

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ



ΑΣΦΑΛΕΙΑ ΔΙΚΤΥΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΡΟΗ Δ (8^ο ΕΞΑΜΗΝΟ)

(2020 – 2021)

Εργαστηριακή Εργασία

Όνοματεπώνυμο:

- Χρήστος Τσούφης

Αριθμός Μητρώου:

- 03117176

Εξάμηνο:

- 8^ο Εξάμηνο

Ροές:

- Δ, Y, Λ

Στοιχεία Επικοινωνίας:

- el17176@mail.ntua.gr



1. Εισαγωγή

Η παρούσα εργαστηριακή εργασία είναι χωρισμένη σε 5 μέρη, κάθε ένα από τα οποία ασχολείται με διαφορετικό αντικείμενο του μαθήματος, το οποίο έχει καλυφθεί στα πλαίσια των διαλέξεων.

Οι απαντήσεις στις ερωτήσεις καταγράφονται σε αυτό το αρχείο που έχει ονομαστεί χρησιμοποιώντας το επώνυμο και τον Α.Μ. ως εξής: (*SecurityProject2021_PAPADOPOULOS_03117074.docx*). Όταν ολοκληρωθεί η εργασία, ζητείται το ανέβασμα των απαντήσεων στην ιστοσελίδα του μαθήματος. Είναι εφικτό το ανέβασμα πάνω από μία φορά των απαντήσεων, αντικαθιστώντας το παλαιό αρχείο. Όταν γίνει τελική υποβολή εργασίας, κλειδώνεται το κείμενο και δεν γίνεται πλέον τροποποίηση. Το ανέβασμα του αρχείου απαντήσεων είναι προτιμότερο να γίνει σε *.pdf format*.

2. Κλασσικοί Αλγόριθμοι Κρυπτογράφησης

Στο κομμάτι αυτό της εργασίας θα μελετηθούν αντιπροσωπευτικοί κλασσικοί αλγόριθμοι κρυπτογράφησης. Θα χρησιμοποιηθεί το εργαλείο *CrypTool 2.1*, το οποίο αποτελεί πλατφόρμα για κρυπτογράφηση και κρυπτανάλυση. To *CrypTool* διαθέτει εύχρηστα built-in tutorials και documentation και το εγκαθίσταται από το σύνδεσμο <https://www.cryptool.org/en/ct2/downloads>.

Όπου ζητείται screenshot από το *CrypTool*, θα πρέπει να αποθηκεύεται το project με όνομα αρχείου το ονοματεπώνυμο με λατινικούς χαρακτήρες, underscore αύξοντα αριθμό της ερώτησης (π.χ., *PAPADOPoulosnikos_1.cwt*) και έπειτα, print screen + paste στο αρχείο που θα παραδώσετε, ώστε να φαίνεται και το όνομα του project.

Σχετική ύλη: Αλγόριθμοι Αντικατάστασης (substitution): Monoalphabetic Cipher (Caesar Cipher), multiple-letter encryption ciphers (Playfair Cipher, Hill Cipher, Vigenere Cipher, Vernam Cipher), Αλγόριθμοι Αντιμετάθεσης (transposition or permutation): Rail Fence Cipher, Columnar Transposition Ciphers. Κρυπτανάλυση: Κρυπταναλυτική επίθεση.

Ζητείται η σύγκριση αλγορίθμων αντικατάστασης, μονοαλφαβητικών και πολυαλφαβητικών ως προς την ασφάλεια και την ανθεκτικότητα σε τεχνικές κρυπτανάλυσης συχνότητας.

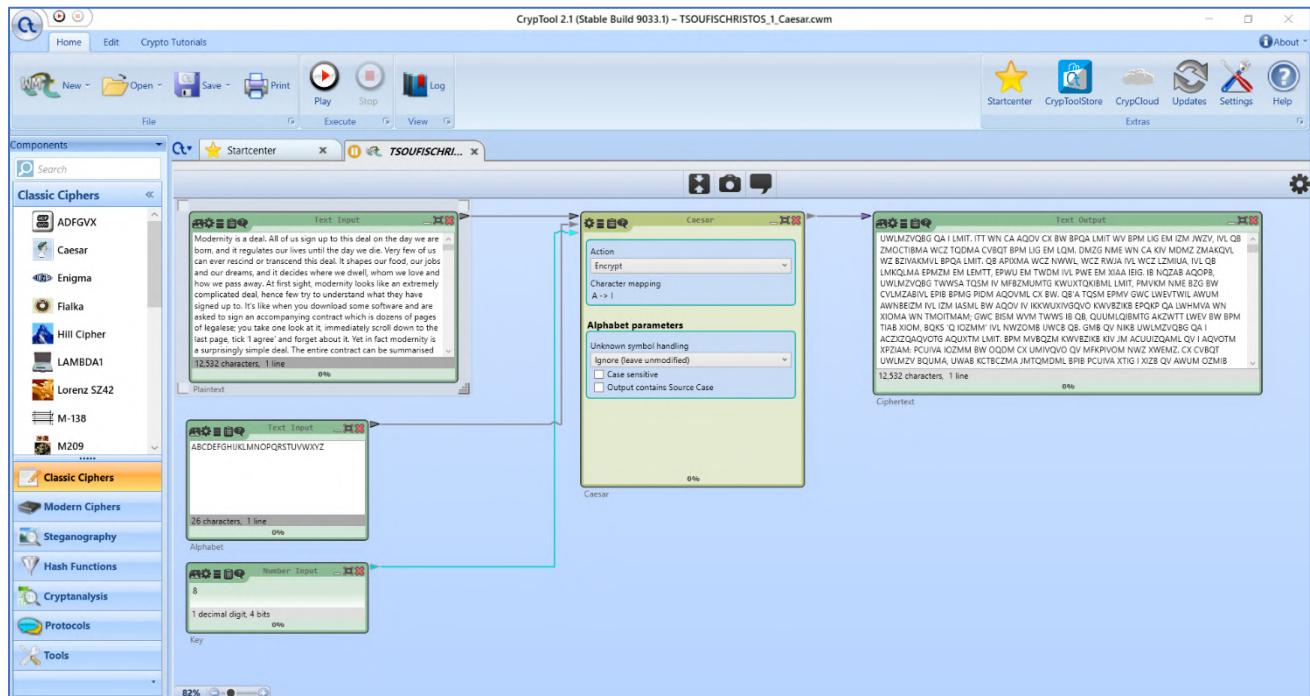
Υπενθυμίζεται ότι θα πρέπει στην εργασία που θα υποβληθεί, να υπάρχουν screenshots από όλα τα projects που χρησιμοποιήθηκαν.

Ζητούνται τα ακόλουθα:

1. Ξεκινώντας από κείμενο αυθαίρετης επιλογής μεγέθους της τάξεως των 2000 λέξεων και κάνοντας χρήση του εργαλείου Cryptool, να πραγματοποιηθεί κρυπτογράφηση βάσει αλγορίθμων αντικατάστασης Caeser, Playfair, Hill, Vigenere και Vernam. Για να επιτευχθεί αυτό, θα γίνει χρήση των έτοιμων αντιστοίχων αλγορίθμων (blocks) του Cryptool καθώς και Text Input για την εισαγωγή των απλού κειμένου και Text Output για να ληφθεί το κρυπτοκείμενο.

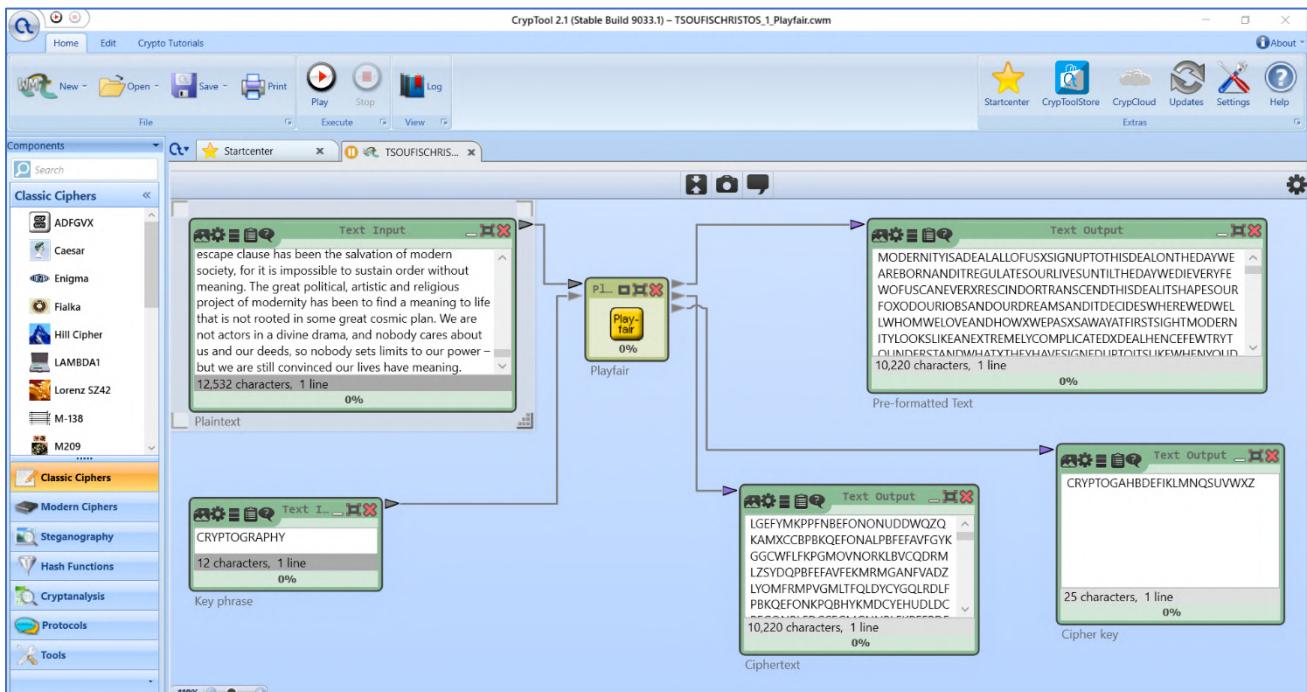
Caesar cipher

Αρχικά, θα γίνει μια σύντομη αναφορά στον τρόπο λειτουργίας του αλγορίθμου αντικατάστασης Caesar. Στην κρυπτογραφία, ο Caesar cipher, γνωστός και ως Caesar's cipher, shift cipher, Caesar's code ή Caesar shift, είναι μια από τις απλούστερες και πιο ευρέως γνωστές τεχνικές κρυπτογράφησης. Είναι ένα είδος Αλγορίθμου Αντικατάστασης όπου κάθε γράμμα σε plaintext αντικαθίσταται από ένα γράμμα που βρίσκεται μερικές θέσεις παρακάτω στο αλφάριθμο ανάλογα με τον fixed αριθμό που δίνεται. Για παράδειγμα, με ένα left shift 3 θέσεων, το D αντικαθίσταται από το A, το E γίνεται Β κ.ο.κ. Η μέθοδος αυτή πήρε το όνομά της από τον Julius Caesar ο οποίος την χρησιμοποίησε στην προσωπική του αλληλογραφία.



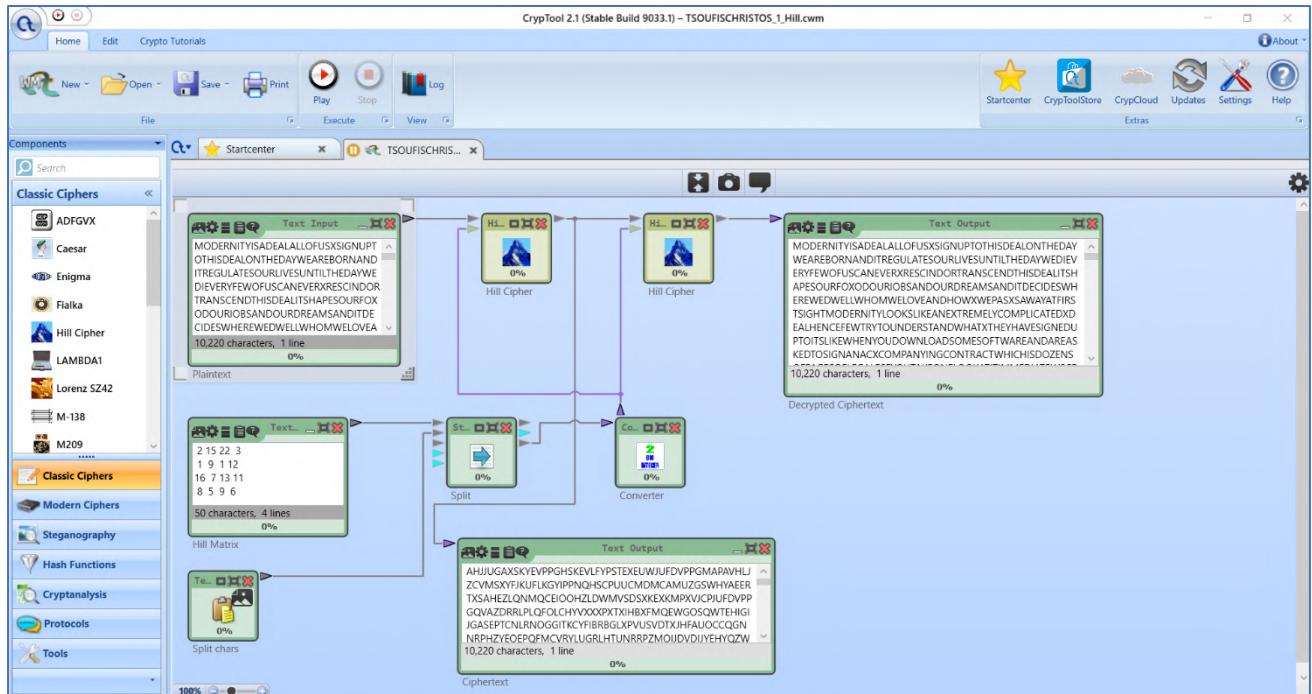
Playfair Cipher

Έπειτα, θα γίνει μια σύντομη αναφορά στον τρόπο λειτουργίας του αλγορίθμου αντικατάστασης Playfair. Ο Playfair cipher ήταν ο πρώτος δίφθογγος (diagraph) αλγόριθμος αντικατάστασης. Εφευρέθηκε το 1854 από τον Charles Wheatstone αλλά πήρε το όνομά του αργότερα από τον Lord Playfair ο οποίος προώθησε την χρήση του cipher. Σε αυτόν τον αλγόριθμο, σε αντίθεση με τους παραδοσιακούς cipher, κρυπτογραφείται ένα ζεύγος αλφαριθμητών (digraphs) αντί για ένα μόνο αλφαριθμητό. Χρησιμοποιήθηκε για στρατηγικούς σκοπούς από τις Βρετανικές δυνάμεις κατά τον 2nd Boer War & World War I και για τον ίδιο σκοπό από τους Αυστραλούς κατά τον World War II. Αυτό συνέβη διότι, είναι σημαντικά γρήγορος στην χρήση και δεν απαιτεί ιδιαίτερο εξοπλισμό. Σύμφωνα με τον αλγόριθμο, το ciphertext και η key phrase που δίνονται, στέλνονται στο Playfair component το οποίο εξάγει το actual key από την key phrase. Αυτό το key έπειτα χρησιμοποιείται για την κρυπτογράφηση το pre-formatted input text.



