Το πρώτο block του OFB είναι όμοιο με αυτό που είδαμε στην OFB κρυπτογράφηση στο ερώτημα Α (δεν θα μπορούσε να είναι διαφορετικό – κρυπτογραφείται το ίδιο block με το ίδιο διάνυσμα αρχικοποίησης και το ίδιο κλειδί).

Γ) Δημιουργούμε τα ακόλουθα αρχεία (με κίτρινο φόντο η αλλαγή που επιφέραμε):

Τροποποιημένο ECB κρυπτοκείμενο (αρχείο new_encrypted_ecb_modified.bin):

B1 F2 A2 B5 79 74 15 DF DF B2 59 88 CA 4C 1C 43

31 F2 A2 B5 79 74 15 DF DF B2 59 88 CA 4C 1C 43

31 F2 A2 B5 79 74 15 DF DF B2 59 88 CA 4C 1C 43

31 F2 A2 B5 79 74 15 DF DF B2 59 88 CA 4C 1C 43

31 F2 A2 B5 79 74 15 DF DF B2 59 88 CA 4C 1C 43

Τροποποιημένο CBC κρυπτοκείμενο (αρχείο new_encrypted_cbc_modified.bin):

B1 F2 A2 B5 79 74 15 DF DF B2 59 88 CA 4C 1C 43
27 7B F1 57 FD FB 28 01 BE 4D 9B C8 4D 72 F0 8C
79 A5 B8 1D B9 BE 87 CE AB ED 5A 73 ED 5A AB 73
FD DC BC E8 68 2C DF F8 55 BB 09 27 93 14 63 CC
90 AE 9B 0E CC F0 7F 7A 59 EC 70 C4 00 E9 6D 31

Τροποποιημένο OFB κρυπτοκείμενο (αρχείο new_encrypted_ofb_modified.bin):

92 C4 B6 B6 59 1B 8D E8 BA E5 A0 08 7E 7E 0B 07
62 B1 D3 EA 8C BE 9F 6B 4E E9 F0 68 F9 7E 5B 26
9D F7 DF EA 73 64 0E D4 CE C5 26 F8 52 17 F1 53
0E 41 4E 75 AF 19 EE 29 1E 3F D9 EF D8 37 2D A5
50 EB A2 D2 91 4E 8D 90 19 9C 5C 85 9E DE 37 E9

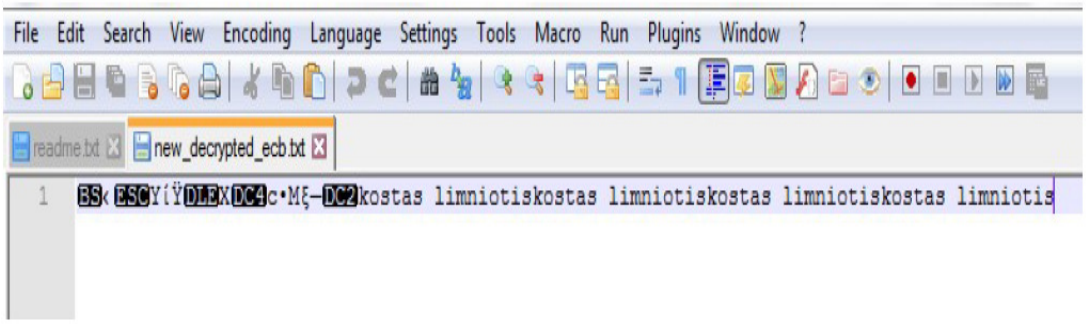
Όλες ο αλλαγές που επιφέραμε αντιστοιχούν σε μεταβολή του πρώτου bit (αυτό γίνεται φανερό αν σε κάθε δεκαεξαδικό χαρακτήρα αναλογιστούμε την ισοδύναμη δυαδική του αναπαράσταση).

Η εντολή που θα εκτελέσουμε για τη CBC αποκρυπτογράφηση – αντίστοιχα εκτελείται και για την ECB και για την OFB – είναι η εξής:

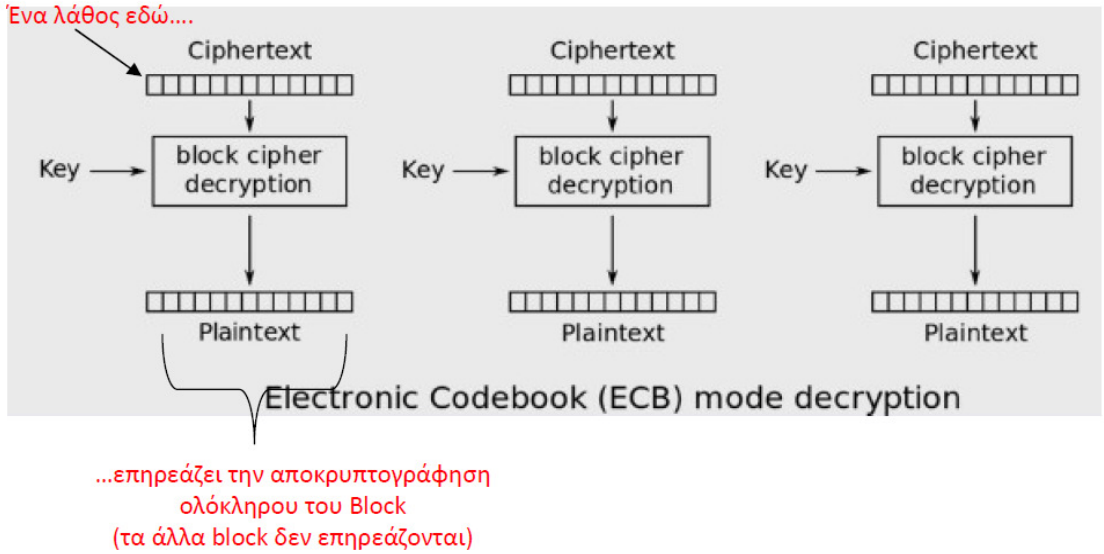
openssl enc –aes-128-cbc -d –K 0123456789abcdef0123456789abcdef –iv 00000000000000000000000000000000 –in new_encrypted_cbc_modified.bin –out new_decrypted_cbc.txt –nopad –nosalt

Ανακτούμε λοιπόν τα εξής μηνύματα, κατά σειρά για τις ECB, CBC, OFB αποκρυπτογραφήσεις:

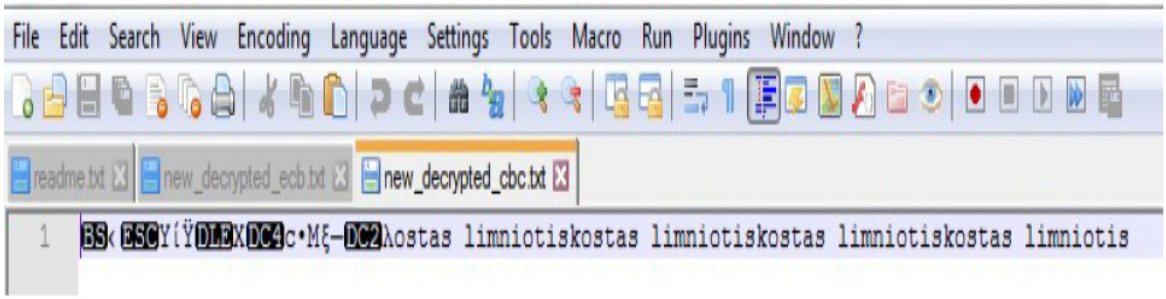
Αρχείο new_dercypted_ecb.txt :



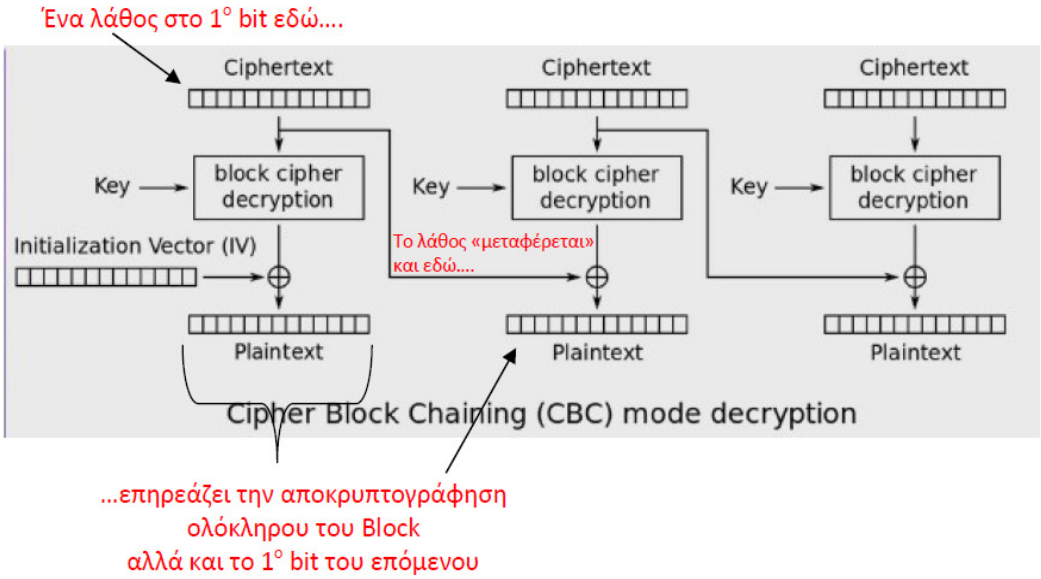
Βλέπουμε ότι αποκρυπτογραφείται λανθασμένα ολόκληρο το πρώτο block του μηνύματος, αλλά δεν επηρεάζονται καθόλου τα υπόλοιπα. Αυτό είναι απόλυτα λογικό, αν αναλογιστούμε το πώς γίνεται η αποκρυπτογράφηση στον ECB:



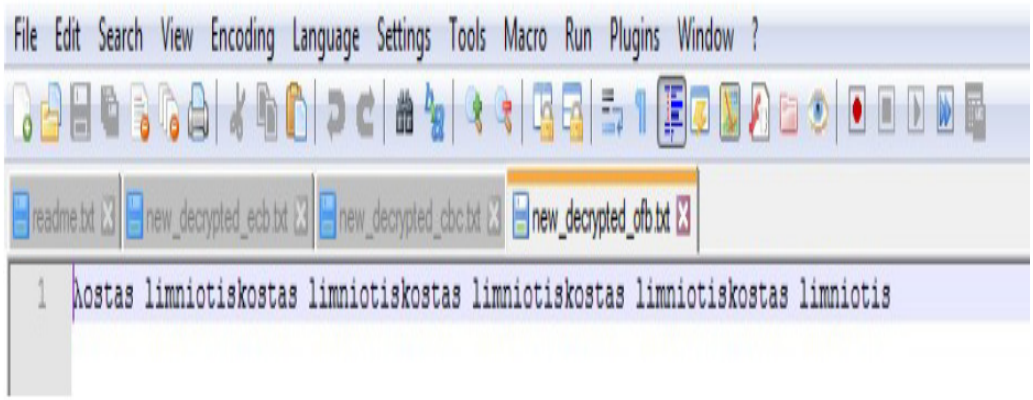
Αρχείο new_dercypted_cbc.txt :



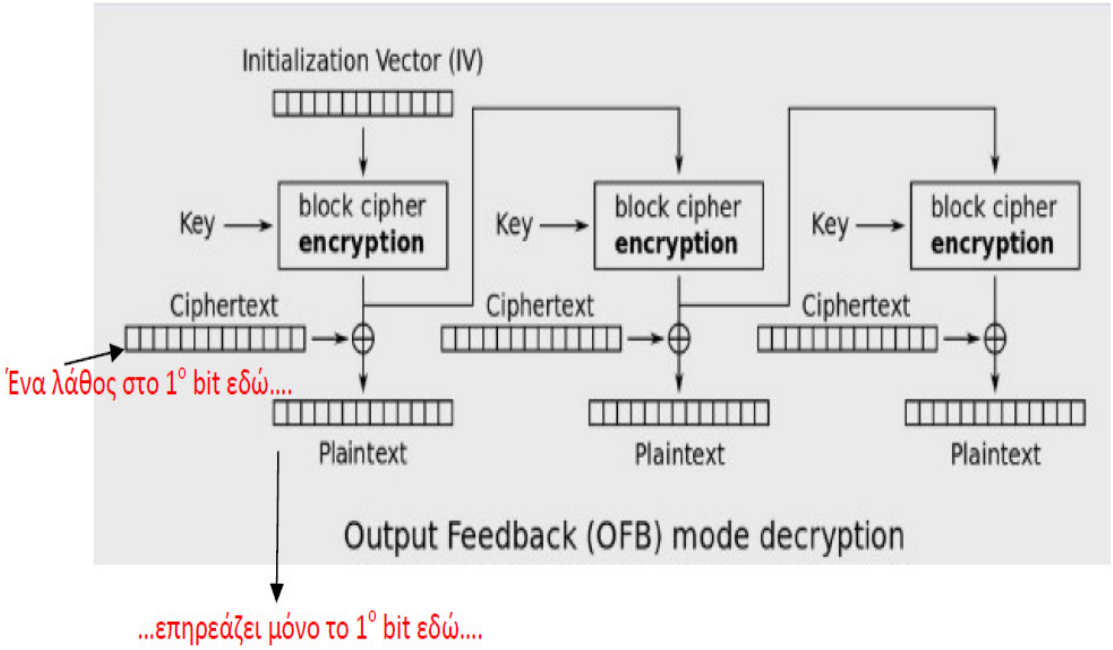
Βλέπουμε ότι αποκρυπτογραφείται λανθασμένα ολόκληρο το πρώτο block του μηνύματος (για τους ίδιους λόγους που ισχύει και στον ECB τρόπο), καθώς επίσης και το πρώτο bit του δεύτερου block – οπότε, αφού επηρεάζεται το πρώτο bit αυτού, αποκρυπτογραφείται τελικά λανθασμένα ολόκληρος ο πρώτος χαρακτήρας αυτό. Αυτό είναι απόλυτα λογικό, αν αναλογιστούμε το πώς γίνεται η αποκρυπτογράφηση στον CBC:



Αρχείο new_dercypted_ofb.txt :