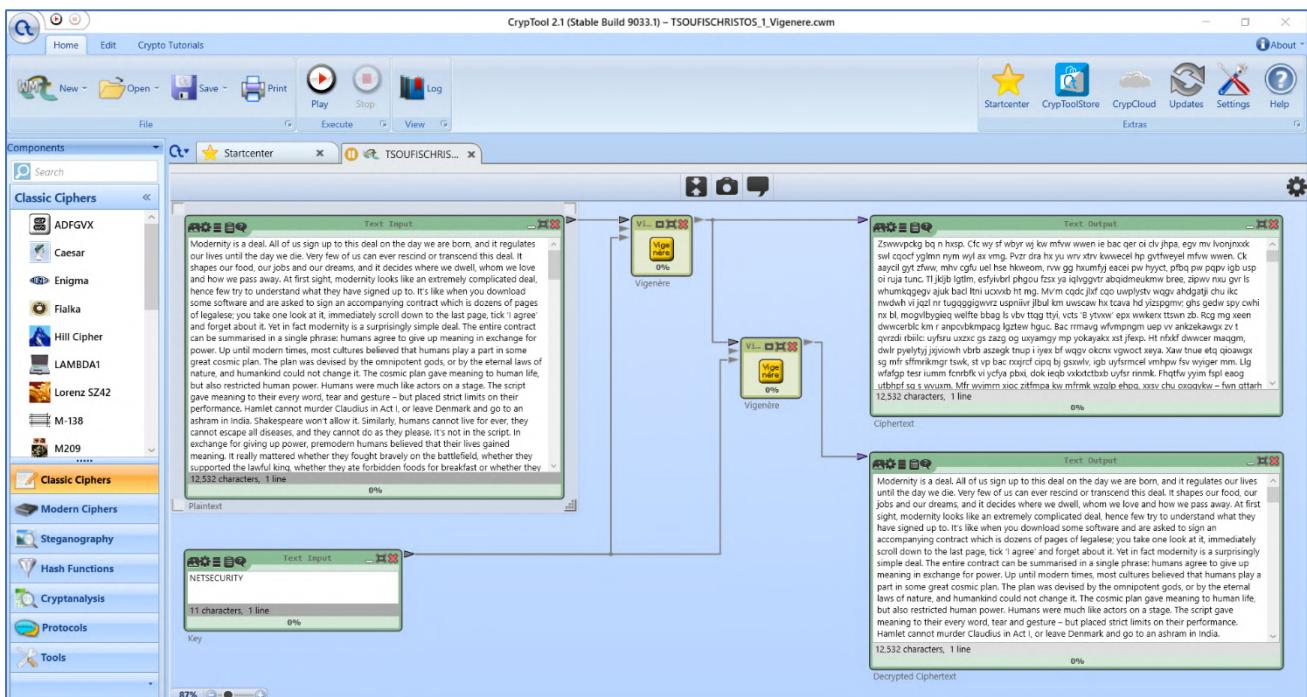
Hill Cipher

Υστερα, θα γίνει μια σύντομη αναφορά στον τρόπο λειτουργίας του αλγορίθμου αντικατάστασης Hill. Ο Hill cipher είναι ένα πολυγραφικός αλγόριθμος αντικατάστασης που βασίζεται στην Γραμμική Αλγεβρα. Κάθε γράμμα εκπροσωπείται από έναν αριθμό modulo 26. Συχνά, χρησιμοποιείται ο απλός συνδυασμός $A = 0, B = 1, \dots, Z = 25$, αλλά αυτό δεν είναι ουσιώδες χαρακτηριστικό του cipher. Για την κρυπτογράφηση ενός μηνύματος, κάθε block από n γράμματα (που θεωρείται ως n -διάστατο διάνυσμα) πολλαπλασιάζεται με ένα αντιστρέψιμο $n \times n$ πίνακα modulus 26. Για την αποκρυπτογράφηση του μηνύματος, κάθε block πολλαπλασιάζεται με τον αντίστροφο του πίνακα που χρησιμοποιείται για την κρυπτογράφηση. Ο πίνακας που χρησιμοποιείται για την κρυπτογράφηση είναι ένα cipher key και πρέπει να επιλέγεται τυχαία από ένα σύνολο αντιστρέψιμων $n \times n$ πινάκων.



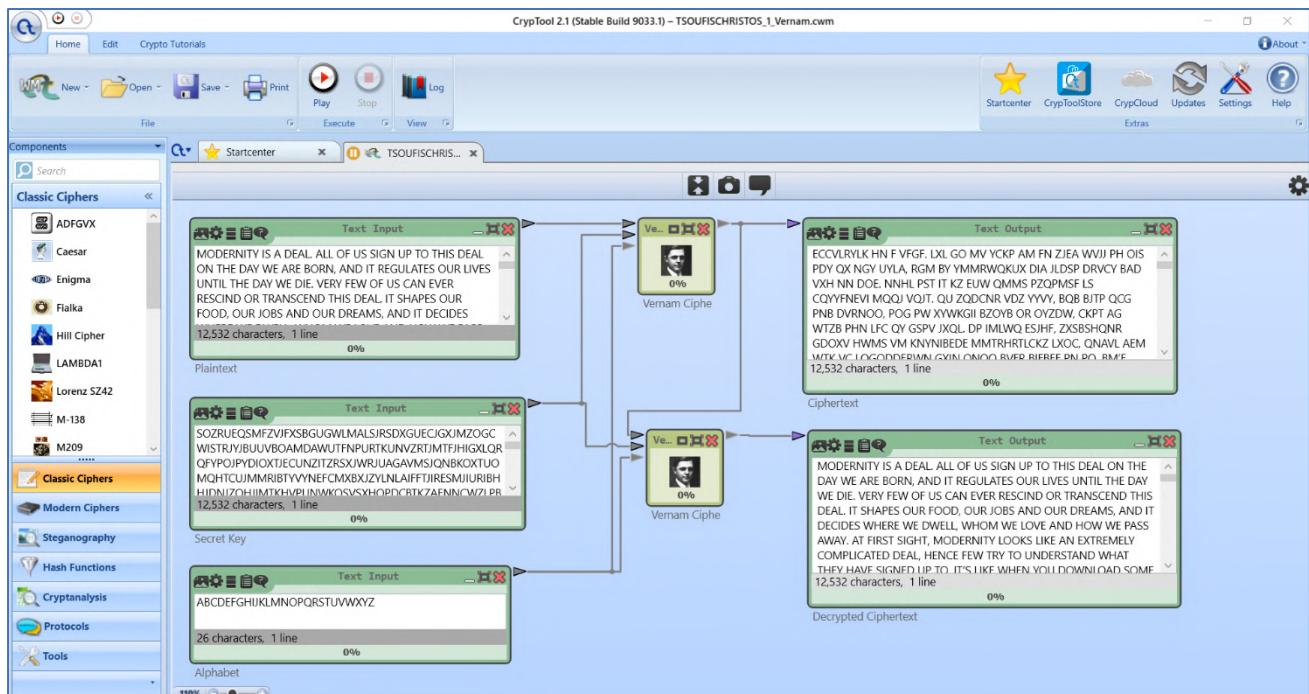
Vigenere Cipher

Ακολουθεί μια σύντομη αναφορά στον τρόπο λειτουργίας του αλγορίθμου αντικατάστασης Vigenere. Ο Vigenere Cipher είναι μια μέθοδος κρυπτογράφησης αλφαβητικού κειμένου. Χρησιμοποιεί μια απλή μορφή πολυαλφαβητικής αντικατάστασης. Ένας πολυαλφαβητικός cipher είναι οποιοσδήποτε cipher που βασίζεται στην αντικατάσταση με χρήση πολλαπλών αλφαβήτων αντικατάστασης. Η κρυπτογράφηση του αρχικού κειμένου γίνεται χρησιμοποιώντας το Vigenere square ή Vigenere table. Το table αποτελείται από αλφάριτα που γράφονται 26 φορές σε διαφορετικές σειρές, και κάθε αλφάριτο μετατοπίζεται κυκλικά προς τα αριστερά σε σχέση με το προηγούμενο, οπότε αντιστοιχεί ουσιαστικά σε 26 πιθανούς Caesar Ciphers. Σε διαφορετικά σημεία κατά την κρυπτογράφηση, ο cipher χρησιμοποιεί διαφορετικό αλφάριτο από την από πάνω σειρά. Τέλος, σημειώνεται ότι το αλφάριτο που χρησιμοποιείται σε κάθε σημείο βασίζεται στο επαναλαμβανόμενο keyword.



Vernam Cipher

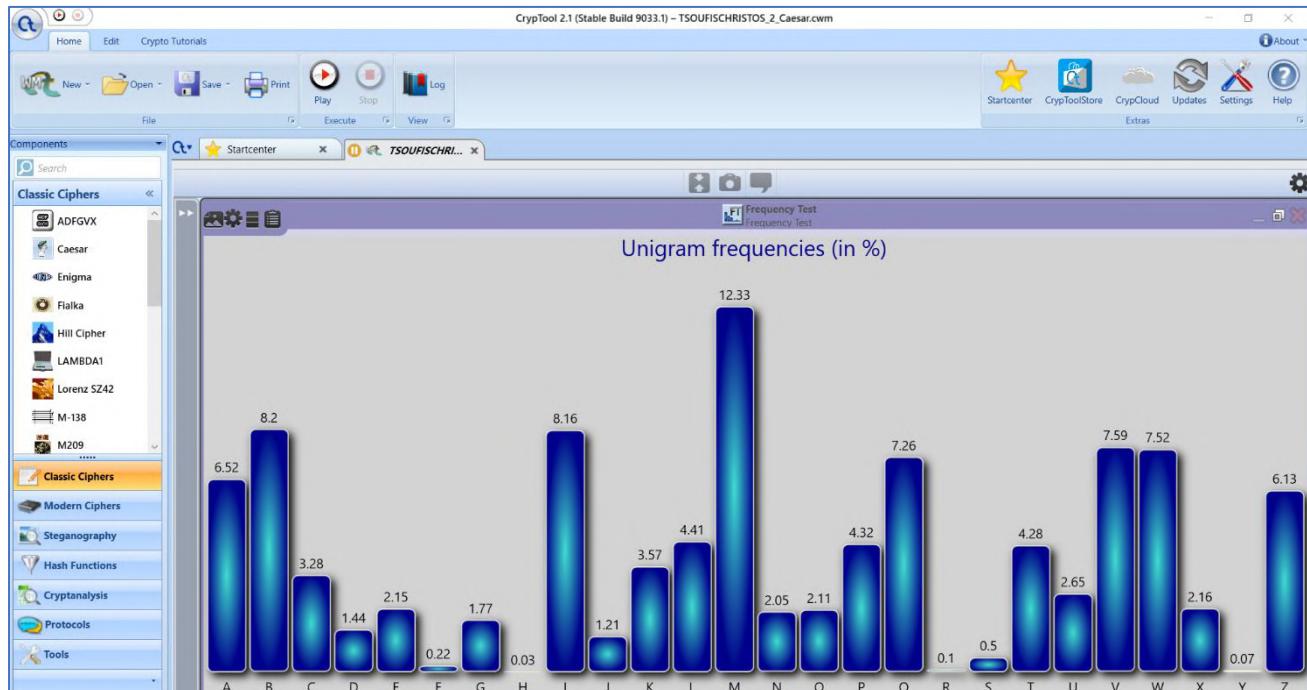
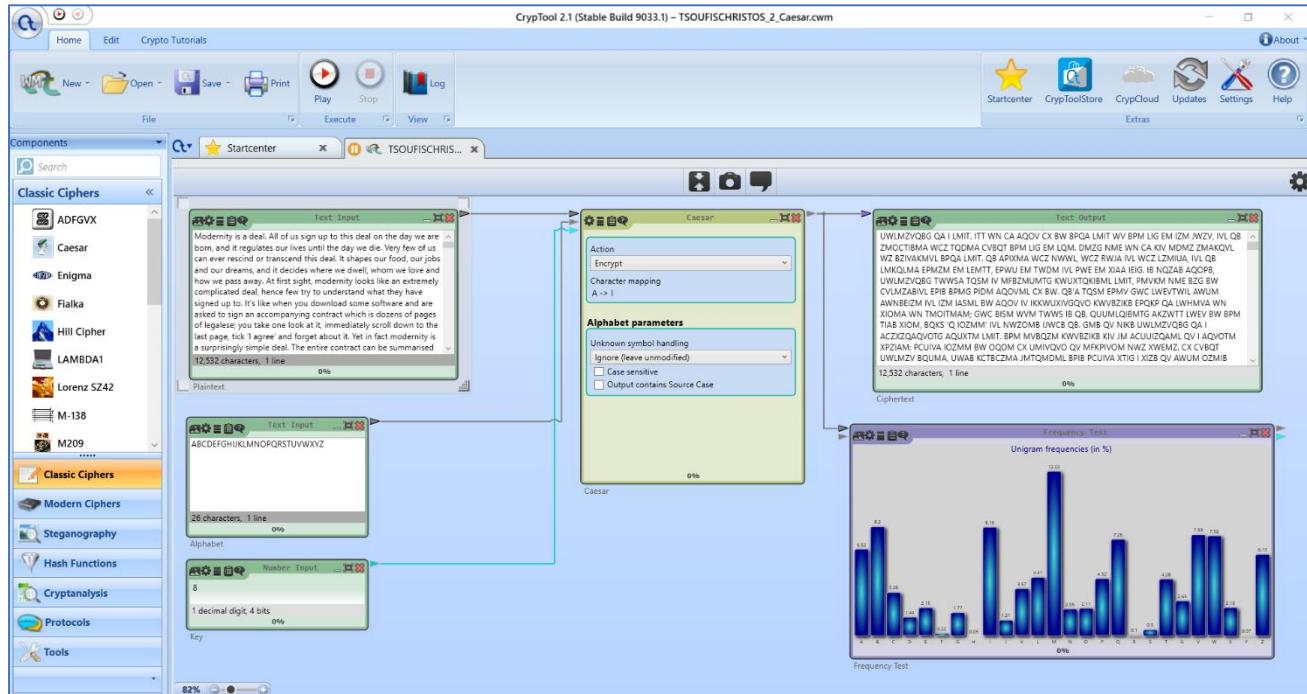
Τέλος, μια σύντομη αναφορά στον τρόπο λειτουργίας του αλγορίθμου αντικατάστασης Vernam. Ο Vernam Cipher είναι μια μέθοδος κρυπτογράφησης αλφαβητικού κειμένου. Αποτελεί μια από τις τεχνικές Transposition για μετατροπή plain text σε cipher text. Σε αυτό το μηχανισμό, ανατίθεται ένας αριθμός σε κάθε χαρακτήρα του plain text (π.χ. a = 0, b = 1, ..., z = 25). Για την κρυπτογράφηση, χρησιμοποιείται ένα κλειδί μήκους ίσου με το μήκος του plain text. Ο αλγόριθμος δουλεύει ως εξής: Αρχικά, ανατίθεται ένας αριθμός σε κάθε χαρακτήρα του plain text και το key με αλφαβητική σειρά. Έπειτα, προστίθεται ο αριθμός (που αντιστοιχεί στον αριθμό του χαρακτήρα του plain text και στον αριθμό του χαρακτήρα του key). Τέλος, αφαιρείται ο αριθμός από το 26 αν ο αριθμός που προστέθηκε είναι μεγαλύτερος από 26, αλλιώς μένει ως έχει.



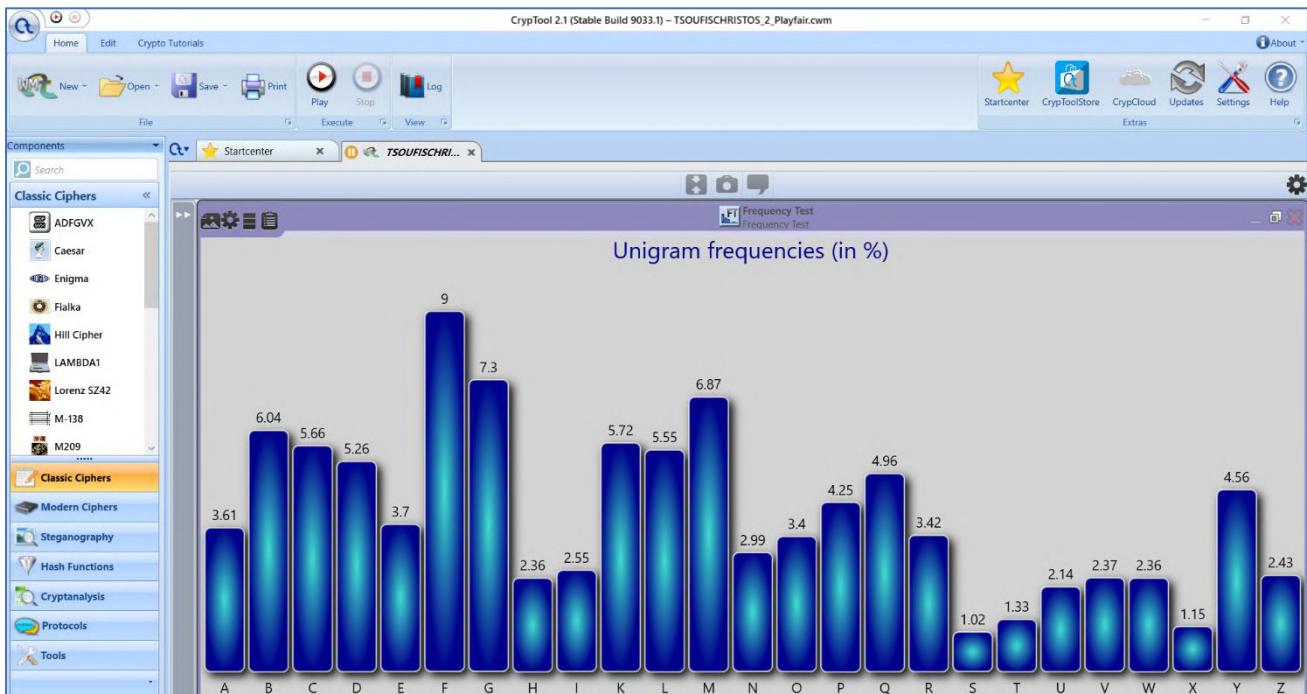
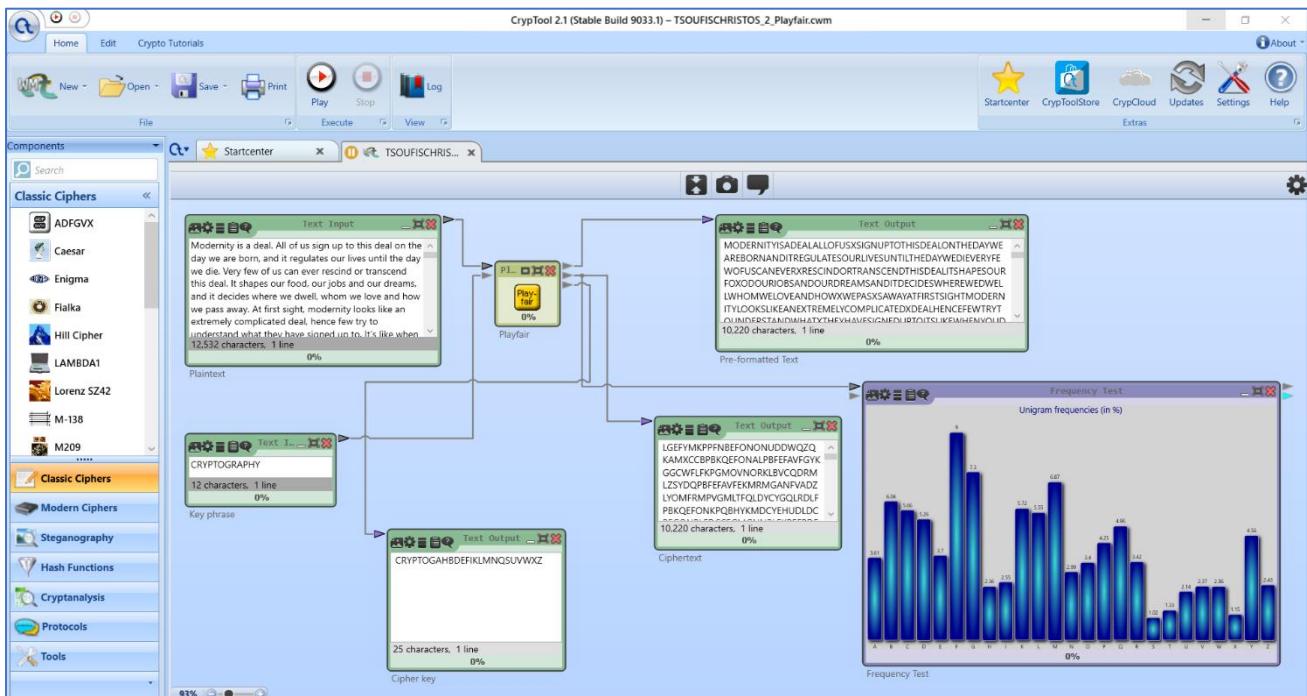
2. Να πραγματοποιηθεί ανάλυση συχνότητας με τη βοήθεια του Cryptool και του Frequency Test block που διαθέτει και να γίνει σχολιασμός με βάση τις συχνότητες την ανθεκτικότητα κάθε αλγορίθμου που δοκιμάστηκαν. Να εντοπιστούν τα αδύναμα σημεία των καθενός και τις παραμέτρους από τις οποίες εξαρτάται η απόδοσή τους.

Με την χρήση του component “Frequency Test” φαίνεται παρακάτω η ανάλυση συχνότητας.

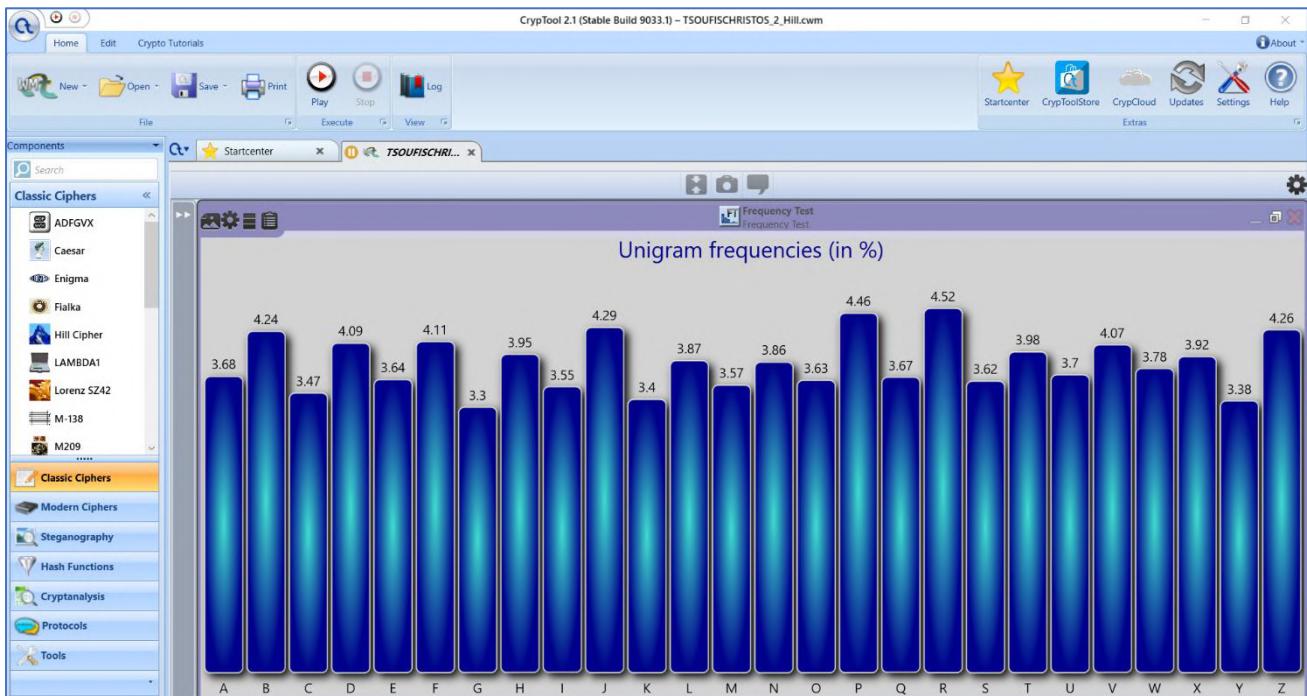
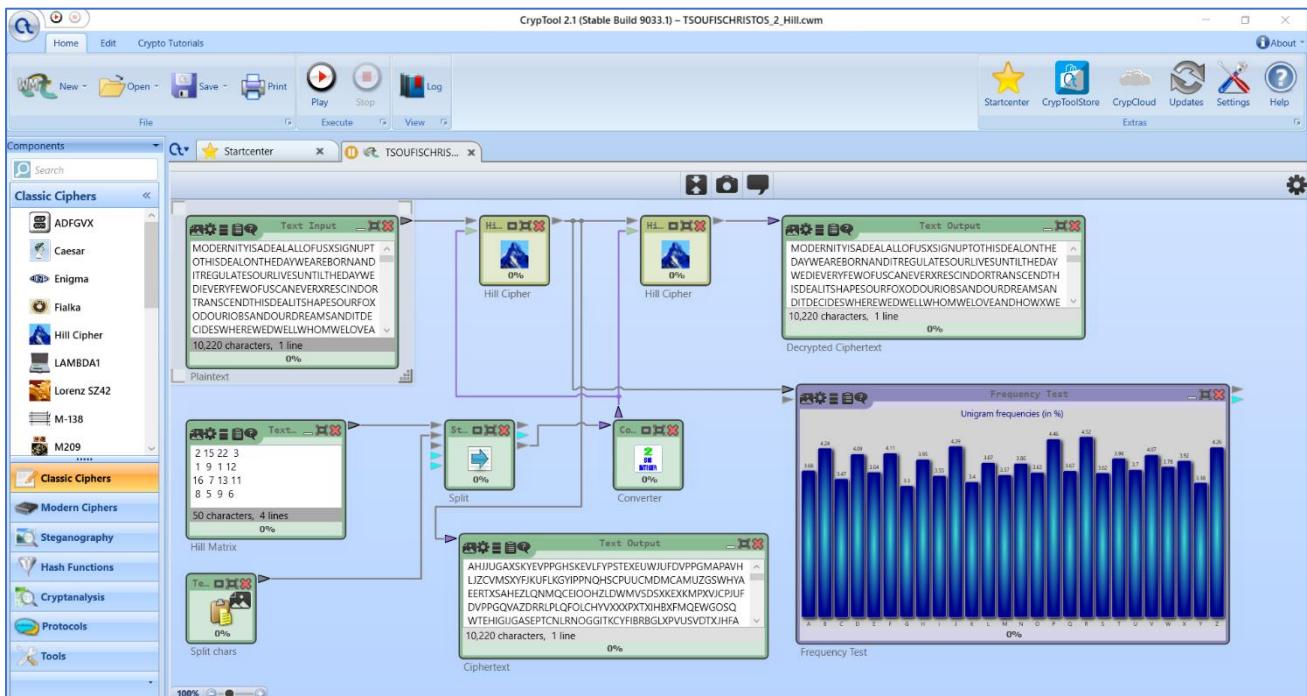
Caesar cipher



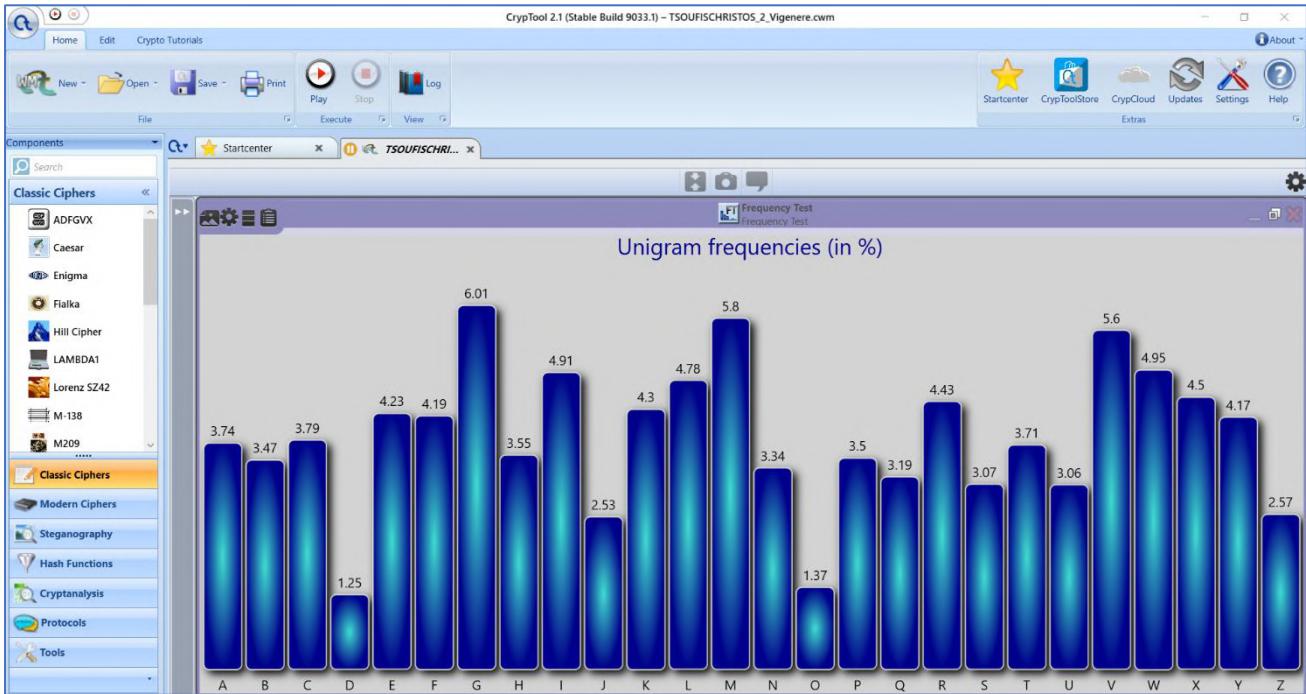
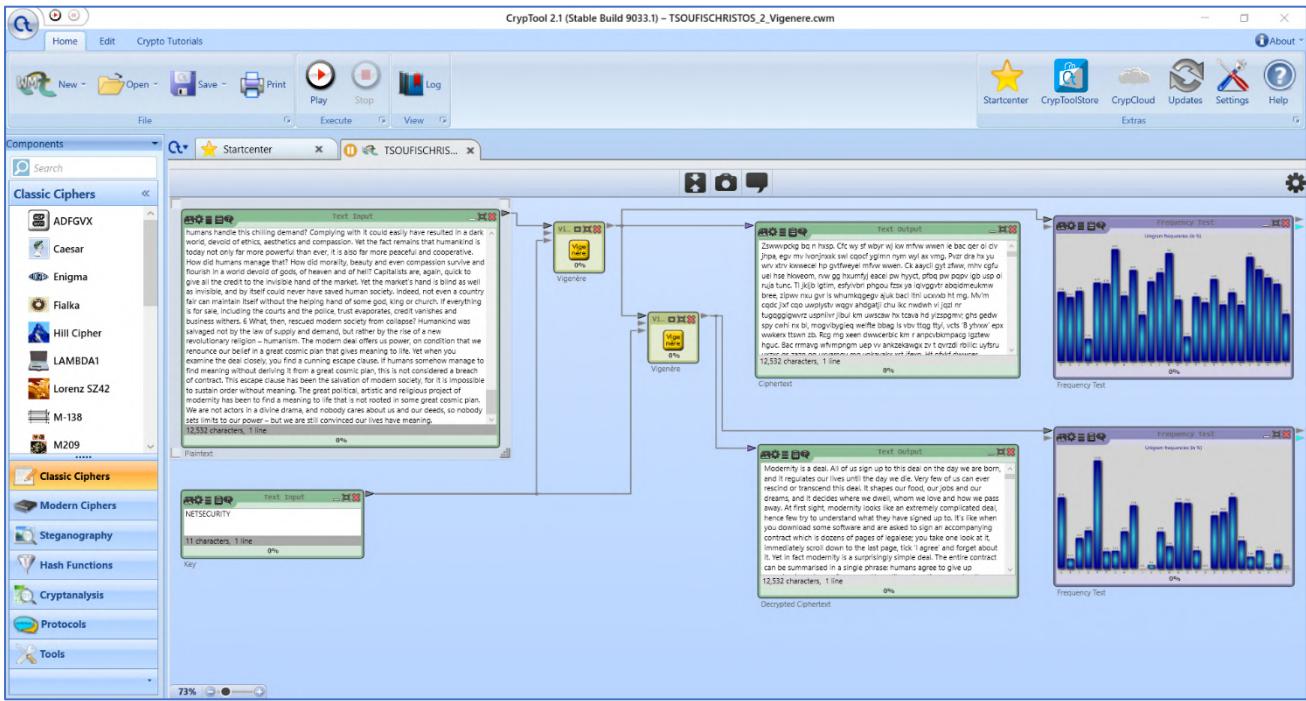
Playfair Cipher