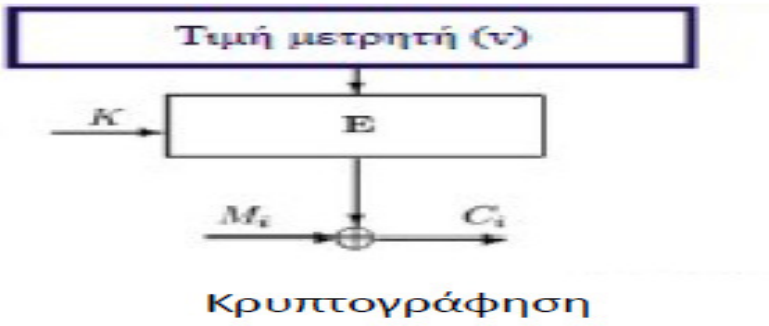


Βλέπουμε ότι αποκρυπτογραφείται λανθασμένα μόνο ο πρώτος χαρακτήρας στο πρώτο block του μηνύματος. Ουσιαστικά, επηρεάζεται μόνο το πρώτο bit του μηνύματος, το οποίο είναι απόλυτα λογικό, αν αναλογιστούμε το πώς γίνεται η αποκρυπτογράφηση στον OFB (προσομοιάζει τη λειτουργία streamcipher):



13.2 ΑΣΚΗΣΗ 2

Α) Το κρίσιμο εδώ είναι να αναλογιστούμε τον τρόπο λειτουργίας CTR



Κρυπτογράφηση

Αποκρυπτογράφηση

Ουσιαστικά, ο τρόπος λειτουργίας CTR προσομοιάζει τη λειτουργία streamcipher. Αυτό σημαίνει ότι αν τροποποιηθεί το j-ιοστό bit του block του κρυπτοκειμένου, θα τροποποιηθεί ακριβώς το αντίστοιχο j-ιοστό bit του block του μηνύματος που θα αποκρυπτογραφηθεί. Αυτό είναι αδιάφορο του μεγέθους του κλειδιού που έχει χρησιμοποιηθεί. Η Eve θέλει να τροποποιήσει μόνο τους τελευταίους τρεις χαρακτήρες – δηλαδή τα τελευταία 24 bit – του κρυπτοκειμένου κατά τρόπο τέτοιο ώστε να αποκρυπτογραφούνται σε νέα, επιθυμητή τιμή. Συγκεκριμένα, οι χαρακτήρες «081» να αποκρυπτογραφηθούν σε «198». Συνεπώς, η Eve

αρκεί να τροποποιήσει κατάλληλα μόνο τα τελευταία 24 bit του κρυπτοκειμένου – το υπόλοιπο κρυπτοκείμενο το αφήνει ως έχει.

HAscii κωδικοποίηση των χαρακτήρων 081 είναι **00110000 00111000 00110001**

Αυτοί προστίθενται με XOR με την έξοδο του κρυπτογραφικού αλγορίθμου (βλ. σχήμα κρυπτογράφησης) και το αποτέλεσμα είναι οι χαρακτήρες 85BAA2 (1000010110111010 10100010) – βλ. αρχείο ciphertext1.bin. Άρα, η έξοδος k' του αλγορίθμου κρυπτογράφησης η οποία αντιστοιχεί στους συγκεκριμένους τελευταίους 3 χαρακτήρες – που ισοδυναμεί με την κλειδοροή (keystream) αν κάνουμε τον παραλληλισμό με τους streamciphers – είναι το xor άθροισμα των δύο ανωτέρω, δηλαδή k'= 00110000 00111000 00110001 ⊕10000101 10111010 10100010 =10110101 10000010 10010011 . Η Eve θέλει η αποκρυπτογράφηση να δώσει ως αποτέλεσμα το «198», για το οποίο η ascii κωδικοποίηση των χαρακτήρων του είναι 00110001 00111001 00111000. Άρα, το «νέο» κρυπτοκείμενο c' θέλουμε να ικανοποιεί (στα τελευταία 24 bit αυτού – 3 χαρακτήρες) τη σχέση:

c' ⊕k' = 00110001 00111001 00111000 δηλαδή

c'= k' ⊕00110001 00111001 00111000 = 10110101 10000010 10010011 ⊕ 00110001 00111001 00111000 = 10000100 10111011 10101011 (σε 16δική μορφή 84 BB AB) . Άρα η Eve πρέπει απλά να μετατρέψει τους τελευταίους 24 χαρακτήρες του κρυπτοκειμένου, από 10000101 10111010 10100010 (85 BA A2) σε 10000100 10111011 10101011 (84 BB AB). Συνεπώς, της αρκεί απλά να αλλάξει τα bit που σημειώνονται με κίτρινο: 1000010**1** 1011101**0** 1010**0**01**0**

Β) Η σωστή προσέγγιση θα ήταν η χρήση MAC (π.χ. HMAC), οπότε κάθε αλλοίωση που θα έκανε η Eve στο μήνυμα θα γινόταν αντιληπτή, αφού δεν θα ήταν σε θέση να «πειράξει» και τον MAC κατάλληλα έτσι ώστε οι αλλοιώσεις της να μην γίνονται αντιληπτές (θα έπρεπε να γνωρίζει το μυστικό κλειδί του MAC για να το κάνει αυτό, που δεν το γνωρίζει). Εναλλακτικά, θα μπορούσε να χρησιμοποιηθεί ο τρόπος λειτουργίας GCM (που είναι τροποποιημένος CounterMode), ο οποίος επίσης διασφαλίζει στον παραλήπτη τον έλεγχο της ακεραιότητας του ληφθέντος μηνύματος.
Προσέξτε ότι μία απλή hashfunction δεν μας εξυπηρετεί στη συγκεκριμένη περίπτωση, αφού η παραπάνω τεχνική μπορεί να εφαρμοστεί ακόμα και αν η Alice πριν κρυπτογραφήσει το μήνυμα message.txt είχε υπολογίσει το αποτύπωμα αυτού (με κάποια συνάρτηση κατακερματισμού) και, τελικά, είχε κρυπτογραφήσει τόσο το μήνυμα όσο και το αποτύπωμα. Αφού η Eve γνωρίζει το αρχικό μήνυμα, θα μπορούσε να υπολογίσει το αποτύπωμα αυτού (είτε το αποτύπωμα είχε υπολογιστεί επί του αρχικού μηνύματος είτε επί του κρυπτογραφημένου) και, άρα, θα μπορούσε κάλλιστα, με το ίδιο σκεπτικό, να τροποποιήσει κατάλληλα και το κρυπτογραφημένο αποτύπωμα έτσι ώστε η σύγκριση στον παραλήπτη να είναι έγκυρη και να μην ανιχνευτεί η «αλλοίωση» του μηνύματος. Στην περίπτωση αυτή λοιπόν, θέλουμε hash με κλειδί (δηλαδή MAC).

13.3 ΑΣΚΗΣΗ 3

Ακολουθούν οι εντολές που εκτελέστηκαν από τους διδάσκοντες, για τη δημιουργία του δικού τους δημόσιου κλειδιού (βάσει και των εντολών του σχετικού εγχειριδίου):

openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:3072 -pkeyopt
rsa_keygen_pubexp:7 -out privatekey-tutor.pem
openssl pkey -in privatekey-tutor.pem -out publickey-tutor.pem -pubout
(τέθηκε e=7 – τυχαία επιλογή)

13.4 ΑΣΚΗΣΗ 4

Δημιουργούμε ένα (ψευδο)τυχαίο κλειδί μεγέθους 128 bit (16 byte):
openssslrand 16 -outkey.bin
Κρυπτογραφούμε το μήνυμα με AES-CTR, με το ως άνω κλειδί:
openssl enc -aes-128-ctr -kfile key.bin -iv 00000000000000000000000000000000 –in name.txt -out encrypted_name.bin -nopad -nosalt
Βρίσκουμε το αποτύπωμα του μηνύματός μας, με SHA-256:
openssl dgst -sha256 name.txt>hashed.bin
Κρυπτογραφούμε το αποτύπωμα (αρχείο hashed_bin), με το ίδιο AES κλειδί:

openssl enc -aes-128-ctr -kfile key.bin -iv 00000000000000000000000000000000 –in hashed.bin -out encrypted_hashed.bin -nopad –nosalt

Β) Το AES συμμετρικό κλειδί (αρχείο key.bin) πρέπει να κρυπτογραφηθεί με το δημόσιο RSA κλειδί του διδάσκοντα (publickey-tutor.pem): με αυτόν τον τρόπο, μόνο ο διδάσκων θα μπορεί να το αποκρυπτογραφήσει. Η εντολή είναι:

openssl pkeyutl -encrypt -in key.bin -pubin -inkey publickey-tutor.pem -out

encrypted_key.bin

Ο διδάσκων πρέπει να λάβει το κρυπτογραφημένο συμμετρικό κλειδί encrypted_key.bin (αυτό που θα προκύψει από τα ανωτέρω εντολή), το κρυπτογραφημένο μήνυμα (αρχείο encrypted_name.bin) και το κρυπτογραφημένο αποτύπωμα (αρχείο encrypted_hashed.bin). Πρώτα θα αποκρυπτογραφήσει το συμμετρικό κλειδί με το δικό του ιδιωτικό RSA κλειδί, που το γνωρίζει μόνο αυτός (αρχείο privatekey-tutor.pem):

openssl pkeyutl -decrypt -in encrypted_key.bin -inkey privatekey-tutor.pem -out

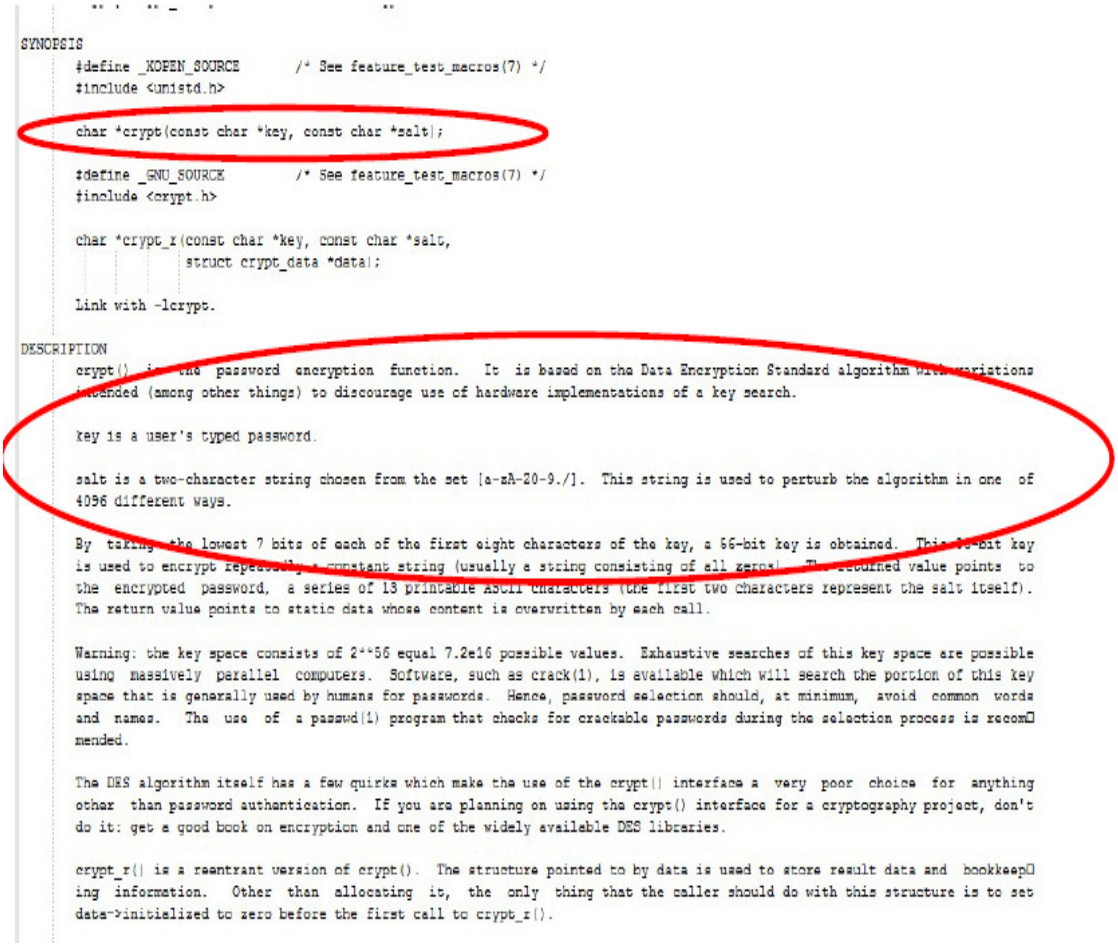
decrypted_key.bin

Με αυτόν τον τρόπο ανακτά το συμμετρικό AES κλειδί (decrypted_key.bin). Πλέον μπορεί να αποκρυπτογραφήσει το κρυπτογραφημένο μήνυμα, να αποκρυπτογραφήσει το κρυπτογραφημένο αποτύπωμα και να ελέγξει αν το αποτύπωμα του μηνύματος που υπολόγισε ταυτίζεται με το αποτύπωμα που έλαβε.