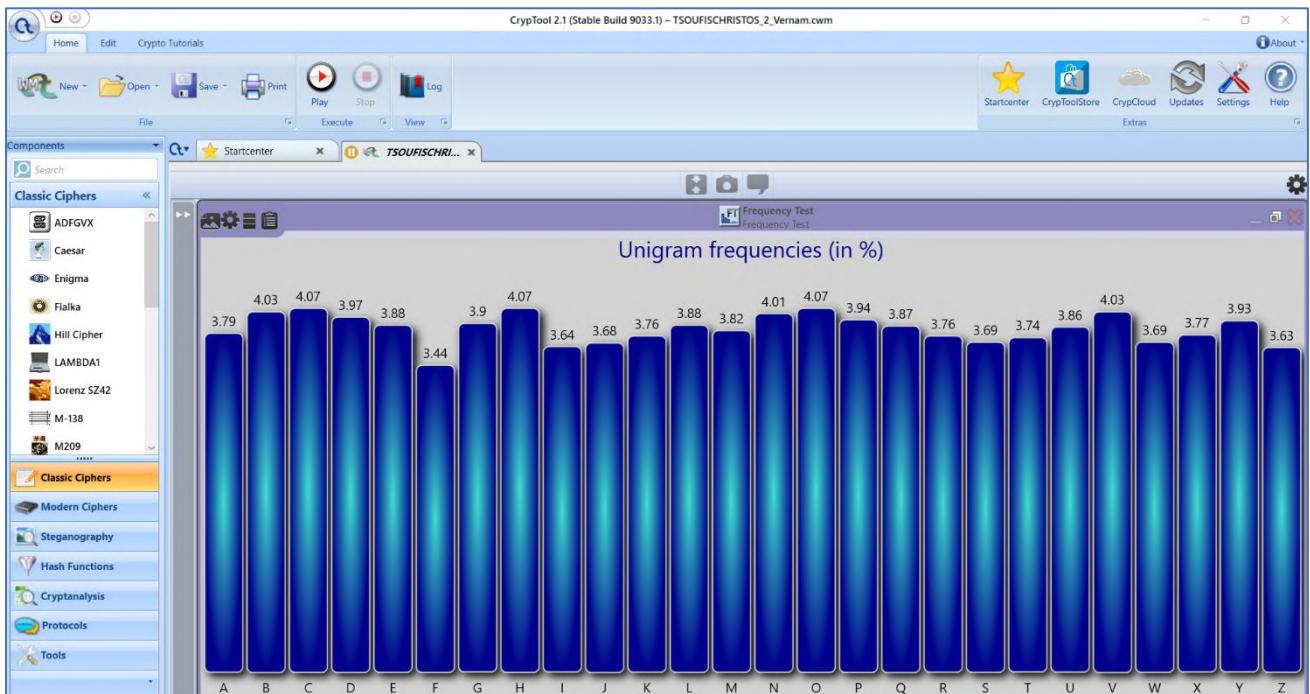
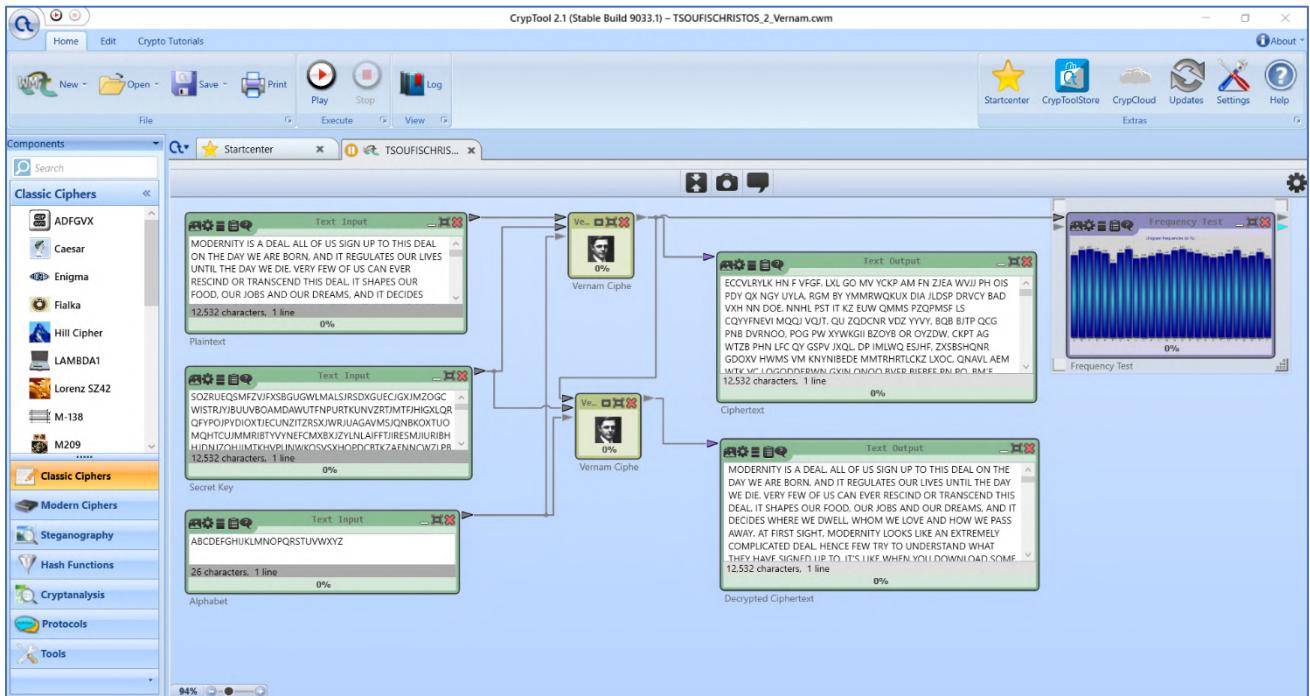
Hill Cipher



Vigenère Cipher



Vernam Cipher



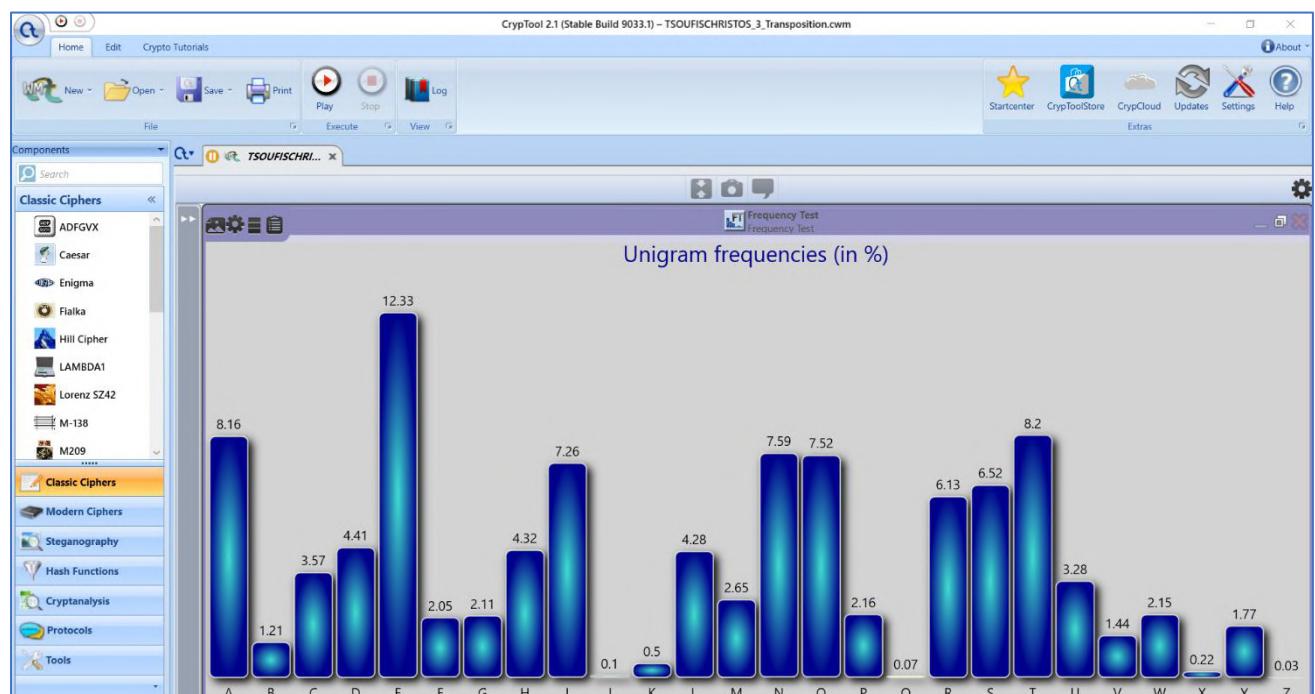
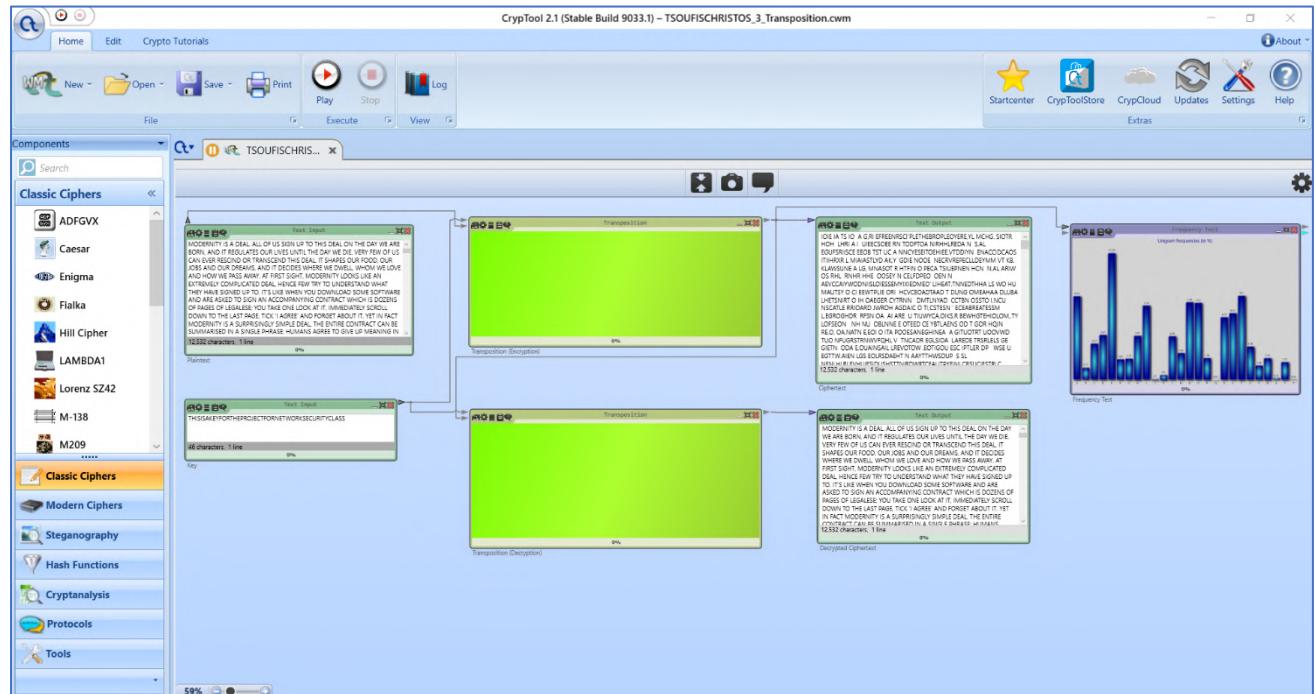
Σχολιασμός & Παρατηρήσεις:

- Ο **Caesar Cipher** παρατηρείται ότι έχει μια πολύ μεγάλη διακύμανση στις συχνότητες των γραμμάτων που χρησιμοποιούνται. Αυτό συνεπάγεται μία μειωμένη ασφάλεια και απόδοση του αλγορίθμου καθώς εύκολα μπορεί κανείς να βρει το κλειδί. Αφενός επειδή δεν μεταβάλλονται οι σχετικές συχνότητες εμφάνισης των γραμμάτων και αφετέρου, γνωρίζοντας τις συχνότητες εμφάνισης γραμμάτων της αγγλικής γλώσσας, μπορεί πολύ εύκολα κανείς να “σπάσει” την κρυπτογράφηση. Είναι γνωστό ότι το 1^o πιο συνηθισμένο γράμμα είναι το E και 2ο το T. Στην ανάλυση συχνότητας του κειμένου φαίνεται πως το 1^o πιο συχνό γράμμα είναι το M και 2ο το B που απέχουν 8 θέσεις από το E και το T αντίστοιχα. Συνεπώς, το shift του αλγορίθμου είναι 8 και προκύπτει από την αντικατάσταση M→E, η οποία είναι πράγματι $13 - 5 = 8$. Επομένως, η αντοχή στην κρυπτανάλυση είναι ελάχιστη και δεν μπορεί να βελτιωθεί. Αυτό επιβεβαιώνεται και από το γεγονός ότι λόγω του μικρού πλήθους των δυνατών κλειδιών που μπορούν να χρησιμοποιηθούν (26 στο λατινικό αλφάβητο) οπότε ο αλγόριθμος “σπάει” και με χειροκίνητη brute force attack.
- Ο **Playfair Cipher** παρατηρείται ότι έχει μεγάλη διακύμανση στις συχνότητες των γραμμάτων, γεγονός που συνεπάγεται ότι έχει χρησιμοποιηθεί κάποιος αλγόριθμος αντικατάστασης. Παρουσιάζει βελτιωμένη ασφάλεια σε σχέση με τον μονοαλφαβητικό αλγόριθμο Caesar, καθώς κωδικοποιεί ζεύγη γραμμάτων. Εδώ, το διάγραμμα συχνοτήτων έχει μικρότερη διακύμανση από αυτό του Caesar, οπότε δεν είναι εύκολο με μια μόνο αντικατάσταση να βρεθεί το key. Όμως, αν γίνει έλεγχος για τη συχνότητα εμφάνισης δυάδων, τότε γίνεται ευκολότερα η αποκρυπτογράφηση του αρχικού κειμένου. Τελικά, και αυτός ο cipher μπορεί να σπάσει σχετικά εύκολα με ικανό μέγεθος κρυπτογραφημένου κειμένου με λίγη παραπάνω υπολογιστική ισχύ, σε σχέση με πριν. Συνεπώς, και εδώ, η αντοχή του στην κρυπτανάλυση είναι μικρή.
- Ο **Hill Cipher** παρατηρείται πως είναι πολύ πιο ασφαλής από τους 2 προηγούμενους ciphers, καθώς κρύβει εντελώς τις συχνότητες εμφάνισης μονών γραμμάτων. Η κατανομή συχνοτήτων είναι αρκετά ομοιόμορφη, οπότε δύσκολα μπορεί να βρεθεί το key. Μάλιστα, όσο μεγαλύτερος είναι ο πίνακας στον Hill, τόσο περισσότερες πληροφορίες κρύβει οπότε τόσο δυσκολότερη είναι η εύρεση του κλειδιού. Επομένως, η αντοχή στην κρυπτανάλυση είναι αυξημένη.
- Ο **Vigenere Cipher** παρατηρείται πως έχει μέτρια διακύμανση στην κατανομή συχνοτήτων. Αντίθετα από τους προηγούμενους ciphers, ο Vigenere χρησιμοποιεί ένα σύνολο μονοαλφαβητικών κανόνων αντικατάστασης για την κρυπτογράφηση με βάση ένα κλειδί, το οποίο με τη σειρά του μπορεί να συνεπάγεται μια πιο εξειδικευμένη κρυπτανάλυση ανά διαστήματα. Μάλιστα, η αποτελεσματικότητα του στηρίζεται στο μέγεθος και την μέθοδο δημιουργίας του κλειδιού, αφού όταν το διάστημα αυτό συμπίπτει με το μήκος κλειδιού, τα αποτελέσματα είναι καλύτερα. Αν δεν επιλεχθούν σωστά τα διαστήματα, τότε αποτελούν αδύναμο σημείο για τον cipher. Εδώ, η αντοχή του cipher στην κρυπτανάλυση είναι μάλλον μέτρια αλλά μπορεί να βελτιωθεί με την αύξηση του κλειδιού. Συνεπώς, προκύπτει ότι η αύξηση της απόδοσης είναι ανάλογη με το μήκοντος κλειδιού (όμως ότι ένα υπερβολικά μεγάλο κλειδί είναι ασύμφορο).
- Ο **Vernam Cipher** παρατηρείται πως έχει κι αυτός μια μέτρια διακύμανση, οπότε και εδώ έχει χρησιμοποιηθεί κάποιος αλγόριθμος αντικατάστασης. Φαίνεται πως η ανάλυση συχνότητας δεν βοηθάει σχεδόν καθόλου στην περίπτωση του Vernam, καθώς οι συχνότητες εμφάνισης των γραμμάτων είναι ομοιόμορφες και σχεδόν ισοκατανεμημένες. Αυτό συμβαίνει διότι γίνεται χρήση ενός κλειδιού με μήκος ίσο με το αρχικό μήνυμα (που δεν έχει καμία στατιστική συσχέτιση με αυτό). Όπως αναφέρθηκε και προηγουμένως, αυτό μειώνει την αντοχή του αλγορίθμου στην κρυπτανάλυση συχνότητας. Ωστόσο, επειδή τα bits που χρησιμοποιούνται παράγονται από γεννήτρια ψευδο-τυχαίων αριθμών, η αντοχή της κρυπτογράφησης αυξάνεται αναλογικά σε σχέση με το μήκος κλειδιού.