13.5 ΑΣΚΗΣΗ 5

Με την εντολή mancrypt βλέπουμε ότι η crypt καλείται με δύο ορίσματα, το κλειδί

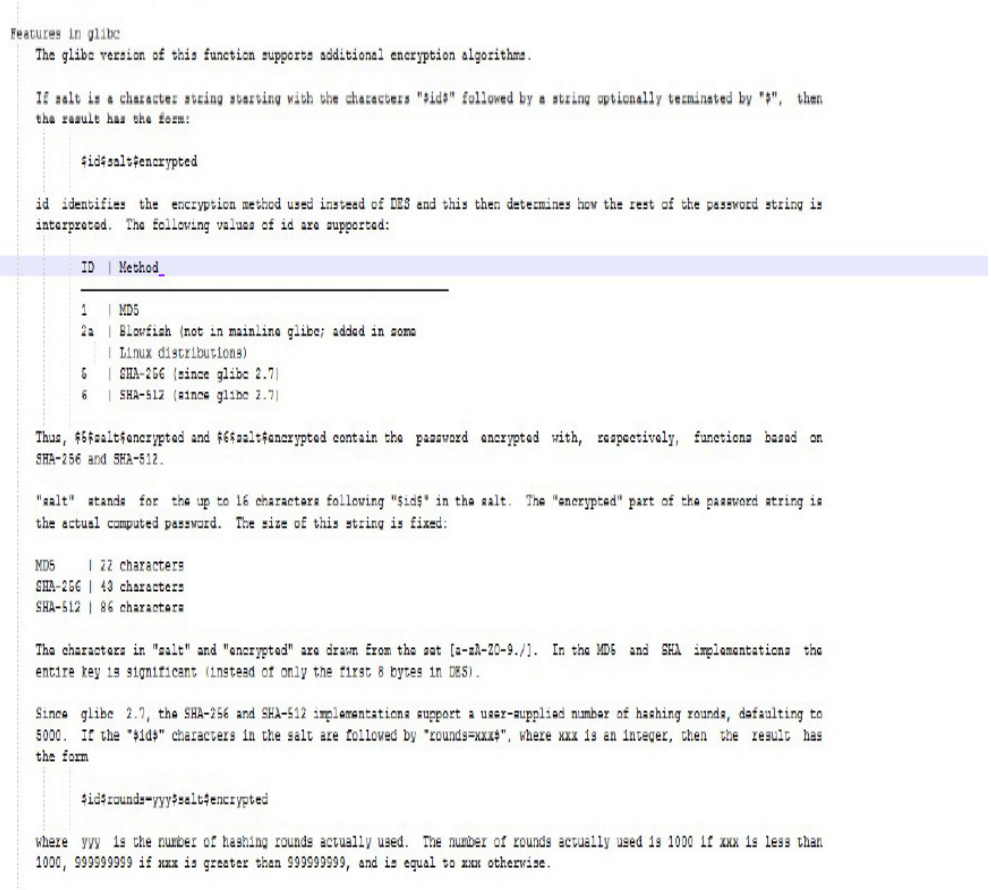
(συνθηματικό) και το salt: (ακολουθεί ενδεικτικό δείγμα οθόνης) :



```
SYNOPSIS
       #define _XOPEN_SOURCE       /* See feature_test_macros(7) */
       #include <unistd.h>

       char *crypt(const char *key, const char *salt);

       #define _GNU_SOURCE         /* See feature_test_macros(7) */
       #include <crypt.h>

       char *crypt_r(const char *key, const char *salt,
                     struct crypt_data *data);

       Link with -lcrypt.

DESCRIPTION
       crypt()  is  the  password  encryption  function.   It  is based on the Data Encryption Standard algorithm with variations
       intended (among other things) to discourage use of hardware implementations of a key search.

       key is a user's typed password.

       salt is a two-character string chosen from the set [a-zA-Z0-9./].  This string is used to perturb the algorithm in one  of
       4096 different ways.

       By  taking  the lowest 7 bits of each of the first eight characters of the key, a 56-bit key is obtained.  This 56-bit key
       is used to encrypt repeatedly a constant string (usually a string consisting of all zeros).  The returned value points  to
       the  encrypted  password,  a series of 13 printable ASCII characters (the first two characters represent the salt itself).
       The return value points to static data whose content is overwritten by each call.

       Warning: the key space consists of 2**56 equal 7.2e16 possible values.  Exhaustive searches of this key space are possible
       using  massively  parallel  computers.  Software, such as crack(1), is available which will search the portion of this key
       space that is generally used by humans for passwords.  Hence, password selection should, at minimum,  avoid  common  words
       and  names.   The  use  of  a passwd(1) program that checks for crackable passwords during the selection process is recom-
       mended.

       The DES algorithm itself has a few quirks which make the use of the crypt() interface a  very  poor  choice  for  anything
       other  than password authentication.  If you are planning on using the crypt() interface for a cryptography project, don't
       do it: get a good book on encryption and one of the widely available DES libraries.

       crypt_r() is a reentrant version of crypt().  The structure pointed to by data is used to store result data and  bookkeep-
       ing  information.   Other  than  allocating  it,  the  only  thing that the caller should do with this structure is to set
       data->initialized to zero before the first call to crypt_r().
```

Με τη συνάρτηση αυτή «προστεύεται» η τήρηση των συνθηματικών – το salt χρησιμοποιείται για να διαφέρει η έξοδος της crypt ακόμα και αν το συνθηματικό είναι το ίδιο, ανάλογα με την τιμή του salt.

Με την ίδια εντολή βλέπουμε επίσης τις εξής πληροφορίες:



```
Features in glibc
   The glibc version of this function supports additional encryption algorithms.

   If salt is a character string starting with the characters "$id$" followed by a string optionally terminated by "$", then
   the result has the form:

         $id$salt$encrypted

   id identifies the encryption method used instead of DES and this then determines how the rest of the password string is
   interpreted. The following values of id are supported:

         ID  | Method
         ---------------------------------------------------------
         1   | MD5
         2a  | Blowfish (not in mainline glibc; added in some
             | Linux distributions)
         5   | SHA-256 (since glibc 2.7)
         6   | SHA-512 (since glibc 2.7)

   Thus, $5$salt$encrypted and $6$salt$encrypted contain the password encrypted with, respectively, functions based on
   SHA-256 and SHA-512.

   "salt" stands for the up to 16 characters following "$id$" in the salt. The "encrypted" part of the password string is
   the actual computed password. The size of this string is fixed:

   MD5     | 22 characters
   SHA-256 | 43 characters
   SHA-512 | 86 characters

   The characters in "salt" and "encrypted" are drawn from the set [a-zA-Z0-9./]. In the MD5 and SHA implementations the
   entire key is significant (instead of only the first 8 bytes in DES).

   Since glibc 2.7, the SHA-256 and SHA-512 implementations support a user-supplied number of hashing rounds, defaulting to
   5000. If the "$id$" characters in the salt are followed by "rounds=xxx$", where xxx is an integer, then the result has
   the form

         $id$rounds=yyy$salt$encrypted

   where yyy is the number of hashing rounds actually used. The number of rounds actually used is 1000 if xxx is less than
   1000, 999999999 if xxx is greater than 999999999, and is equal to xxx otherwise.
```

Το οποίο είναι και αυτό που περιγράφει το shadow αρχείο που διαθέτουμε. Βάσειαυτού, καταλαβαίνουμε ότι αν το salt ξεκινά με $*$ και τελειώνει με $, η τιμή $*$ υποδηλώνει ποια hashfunction έχει χρησιμοποιηθεί για το ψηφιακό αποτύπωμα. Για παράδειγμα, για 6 έχει χρησιμοποιηθεί ο SHA-512. Οι χαρακτήρες που ακολουθούν το salt (μετά το $ στο τέλος), είναι το αποτύπωμα του συνθηματικού, για το συγκεκριμένο salt.