3. Να κρυπτογραφηθεί το ίδιο κείμενο κάνοντας χρήση του αλγόριθμου *Transposition*, να πραγματοποιηθεί ανάλυση συχνότητας και να καταγραφούν οι παρατηρήσεις.

Transposition Cipher

Στην κρυπτογραφία, ο Transposition Cipher είναι μια μέθοδος κρυπτογράφησης κατά την οποία οι θέσεις που βρίσκονται οι χαρακτήρες του plaintext γίνονται shift σύμφωνα με ένα regular system έτσι ώστε το ciphertext να είναι μια μετάθεση του plaintext. Με άλλα λόγια, αναδιατάσσει την θέση των χαρακτήρων του plaintext χωρίς όμως να αλλάζει τους ίδιους του χαρακτήρες.

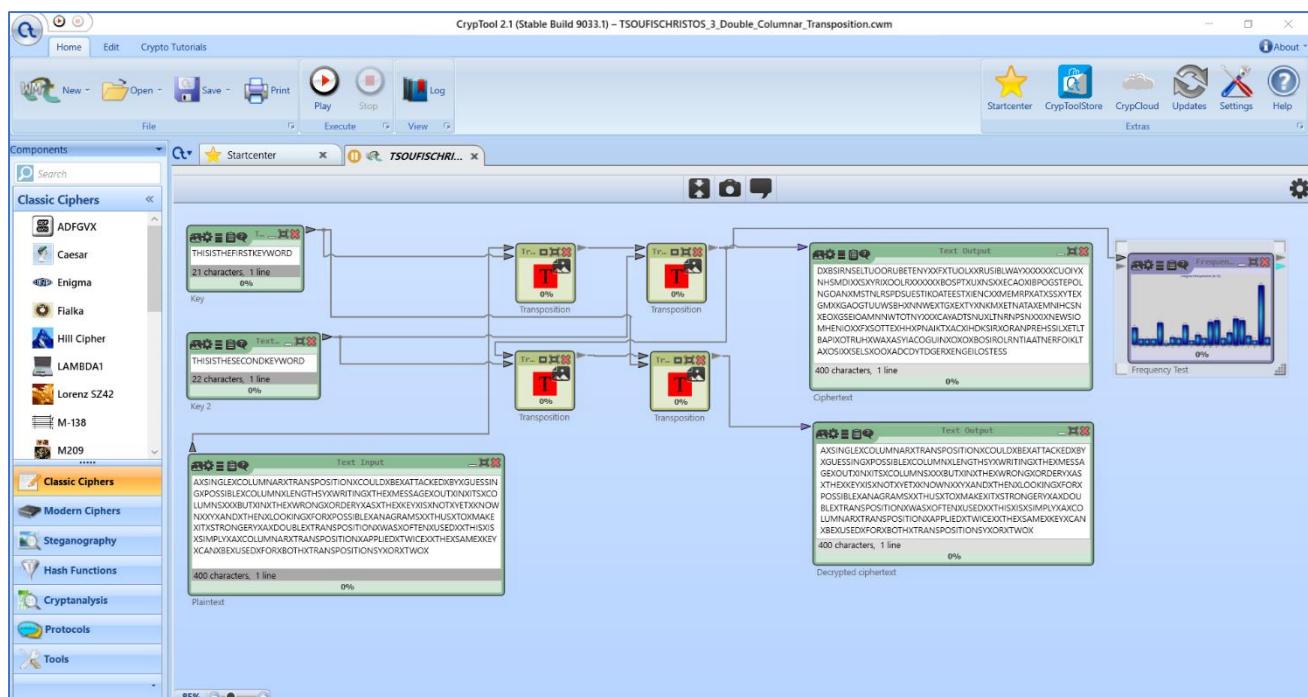


Σχολιασμός & Παρατηρήσεις:

Για τον αλγόριθμο Transposition, παρατηρείται πως δεν έχει αποκρύψει τις συχνότητες εμφάνισης των γραμμάτων. Αυτό δικαιολογείται από το γεγονός ότι είναι αλγόριθμος Αντιμετάθεσης και όχι Αντικατάστασης. Μάλιστα, οι συχνότητες εμφάνισης των γραμμάτων είναι ακριβώς ίδιες με τις αρχικές και από την ανάλυση συχνότητας για το αρχικό και για το κρυπτογραφημένο μήνυμα, προκύπτει ακριβώς το ίδιο διάγραμμα συχνοτήτων. Το αποτέλεσμα αυτό είναι αναμενόμενο αφού ο Transposition cipher δεν προκαλεί καμία αντικατάσταση στους χαρακτήρες που χρησιμοποιούνται, αλλά μόνο τους αντιμεταθέτει. Συνεπώς, ο cipher είναι πλήρως ανθεκτικός στις επιθέσεις κρυπτανάλυσης συχνότητας χαρακτήρων. Επιπροσθέτως, για την αποφυγή επιπλέον brute force attack, αξίζει να σημειωθεί πως η χρήση ενός μεγάλου κλειδιού βελτιώνει αρκετά την αντοχή του.

Double Columnar Transposition

Το ακόλουθο template προστίθεται ενδεικτικά και δείχνει πως μπορεί να προκύψει ένα double columnar transposition cipher από 2 transposition components. Πρώτα, το plaintext κρυπτογραφείται με το 1^o keyword "THISISSTHEFIRSTKEYWORD" με το 1^o Transposition component. Έπειτα, το text κρυπτογραφείται ξανά με το 2^o keyword "THISISSTHESCONDKEYWORD" με το 2^o Transposition component. Τέλος, το ciphertext φαίνεται στο ciphertext Text Output component. Επιπλέον, φαίνεται και η αποκρυπτογράφηση. Πρώτα αποκρυπτογραφείται με το 2^o keyword και μετά αποκρυπτογραφείται με το 1^o keyword. Το αποκρυπτογραφημένο plaintext φαίνεται στο "Decrypted ciphertext" Text Output component.



3. Στεγανογραφία

Στο κομμάτι αυτό της εργασίας θα γίνει χρήση μεθόδων στεγανογραφίας. Θα χρησιμοποιηθεί το εργαλείο *Cryptool 2.1*, το οποίο αποτελεί πλατφόρμα για κρυπτογράφηση και κρυπτανάλυση. Το *CrypTool* διαθέτει εύχρηστα *built-in tutorials* και *documentation*.

Όπως και πριν, όπου ζητείται *screenshot* από το *CrypTool*, θα πρέπει να αποθηκεύεται το *project* με όνομα αρχείου το ονοματεπώνυμο με λατινικούς χαρακτήρες, *underscore* και αύξοντα αριθμό της ερώτησης (π.χ., *PAPADOPoulosnikos_1.cwm*) και ύστερα, *print screen + paste* στο αρχείο που θα παραδώσετε, ώστε να φαίνεται και το όνομα του *project*.

Σχετική ύλη: Στεγανογραφία: Watermarking, LSB.

Ζητούνται τα ακόλουθα:

1. Εφαρμογή στεγανογραφίας με Watermarking. Το *watermarking* είναι μια ειδική τεχνική που εγκαθιστά αόρατα ψηφιακά σημάδια στις εικόνες και στα αρχεία ήχου τα οποία φανερώνουν πληροφορίες, όπως πνευματικά δικαιώματα. Ξεκινώντας από μια αυθαίρετη εικόνα και κάνοντας χρήση του *Cryptool* και ειδικότερα του *Watermark creator* να προστεθεί ένα αόρατο *watermark* – μυστικό μήνυμα – στην εικόνα αυτή, επιλέγοντας κατάλληλα στο πεδίο *modification type*. Παρατηρείται ότι η εικόνα εξόδου του *watermark creator* δεν έχει διαφορά με την αρχική εικόνα. Τρέχοντας πάλι το *project*, το *watermark* θα είναι ορατό. Στη συνέχεια, να τεθεί την εικόνα εξόδου ως είσοδο σε νέο *watermark creator* ώστε να εξαχθεί το αρχικό μυστικό μήνυμα.

Στη συνέχεια να γίνει επεξεργασία της εικόνας με τη χρήση *ImageProcessor* του *CrypTool* επιλέγοντας *action Gray Scale* και να ελεγχθεί αν μπορεί να εξαχθεί το μυστικό μήνυμα (*watermark*) από τη νέα ασπρόμαυρη εικόνα.

Steganography

Στεγανογραφία είναι η πρακτική της απόκρυψης ενός μηνύματος εντός άλλου μηνύματος ή φυσικού αντικειμένου. Προέρχεται από την λέξη “στέγος” που σημαίνει κάλυμμα και “γραφία” που σημαίνει γραφή οπότε μαζί σημαίνει “κρυμμένο κείμενο”. Αποτελεί μια μέθοδο απόκρυψης κρυφών δεδομένων με την ενσωμάτωση σε ήχο, βίντεο, εικόνα ή κείμενο και χρησιμοποιείται για την προστασία μυστικών ή ευαίσθητων δεδομένων από κακόβουλες επιθέσεις.

Watermarking

Ένα digital watermark είναι ένα είδος marker κρυφά ενσωματωμένου σε ένα noise-tolerant σήμα όπως είναι ο ήχος, το βίντεο ή μια εικόνα. Συχνά, χρησιμοποιείται για την ταυτοποίηση της ιδιοκτησίας των copyright ενός τέτοιου σήματος. Η τεχνική Watermarking είναι η διαδικασία απόκρυψης ψηφιακών πληροφοριών σε ένα φέρον σήμα και μάλιστα, η κρυμμένη πληροφορία πρέπει να περιέχει μια συσχέτιση με το φέρον σήμα.

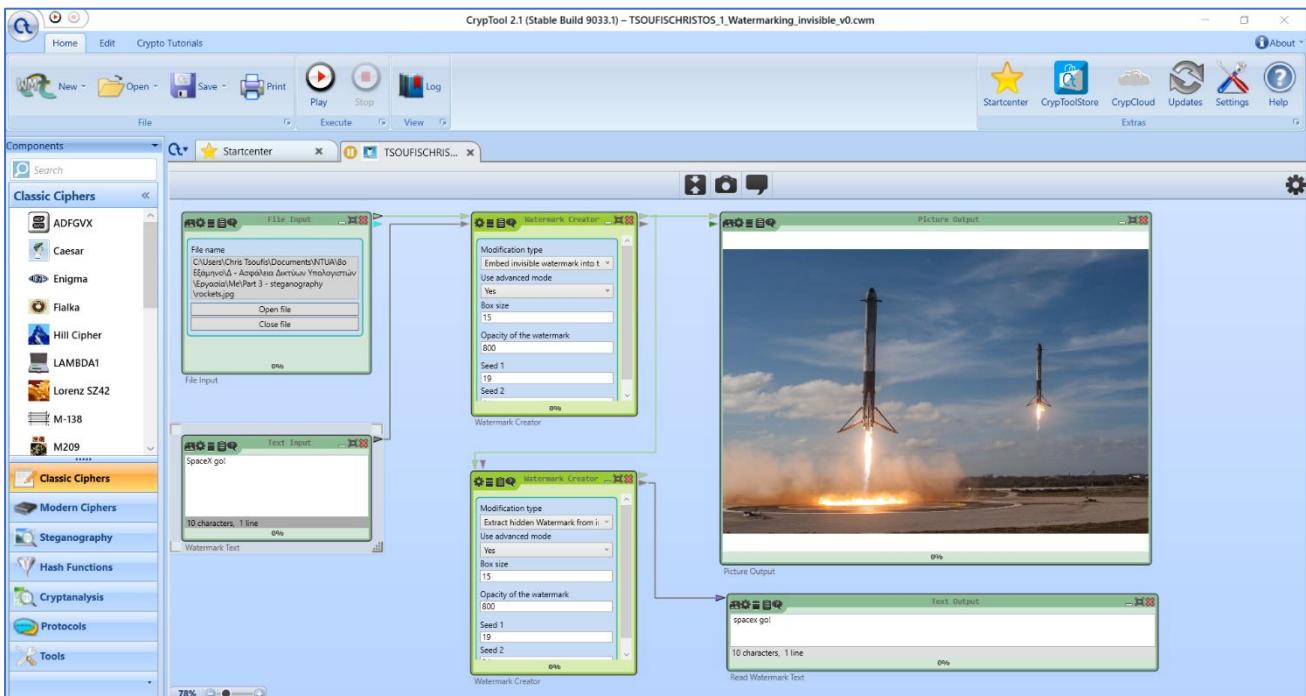
Επομένως, για το Watermarking, θα χρησιμοποιηθεί η παρακάτω φωτογραφία:



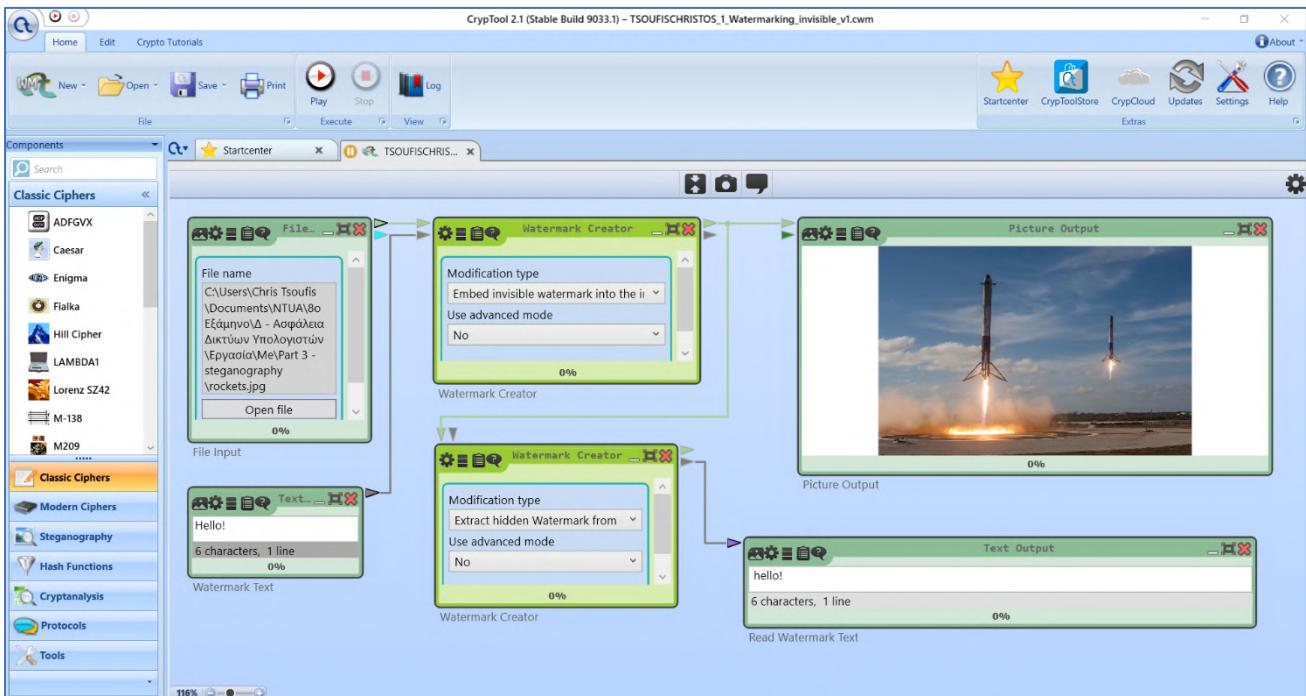
Αρχικά, δημιουργείται ένα Αόρατο Watermark.