13.6 ΑΣΚΗΣΗ 6

Α) Από το αρχείο shadow βλέπουμε ότι:

Ο bill έχει salt 6AWesFk/4$ (ουσιαστικά, είναι οι χαρακτήρες AWesFk/4, ενώ χρησιμοποιήθηκε η συνάρτηση κατακερματισμού SHA-

512) και αποτύπωμα το **9XlCyEIN0D2Mk2FXaBALk7A81aZR0D1XpGOqlObkQCwp7VY2DwLFYOhpY 3Yu7XnVmupDviJwp3ssi7xqaGPA/**

Η helen έχει salt το 6QNy/phwS$ και αποτύπωμα το **Le4wVYuK8PiBOBijX975KGa/dE3YpfdTPceV2KH2lSHYetCAYGct8q1P.62pO8UPbkFbLIY2Br9S1TSmVmucd0**

Αν π.χ θέλουμε να δούμε αν το συνθηματικό του bill είναι το lefkosia, θα εκτελέσουμε το εξής πρόγραμμα:

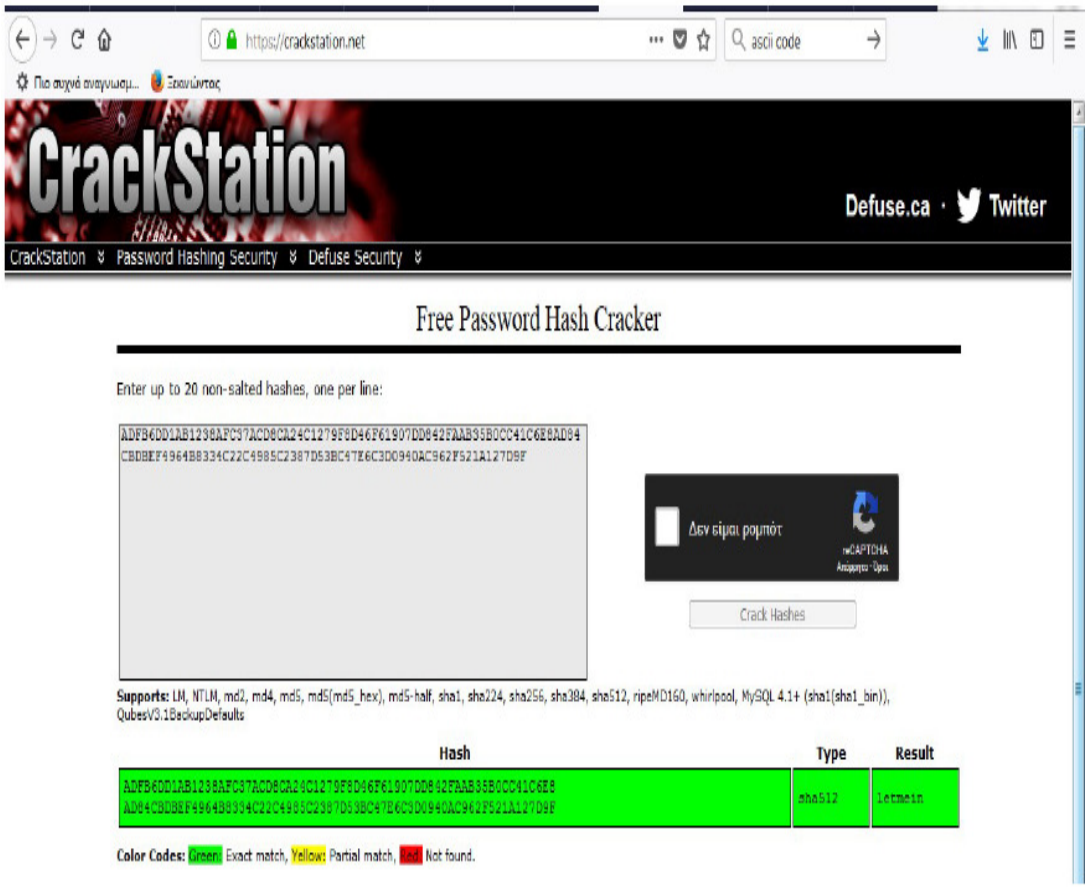
```
#include <stdio.h>
#include <crypt.h>

int main()
{
char *hashed;
hashed=crypt("lefkosia","$6$AWesFk/4$ ");
printf("%s\n",hashed);
return 0;
}
```

Όπου lefkosia το πιθανό συνθηματικό που επιλέγουμε ως δοκιμή «μαντέματος» και 6AWesFk/4$ το salt του bill. Το αποτέλεσμα θα μας δώσει την έξοδο της crypt για αυτά τα δύο ορίσματα (θα τυπωθεί η τιμή της μεταβλητής hashed, την οποία έχουμε ορίσει για να λαμβάνει την έξοδο της crypt): αν μαντέψαμε σωστά, η έξοδος της crypt θα είναι η καταχώρηση για τον tom στο αρχείο shadow. Αλλιώς, επιχειρούμε άλλη δοκιμή. (Όπως θα παρατηρήσουμε, η crypt «επιστρέφει» όχι μόνο το αποτύπωμα αλλά και το salt – όπως ακριβώς είναι η καταχώρηση στο shadow αρχείο).

Αντίστοιχα εργαζόμαστε και για την helen: απλά προσοχή στο ότι για τον κάθε χρήστη το δεύτερο όρισμα της crypt πρέπει να είναι το salt του

εκάστοτε χρήστη. Με αυτόν τον τρόπο, μπορούμε να βρούμε ότι το συνθηματικό του bill είναι το letmein (είναι στη λίστα των πιο συχνών συνθηματικών), και για τη helen είναι το zka5231.

Β) Υπολογίζουμε τα αποτυπώματα με SHA-512 για τα συνθηματικά των bill και helen. Για το συνθηματικό του bill, χρησιμοποιώντας, π.χ., το διαδικτυακό τόπο
https://www.pelock.com/products/hash-calculator, βρίσκουμε το εξής SHA-512
αποτύπωμα:
ADFB6DD1AB1238AFC37ACD8CA24C1279F8D46F61907DD842FAAB35B0CC41C6E8AD84CBDBEF4964B8334C22C4985C2387D53BC47E6C3D0940AC962F521A127D9F
Θέτουμε το αποτύπωμα αυτό στο εργαλείο https://crackstation.net/:
Μας βρίσκει πράγματι το συνθηματικό:



Πράττουμε ακριβώς αντίστοιχα για το συνθηματικό της helen – στην περίπτωση αυτή

δεν το βρίσκουμε. Ο λόγος είναι ότι η λέξη “letmein” ήταν πράγματι πολύ πιθανό να

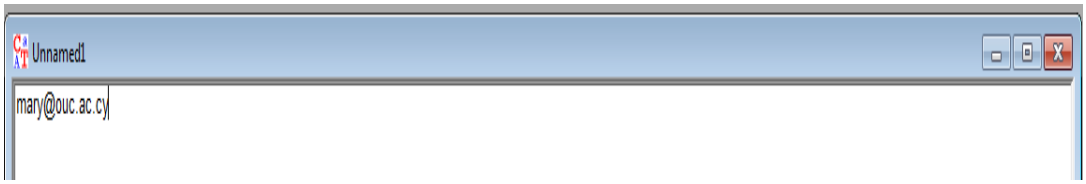
βρίσκεται σε έναν rainbow πίνακα, ενώ για τη λέξη “zka5231” δεν ήταν τόσο πιθανό

(κάτι που επιβεβαιώθηκε).

13.7 ΑΣΚΗΣΗ 7

Α) Για να βρούμε σε ποιον χρήστη απευθυνεται το μήνυμα θα βρουμε το αποτύπωμα τους με τον αλγόριθμο SHA-512 και θα ελέγξουμε αν ταυτιζεται με το δοθέν B7 70 0B 37.