Το έτοιμο template του Cryptool δείχνει το πως μπορεί κανείς να προσθέσει ένα αόρατο (invisible) watermark σε μια εικόνα. Στο “File input” component μπορεί να επιλεχθεί η εικόνα που θα χρησιμοποιηθεί και στο “Text Input” component μπορεί να τροποποιηθεί το κείμενο που θα χρησιμοποιηθεί. Αυτό το component προσφέρει μερικές ρυθμίσεις αν επιλεχθεί το “Advanced mode”, αλλιώς παραμένουν οι default. Τέλος, το watermark διαβάζεται και από ένα άλλο “Watermark Creator” component και παρουσιάζεται στο “Text Output” component.

Χωρίς “Advanced mode”:



Με “Advanced mode”:

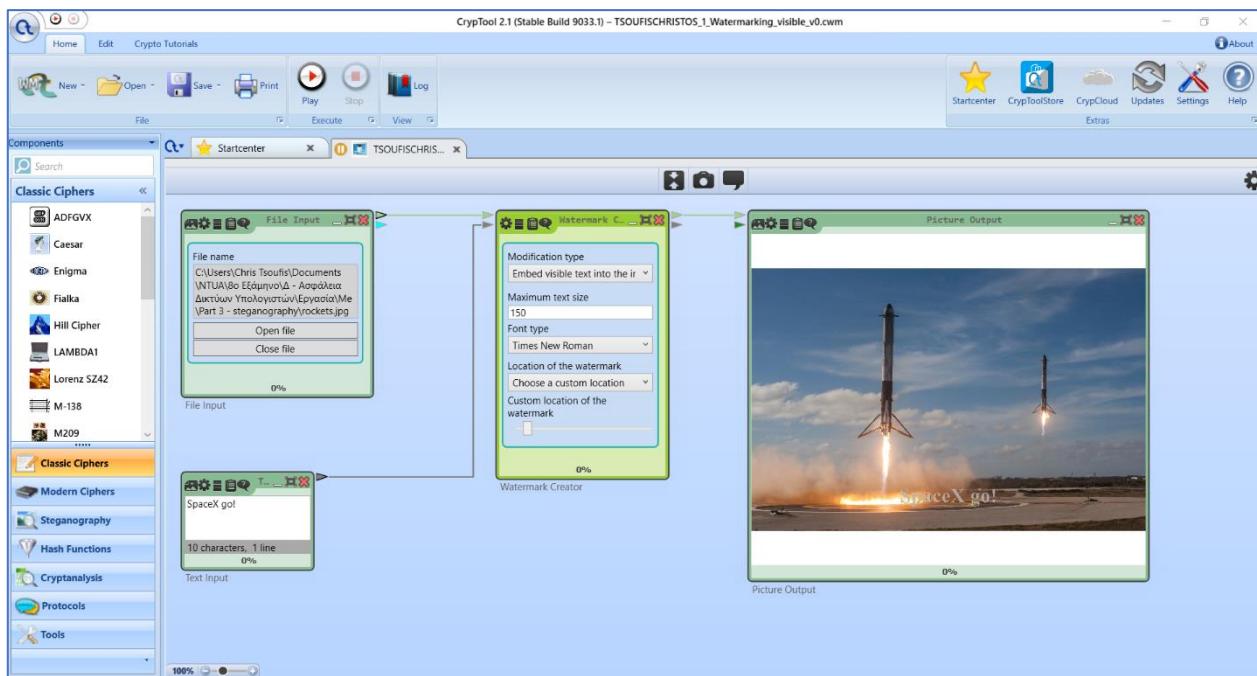


Σχολιασμός & Παρατηρήσεις:

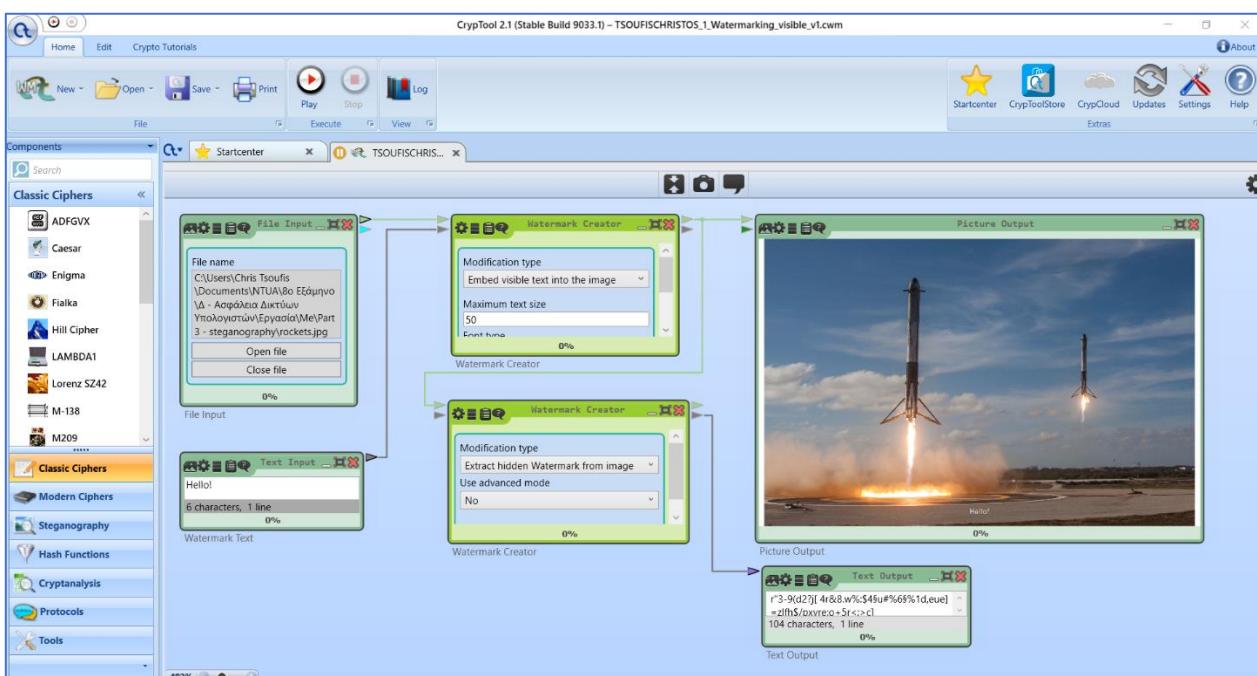
Στην εικόνα δεν παρατηρείται κάποια διαφορά και το μήνυμα εξάγεται από την εικόνα. Επιπλέον, τα κεφαλαία γράμματα μετατράπηκαν σε πεζά αφού από το κείμενο “SpaceX go!” προκύπτει “spacex go!”.

Έπειτα, δημιουργείται ένα άλλο project για το Ορατό (Visible) Watermark. Εδώ, και με γυμνό μάτι είναι ορατό το watermark.

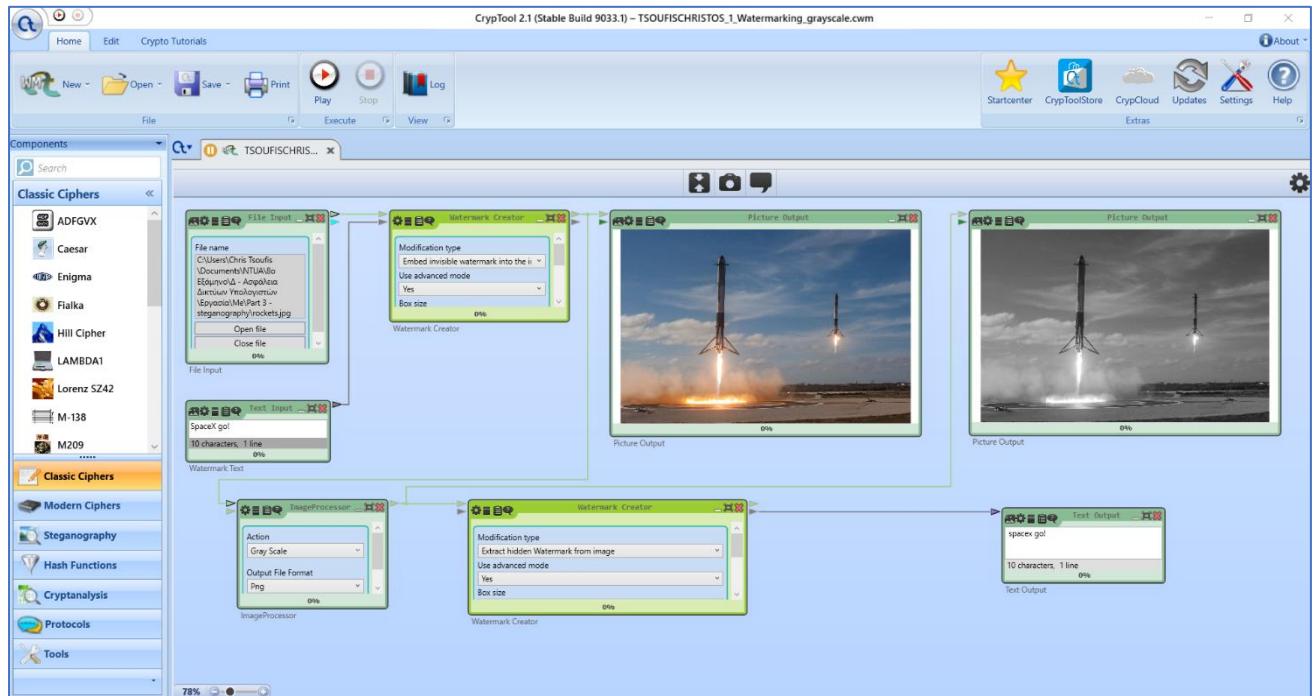
Αυτό το template του CrypTool, δείχνει πως να ενσωματώσει κανείς ορατό κείμενο ως watermark σε μια εικόνα. Στο "File input" component επιλέγεται η εικόνα που θα χρησιμοποιηθεί. Στο "Text input" component γράφεται το κείμενο που θα προστεθεί στην εικόνα ως watermark. Αυτό το μπορεί κανείς να το αλλάξει όσο εκτελείται αυτό το component. Από τις ρυθμίσεις του component μπορεί να επιλέξει κανείς (από πάνω προς τα κάτω): (1) το maximum μέγεθος text που θα χρησιμοποιηθεί για το watermark text, (2) το font που θα χρησιμοποιηθεί για το watermark και (3) την θέση του watermark στην εικόνα.



Και μικρές προσθήκες, φαίνεται ότι το κείμενο δεν μπορεί να εξαχθεί.



Τέλος, με τον ImageProcessor του CrypTool και επιλέγοντας action Gray Scale, μπορεί να εξαχθεί μπορεί να εξαχθεί σωστά το μυστικό μήνυμα (watermark) ακόμα και αν πρώτα εφαρμοστεί στη φωτογραφία Gray Scale.



2. Εφαρμογή στεγανογραφίας με χρήση εικόνας και τη μέθοδο LSB (τελευταίο σημαντικό ψηφίο). Η μέθοδος εισαγωγής LSB τροποποιεί το LSB κάθε χρώματος στις εικόνες 24-bit ή 8-bit. Ξεκινώντας από ανθαίρετη εικόνα και κάνοντας χρήση του Cryptool και ειδικότερα του LSB Steganography, να προστεθεί ένα μυστικό μήνυμα στην εικόνα αυτή. Στη συνέχεια να αποκρυπτογραφηθεί το μυστικό μήνυμα από την εικόνα. Να γίνει σύγκριση των δύο εικόνων, πριν την εισαγωγή και μετά την εισαγωγή του μυστικού μηνύματος. Να δοθεί εξήγηση σχετικά με το πώς λειτουργεί η συγκεκριμένη μέθοδος στεγανογραφίας. Να σχολιαστούν τα πλεονεκτήματα και τα μειονεκτήματα.

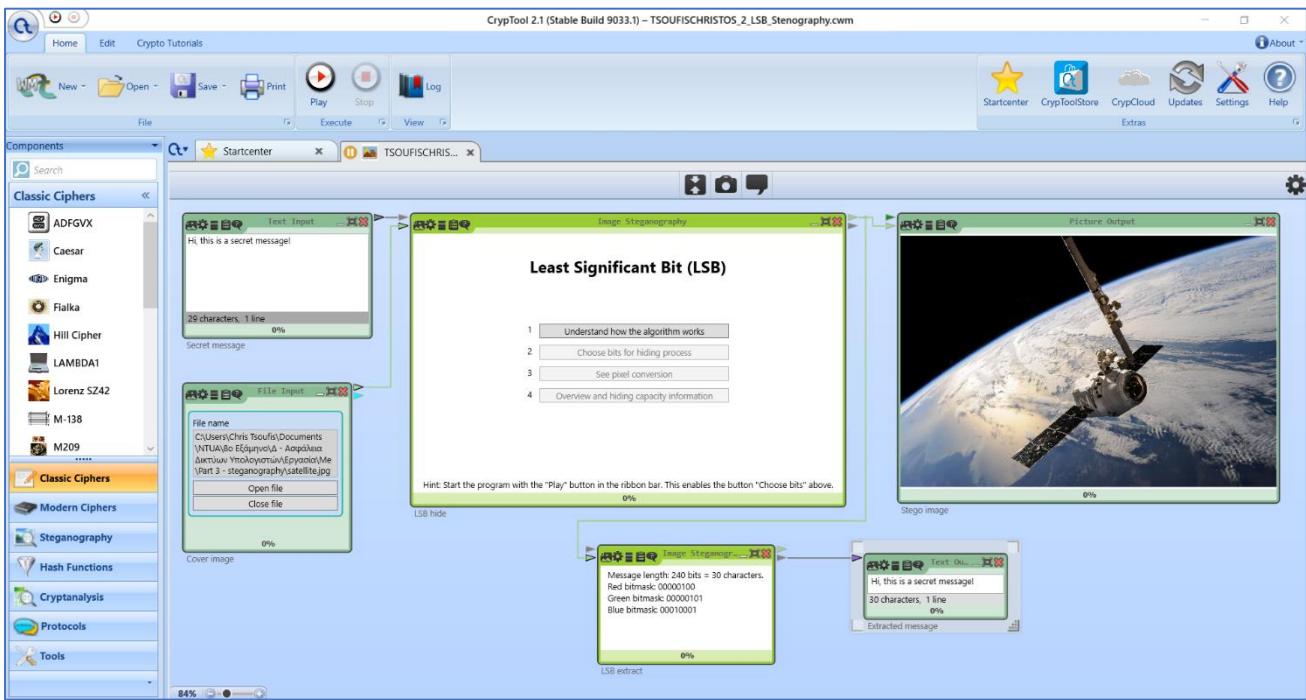
Το έτοιμο template απεικονίζει την Steganography εικόνας με χρήση της LSB τεχνικής. Οι bitmasks επιλέγονται από τις ρυθμίσεις ή από το presentation αν ενεργοποιηθεί από το "Choose bits" (2). Τα bitmasks δείχνουν ποια bits χρησιμοποιούνται για την απόκρυψη του μηνύματος (1 = chosen, 0 = not chosen). Όταν επιλέξει κανείς παραπάνω bits, μπορεί να πετύχει υψηλότερο hiding capacity και λιγότερα pixels επηρεάζονται από τις αλλαγές, αλλά αυτό μπορεί να οδηγήσει σε distortion της εικόνας. Επομένως, τα υπόλοιπα views χρησιμοποιούνται για την σύγκριση και επιλογή των καλύτερων ρυθμίσεων με βάση τα provided inputs. Μετά την εκτέλεση της κρυφής διαδικασίας, μπορεί να γίνει σύγκριση των pixels μεταξύ της original και της cover image στο pixel conversion view (3). Στο overview (4) μπορεί να βρει κανείς πληροφορίες για την εικόνα και το hiding capacity.