Για το σκοπο αυτό θα χρησιμοποιήσουμε το Cryptool ως εξής: δημιουργούμε ένα αρχειο με την διευθυνση του πρώτου χρηστη:



Εφαρμόζουμε την ακολουθια εντολών :**Inv.Prosedures➔Hash➔SHA-512** και παιρνουμε: 80 8F 15 A2 0B 8A 8D 1C F6 6E E4 D6 03 E8 18 12 A9 18 C1 92 1F 30 10 5B 8E 8C 0A 36 A4 64 A0 F9 FB FC B4 AA 20 E8 7C 4D 1B 7C BF 8B 36 71 81 83 0F B3 0D 54 C0 30 3F CF 40 22 30 1C E9 3F F1 FC

οπου με κοκκινο χρώμα είναι σημειωμένα τα τέσσερα τελαυταια bytes του αποτυπωματος . Παρατηρούμε ότι αυτά δεν ταυτίζονται με το δοθέν αρα το μήνυμα δεν απευθύνεται στη Mary. Ομοιως εργαζόμαστε για τους υπολοιπους χρήστες

Για τον Tom:

9EBB 38 BCDC 5DDBA2 B7 F2 BC 72 DD 88 1FD9 1DE0 CA 95 FE 61 5FC1 77 98 24 1DF1 D5 F6 4E 2AEDE9 29 92 97 E0 04 2CDFEEC6 85 1C 33 10 73 22 A8 6D 9F 69 3C 51 D0 34 7BB0 EDEC 88 37

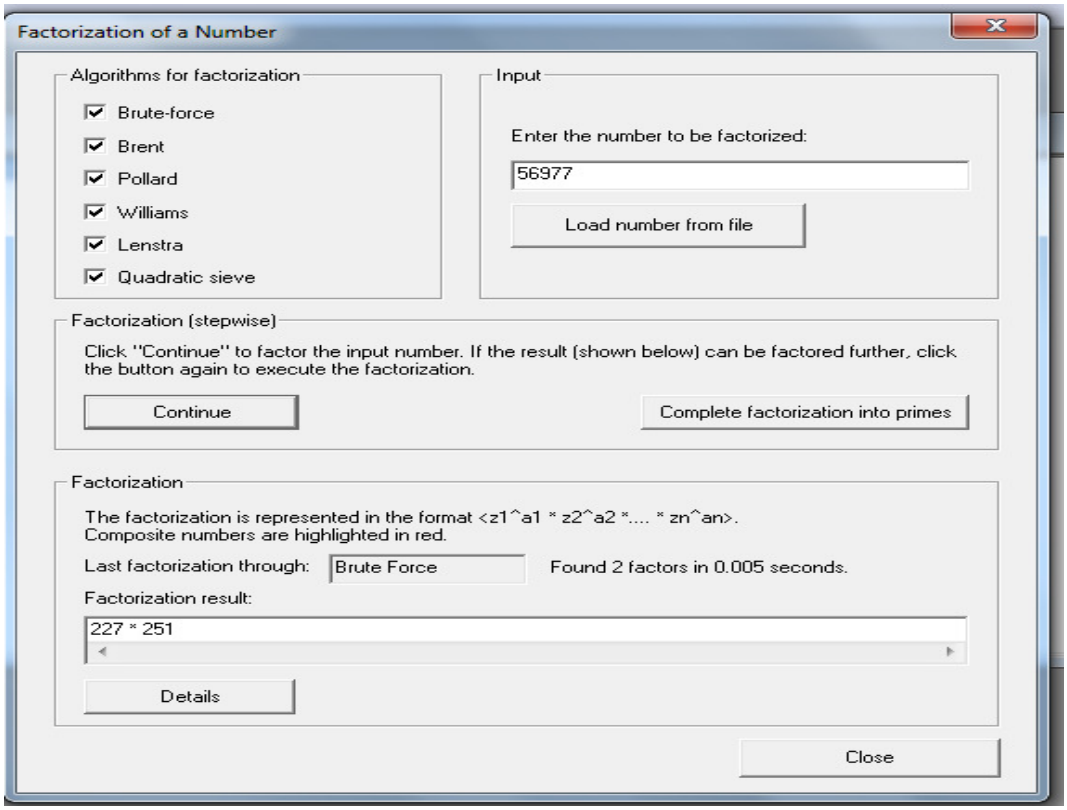
Για τον Bob :

63 35 E4 45 8B 32 AD D9 4C 0C 37 B5 82 74 EC A3 06 6E 92 A1 84 5D 3F F6 D6 D6 6F 03 BA 9D 5D B1 FE 40 E3 3E 0B 4C 11 35 DA B2 53 69 CE 70 27 97 BC BE 23 7D B9 AC 5E 4B 27 20 17 74 B7 70 0B 37

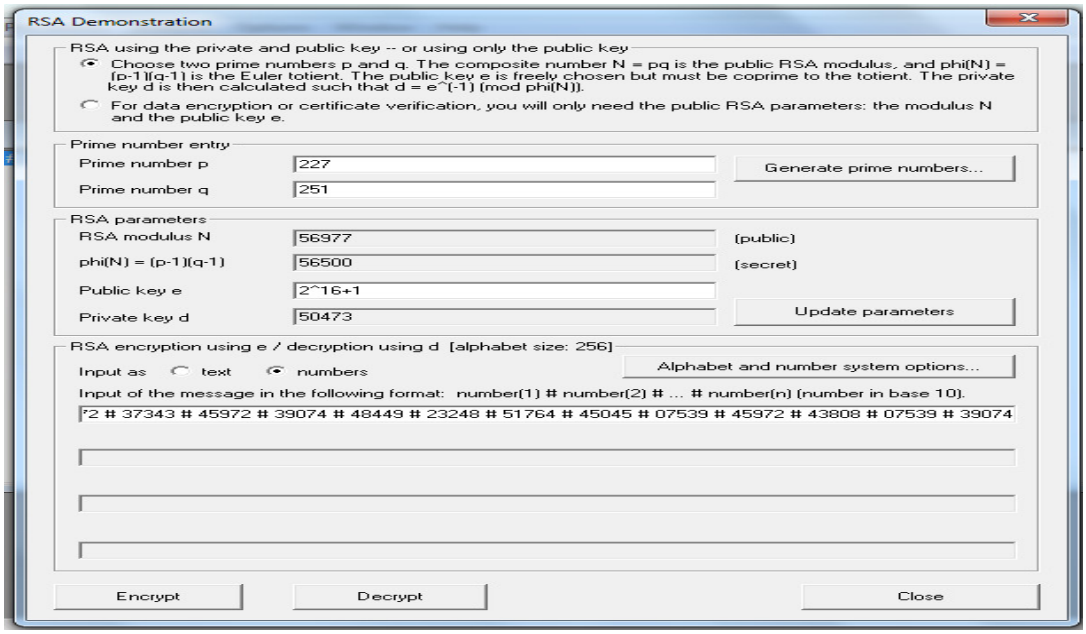
Παρατηρουμε ότι τα τελευταία 4 bytes ταυτίζονται με το ψευδώνυμο άρα βρήκαμε ότι το μήνυμα απευθύνεται στον Bob.