Το LSB σημαίνει Least Significant Bit και η ιδέα πίσω από την ενσωμάτωση LSB είναι η εξής ότι εάν γίνει αλλαγή στο τελευταίο bit value ενός pixel, δεν θα υπάρχει μεγάλη ορατή αλλαγή στο χρώμα. Για παράδειγμα, το 0 να είναι μαύρο. Αλλάζοντας την τιμή σε 1, δεν θα κάνει μεγάλη διαφορά αφού ακόμα θα είναι μαύρο αλλά σε πιο διάφανη σκιά. Περισσότερες λεπτομέρειες αναφέρονται και στο CryptTool.

Για αυτό το ερώτημα, επιλέγεται η εικόνα “satellite”. Η αρχική εικόνα:



Έτσι, η εικόνα που προκύπτει είναι η ακόλουθη.



Και σε μια πιο σύνθετη διάταξη, η εικόνα είναι η ακόλουθη:



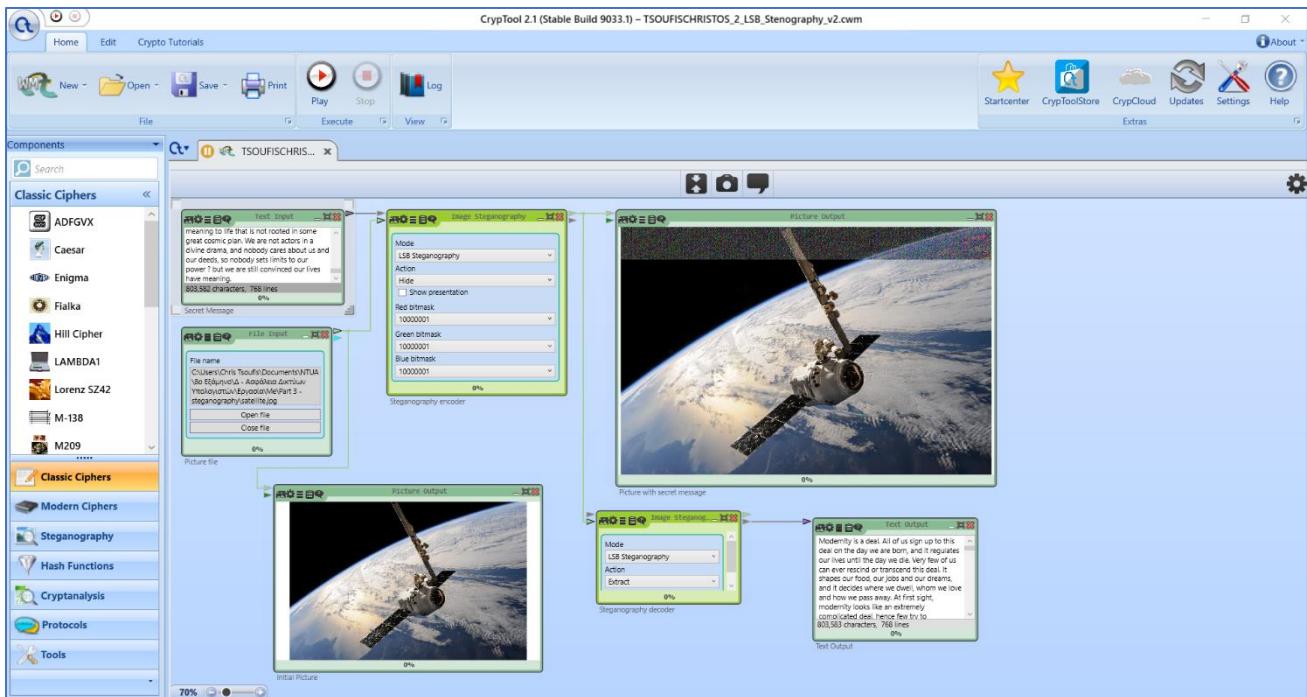
Εδώ, ως ciphertext, επιλέχθηκε το ίδιο μεγάλο κείμενο με το προηγούμενο μέρος.

Σχολιασμός & Παρατηρήσεις:

Παρατηρείται ότι το μήνυμα κωδικοποιείται και αποκωδικοποιείται χωρίς κάποιο πρόβλημα και δεν φαίνεται κάποια μεταβολή στην εικόνα που χρησιμοποιήθηκε. Αν ληφθεί υπόψη ο τρόπος λειτουργίας της μεθόδου LSB, το αποτέλεσμα αυτό είναι αναμενόμενο. Ειδικότερα, μια εικόνα κωδικοποιείται από pixels, και κάθε pixel περιλαμβάνει ειδικά bytes για την απόχρωση σε RGB. Η μέθοδος LSB χρησιμοποιεί το τελευταίο bit από κάθε χρώμα για την αποθήκευση του κρυφού μηνύματος. Όμως, με την αλλαγή του lsb, η διαφορά στην απόχρωση είναι σχεδόν αδύνατο να εντοπισθεί με γυμνό μάτι.

Για να δει κανείς μια αλλαγή πρέπει να χρησιμοποιήσει το MSB και να αυξήσει το text input κατά πολύ (εδώ 800.000 χαρακτήρες) οπότε η εικόνα θα είναι η ακόλουθη.

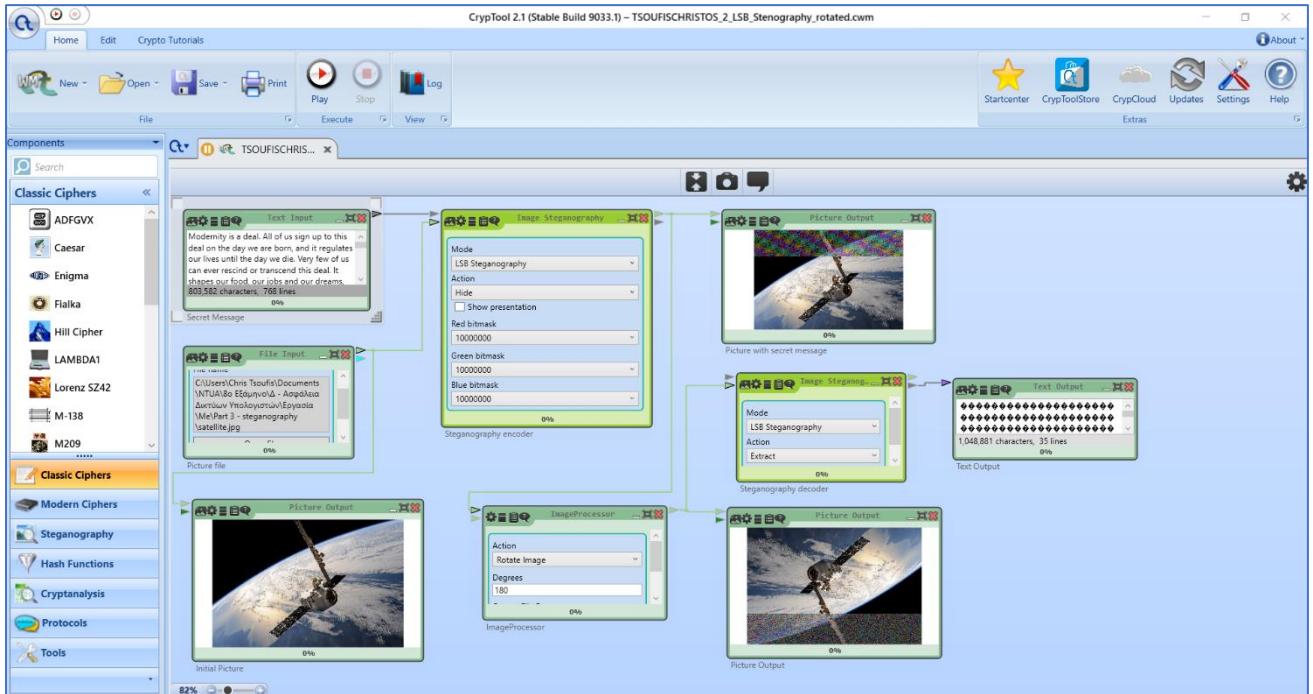
Σημειώνεται ότι δεν έχει παραμορφωθεί ολόκληρη η φωτογραφία διότι το cipher text δεν είναι αρκετά μεγάλο για να καλύψει όλα τα pixels της εικόνας.



Κύριο πλεονέκτημα της LSB μεθόδου είναι ότι, ειδικά σε πολύ μεγάλης ευκρίνειας φωτογραφίες, μπορεί να ενσωματωθεί μια σημαντική κρυφή πληροφορία, χωρίς να γίνει αντιληπτό. Αυτό είναι ιδιαίτερα χρήσιμο σε περιπτώσεις κατασκοπίας. Σε διαύλους επικοινωνίας όπου διακινούνται καθημερινά εκατομμύρια φωτογραφίες, μια εικόνα με κρυφό μήνυμα μπορεί να περάσει πολύ εύκολα απαρατήρητη. Επιπλέον, η Steganography σε συνδυασμό με την κρυπτογράφηση οδηγεί σε μεγαλύτερη ασφάλεια.

Από την άλλη, στα μειονεκτήματα της μεθόδου πρέπει να επισημάνει κανείς ότι η εξαγωγή του κρυφού κειμένου εξαρτάται απόλυτα από την εικόνα με αποτέλεσμα, οποιαδήποτε αλλοίωσή της να την καθιστά δύσκολη. Επιπροσθέτως, είναι σχετικά εύκολο για έναν υπολογιστή να ανακαλύψει το μυστικό μήνυμα αν έχει την αρχική εικόνα όμως αυτό μπορεί να διορθωθεί αν παράλληλα με τη Steganography χρησιμοποιηθεί και κάποια κρυπτογραφική μέθοδος. Τέλος, η παραπάνω μέθοδος απαιτεί περίσσεια πληροφορία ώστε να αποκρύψει ένα σχετικά μικρό text ενώ η αποστολή του κρυφού μηνύματος απαιτεί και την ύπαρξη ενός ασφαλούς διαύλου.

Ένα παράδειγμα φαίνεται παρακάτω, κατά το οποίο έχει γίνει περιστροφή κατά 180° και το CrypTool αδυνατεί να εξάγει το κρυφό μήνυμα.



3. Εφαρμογή στεγανογραφίας με χρήση εικόνας και τη μέθοδο BPCS. Ξεκινώντας από αυθαίρετη εικόνα και κάνοντας χρήση του Cryptool και ειδικότερα του Image Steganography να προστεθεί ένα μυστικό μήνυμα στην εικόνα αυτή. Στη συνέχεια, να αποκρυπτογραφηθεί το μυστικό μήνυμα από την εικόνα. Να γίνει σύγκριση των δύο εικόνων, πριν την εισαγωγή και μετά την εισαγωγή του μυστικού μηνύματος. Να δοθεί εξήγηση σχετικά με το πώς λειτουργεί η συγκεκριμένη μέθοδος στεγανογραφίας. Να σχολιαστούν τα πλεονεκτήματα και τα μειονεκτήματα σε σχέση με την LSB.

BPCS Steganography

Η digital steganography, μπορεί να κρύψει εμπιστευτικές πληροφορίες με ασφάλεια ενσωματώνοντας τες σε κάποια media data που λέγονται “vessel data”. Στην BPCS-steganography, οι true color εικόνες χρησιμοποιούνται κυρίως για vessel data. Στην πράξη, αντικαθίστανται οι “complex areas” στα bit planes της vessel image με τα εμπιστευτικά δεδομένα. Σημειώνεται ότι η χωρητικότητα ενσωμάτωσης είναι πολύ μεγάλη σε αντίθεση με τις υπόλοιπες image based steganography.

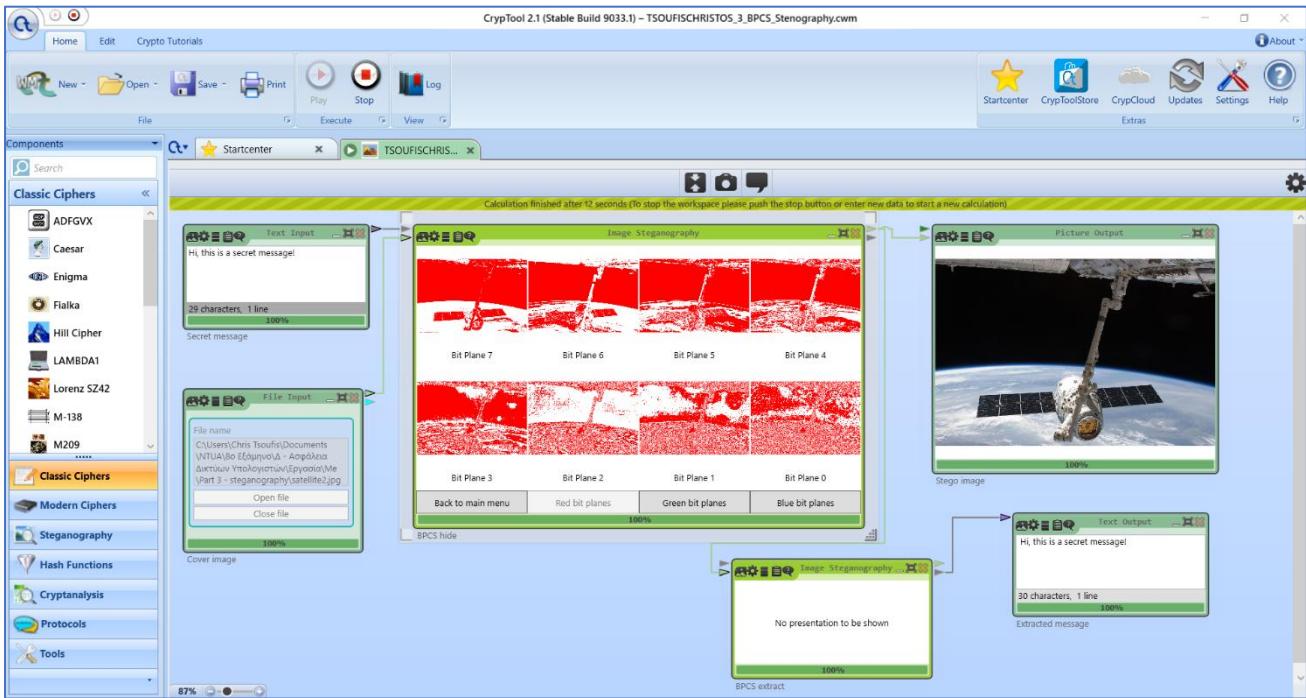
Αρχικά, χρησιμοποιείται το έτοιμο template για την image steganography με χρήση της BPCS technique. Με την έναρξη του προγράμματος μπορεί να δει κανείς το created message και τα image blocks στο hiding process view (2). Επίσης, μπορεί να γίνει σύγκριση των pixels μεταξύ της original και της cover image στο pixel conversion view (3). Τα διαφορετικά bit planes της εικόνας μπορεί να τα δει κανείς αν πλοηγηθεί στο bit planes view (4). Στις ρυθμίσεις, μπορεί να τεθεί το complexity threshold και η σειρά των color channels που χρησιμοποιούνται κατά το hiding/extracting του message.

Ο BPCS algorithm εισήχθη για να ξεπεράσει τις παραδοσιακές steganography techniques με περιορισμό στο data hiding capacity. Αυτή η τεχνική μπορεί να κρύψει πάνω από το 50% της cover image data amount χωρίς μεγάλο distortion στην visual appearance της original image. Στην BPCS, μια multi-valued εικόνα που αποτελείται από 24-bit pixels μπορεί να γίνει decomposed σε ένα σύνολο από 24 binary εικόνες (bit planes). Αυτά τα planes χωρίζονται σε 8×8 pixel blocks και τα blocks μετά χωρίζονται σε noise like περιοχές και informative περιοχές. Περισσότερες λεπτομέρειες αναφέρονται και στο CrypTool.