B) Από το προηγούμενο ερώτημα βρήκαμε ότι το μήνυμα απευθύνεται στον Bob του οποίου το κλειδί είναι N=56977 και e=9037

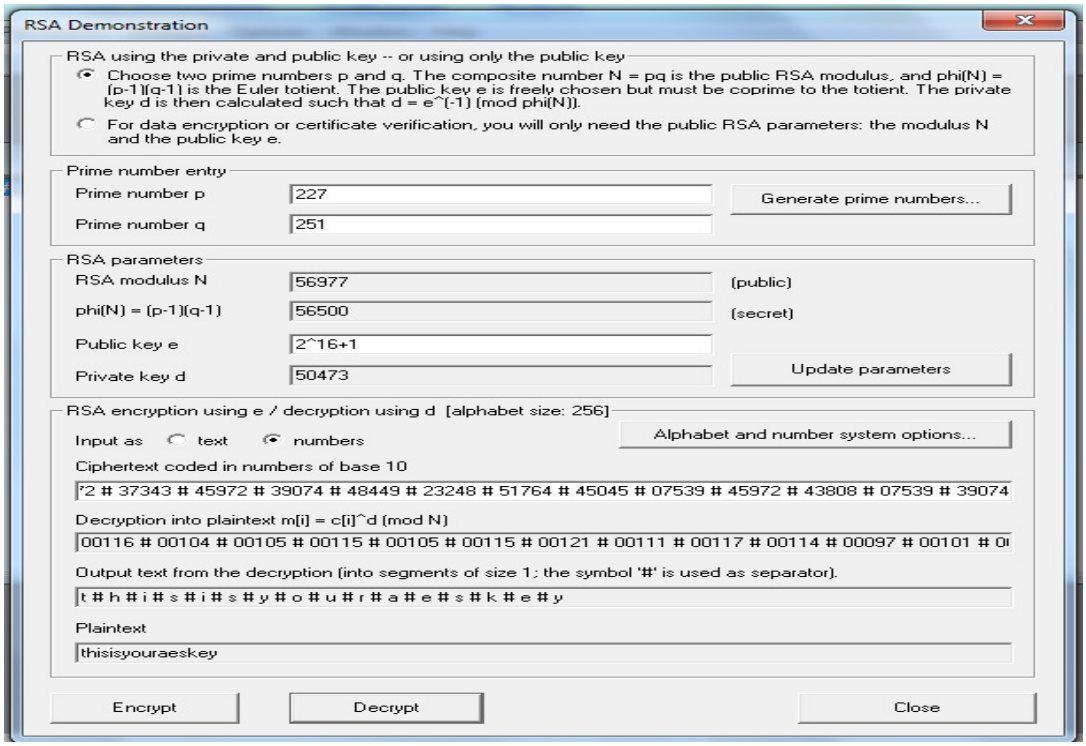
Χρησιμοποιούμε το Cryptool για να παραγοντοποίσουμε το N με την εξής ακολουθία εντολών : Inv. Procedures➔RSA Cryptosystem ➔Factorization of a number



Έτσι βρίσκουμε ότι p=227 και q=251 . Μεταβαίνουμε τώρα μεσω των εντολών Encrypt/Decrypt➔Assymetric➔RSADemonstration στην εξής οθόνη :



Και πατώντας Decrypt παίρνουμε :

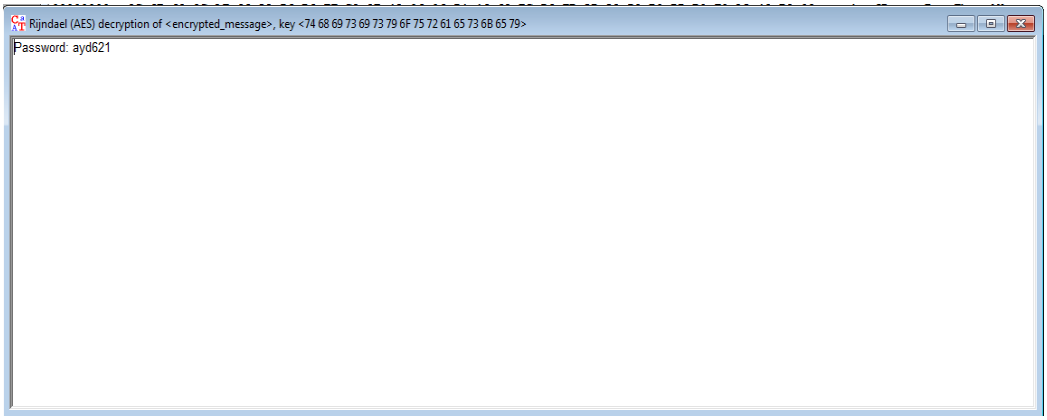


Άρα το κλειδί σε μορφή κειμένου είναι : k=thisisyouraeskey

Με γνωστό πλέον το κλειδί και το κρυπτοκείμενο και τον τρόπο κρυπτογράφησης με την βοήθεια του Cryptool θα αποκρυπτογραφήσουμε το μήνυμα της Alice στον Bob: Θα αποκρυπτογραφήσουμε δηλαδή το encrypted_message.hex με το γνωστό πλέον κλειδί το οποίο σε δεκαεξαδική μορφή είναι :

k=74 68 69 73 69 73 79 6F 75 72 61 65 73 6B 65 79

Η ακολουθία εντολών στο Cryptool για το encrypted_message : Encrypt/Decrypt➔Symmetric(modern)➔AESCBC και παίρνουμε:



Password: ayd621

Γ) Εφόσον γνωρίζουμε ότι η Alice υπολόγισε το αποτύπωμα του κλειδιού με τον αλγόριθμο SHA -512 και χρησιμοποίησε τα τέσσερα τελευταία bytes αυτού για ψευδώνυμο του χρήστη, δηλαδή στην ουσία μπορούμε να βρούμε τα 4*8=32 bits του κλειδιού αν χρησιμοποιούσαμε bruteforceattack θα έπρεπε να ελέγξουμε τους υπόλοιπους 128-32 =96 δυνατούς συνδυασμούς δηλαδή θα έπρεπε να κάνουμε 2^{96} υπολογισμούς.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] G. Brassard. Modern Cryptology: A Tutorial, *Vol. 325 of Lecture Notes in Computer Science*, Springer-Verlag, 1988.

[2] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, Massachusetts, 1990.

[3] J. Hunter. *Αριθμοθεωρία*. Μετάφραση Ν. Κριτικού, ομότιμου καθηγητή ΕΜΠ, 1971, επανέκδοση το 1981.

[4] D. Kahn. *The Codebreakers*. Macmillan Publishing Company, 1976.

[5] N. Koblitz. *Algebraic Aspects of Cryptography*. Springer-Verlag, 1998.

[6] E. Kranakis. *Primality and Cryptography*. Wiley-Teubner Series in Computer Science, 1986.

[7] H.R. Lewis and C.H. Papadimitriou. *Elements of the Theory of Computation, 2nd edition*, Prentice Hall, Upper Saddle River, NJ, 1998. Επίσης, υπάρχει και σε ελληνική μετάφραση από το Τεχνικό Επιμελητήριο της Ελλάδος.

[8] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996. Διαθέσιμο δωρεάν, σε ηλεκρονική μορφή, στην ιστοσελίδα http://www.cacr.math.uwaterloo.ca/hac

[9] I. Niven, H.S. Zuckerman, and H.L. Montgomery. *An Introduction to the Theory of Numbers*. Fifth edition, John Wiley and Sons, 1991.

[10] B. Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. Wiley, 2nd Edition, 1996.

[11] W. Stallings. *Cryptography and Network Security: Principles and Practice*. 3rd edition, Prentice Hall, 2002.