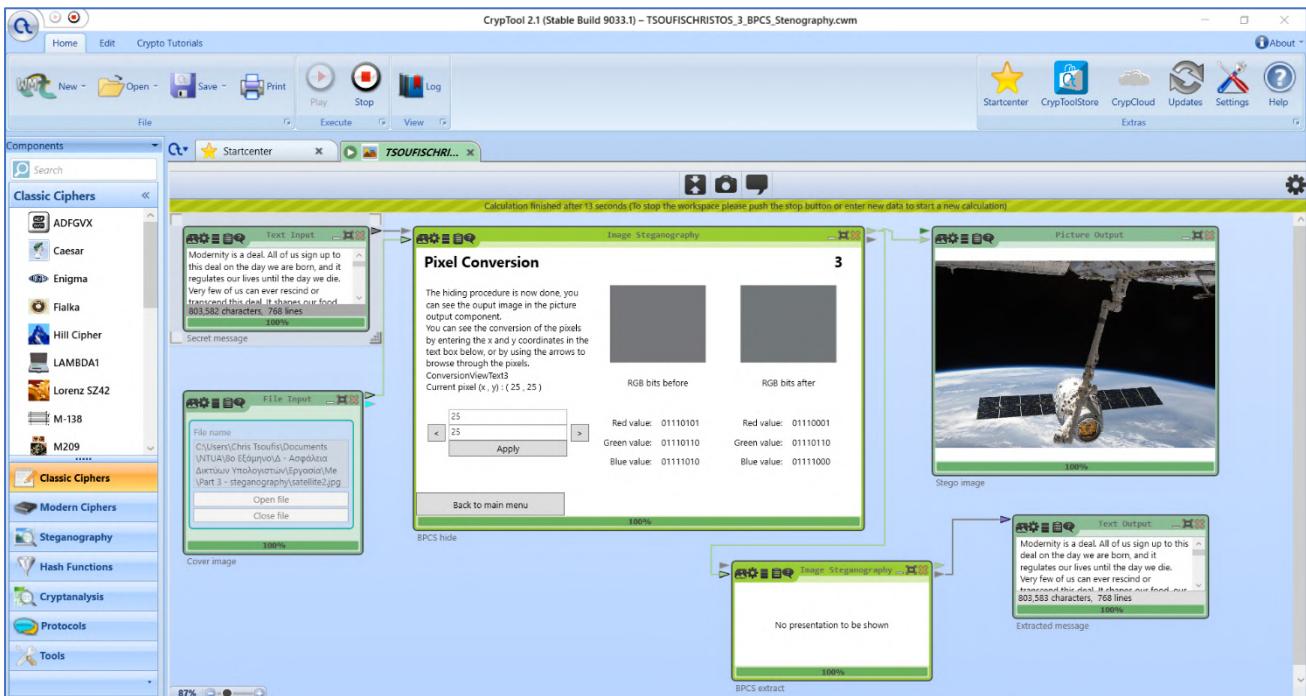
Εδώ, χρησιμοποιείται η ακόλουθη εικόνα με όνομα “satellite2”.



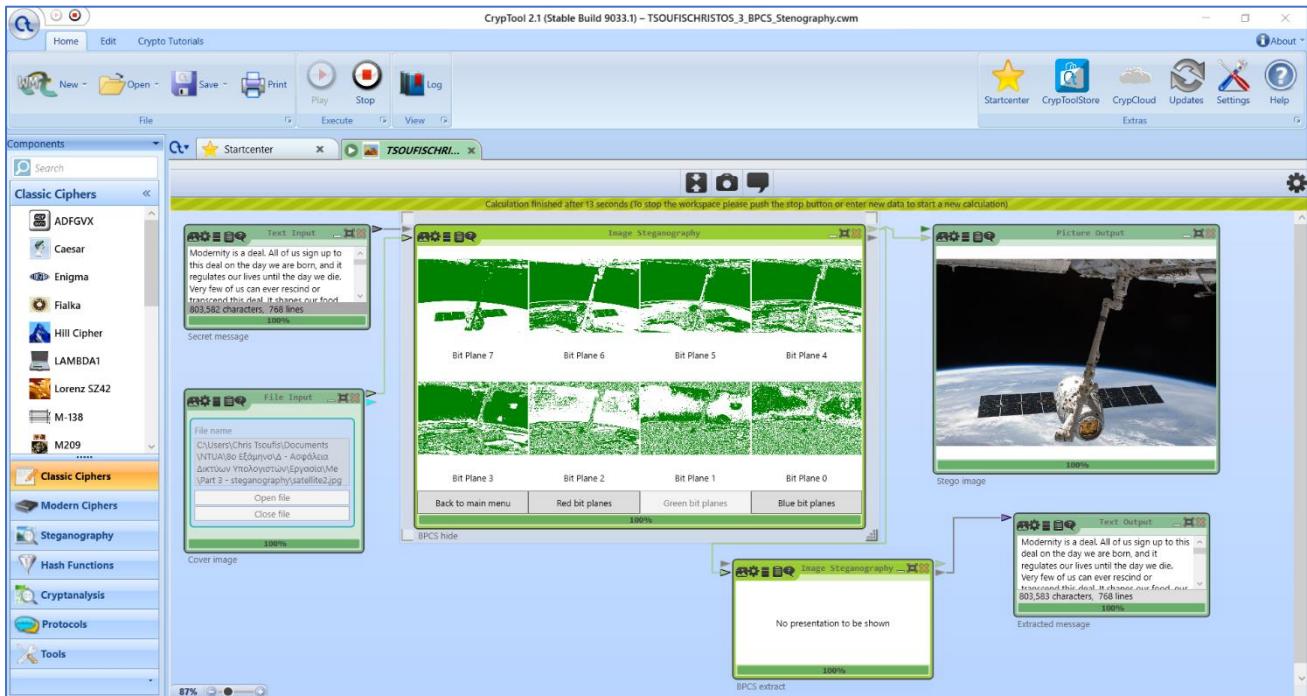
Και η διάταξη φαίνεται παρακάτω.



Με την εισαγωγή πολύ μεγαλύτερου κειμένου, η εικόνα είναι ακόλουθη.



Και τροποποιώντας τις παραμέτρους, προκύπτει η ακόλουθη εικόνα.



Εναλλακτικά, μπορεί να υπολογίσει κανείς την BPCS Steganography και με την ακόλουθη υλοποίηση. Ουσιαστικά, με την χρήση ενός open source κώδικα από το Github (βλ. Resources), μπορεί κανείς να κάνει την κωδικοποίηση και αποκωδικοποίηση ενός μηνύματος.

Η Στεγανογραφία (midi insertion) τοποθετεί πρώτα το μήνυμα προς κρυπτογράφηση σε ένα midi file και έπειτα τοποθετείται η μουσική που θα παιχτεί από το κομμάτι. Υστερα, ο interpreter του midi file βρίσκει το μήνυμα ανάμεσα στις περιττές πληροφορίες που υπάρχουν στο κομμάτι.

Σχολιασμός & Παρατηρήσεις:

Στα πλεονεκτήματα του BPCS, πρέπει να αναφερθεί η αποδοτικότητα του αλγορίθμου. Επιπλέον, είναι αξιοσημείωτο ότι είναι εξαιρετικά δύσκολο να εντοπιστεί το αρχικό κείμενο καθώς δεν αλλοιώνει το τελικό αρχείο.

Στα μειονεκτήματα του BPCS, πρέπει να επισημανθεί ότι είναι δύσκολη η απόκρυψη μεγάλου μεγέθους πληροφορίας διότι αυτό απαιτεί αρκετά μεγάλα αρχεία (π.χ. midi για την παραπάνω περίπτωση). Ακόμη, τονίζεται πως το κωδικοποιημένο αρχείο (π.χ. midi) που παράγεται θα είναι και αυτό αρκετά μεγάλο σε μέγεθος και αυτό μπορεί να είναι και δύσχρηστο αλλά και ύποπτο.

4. Σύγχρονοι Αλγόριθμοι Κρυπτογράφησης

Στο κομμάτι αυτό της εργασίας θα μελετηθούν αντιπροσωπευτικοί σύγχρονοι αλγόριθμοι κρυπτογράφησης. Θα χρησιμοποιηθεί το εργαλείο *Cryptool 2.1*, το οποίο αποτελεί πλατφόρμα για κρυπτογράφηση και κρυπτανάλυση. Το *CrypTool* διαθέτει εύχρηστα built-in tutorials και documentation.

Όπως και πριν, όπου ζητείται screenshot από το *CrypTool*, θα πρέπει να αποθηκεύεται το project με όνομα αρχείου το ονοματεπώνυμο με λατινικούς χαρακτήρες, underscore και αύξοντα αριθμό της ερώτησης (π.χ., *PAPADOPoulosnikos_1.cwm*) και ύστερα, print screen + paste στο αρχείο που θα παραδώσετε, ώστε να φαίνεται και το όνομα του project.

Σχετική όλη: Συμμετρικοί αλγόριθμοι: XOR, DES, TripleDES, AES, Κρυπτογράφηση δημοσίου Κλειδιού: RSA.

Ζητούνται τα ακόλουθα:

1. Ξεκινώντας από αυθαίρετο κείμενο (τουλάχιστον 10000 Bytes) και κάνοντας χρήση του *Cryptool* να πραγματοποιηθεί κρυπτογράφηση κάνοντας χρήση συμμετρικών αλγορίθμων κρυπτογράφησης XOR, DES, TripleDES, AES χρησιμοποιώντας κλειδί κατάλληλου μήκους δικής σας επιλογής. Στη συνέχεια να πραγματοποιηθεί ανάλυση συχνότητας και σχολιάστε με βάση τις συχνότητες την ανθεκτικότητα των αλγορίθμων, εντοπίζοντας τα πλεονεκτήματα και τα μειονεκτήματά τους.

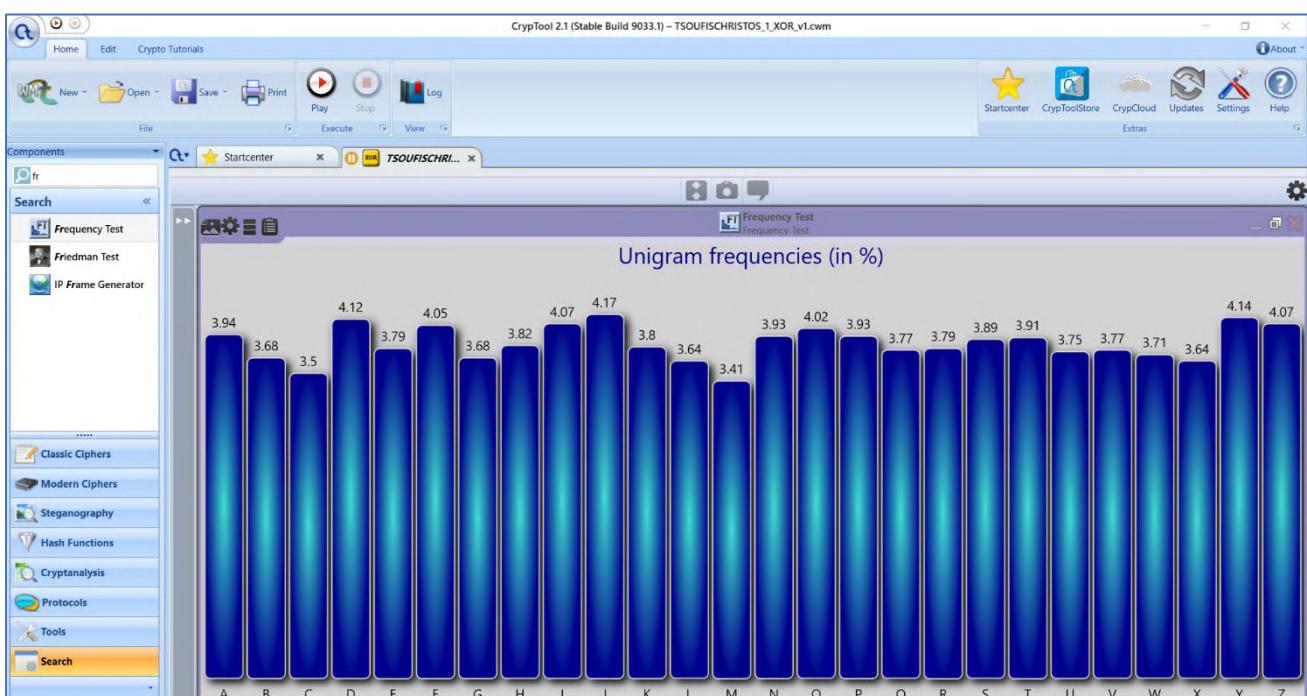
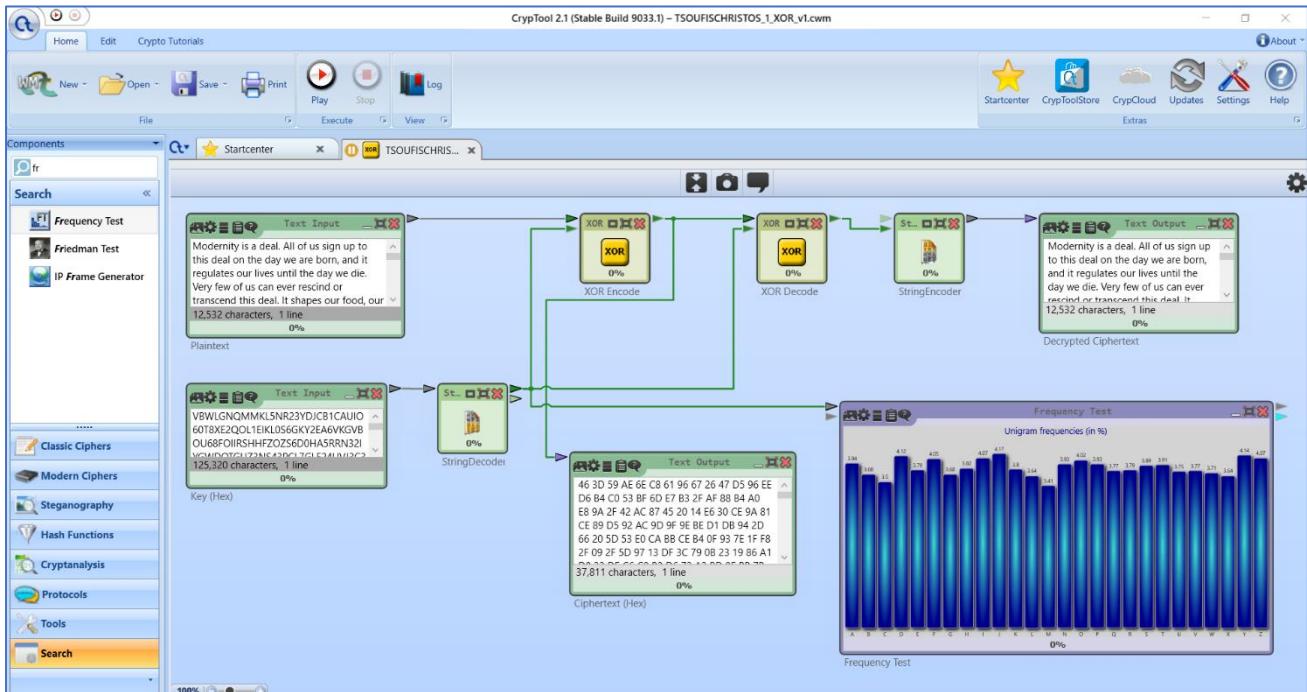
Γενική επισήμανση: Τα αρχεία με όνομα *v1* αποτελούν την υλοποίηση του Cipher, ακριβώς όπως δίνεται από το *CrypTool* ενώ αυτά με όνομα *v2* αποτελούν κάποιες παραλλαγές που υλοποιήθηκαν. Τα αποτελέσματα αναλύονται παρακάτω.

Το κείμενο είναι 12.2 KB.

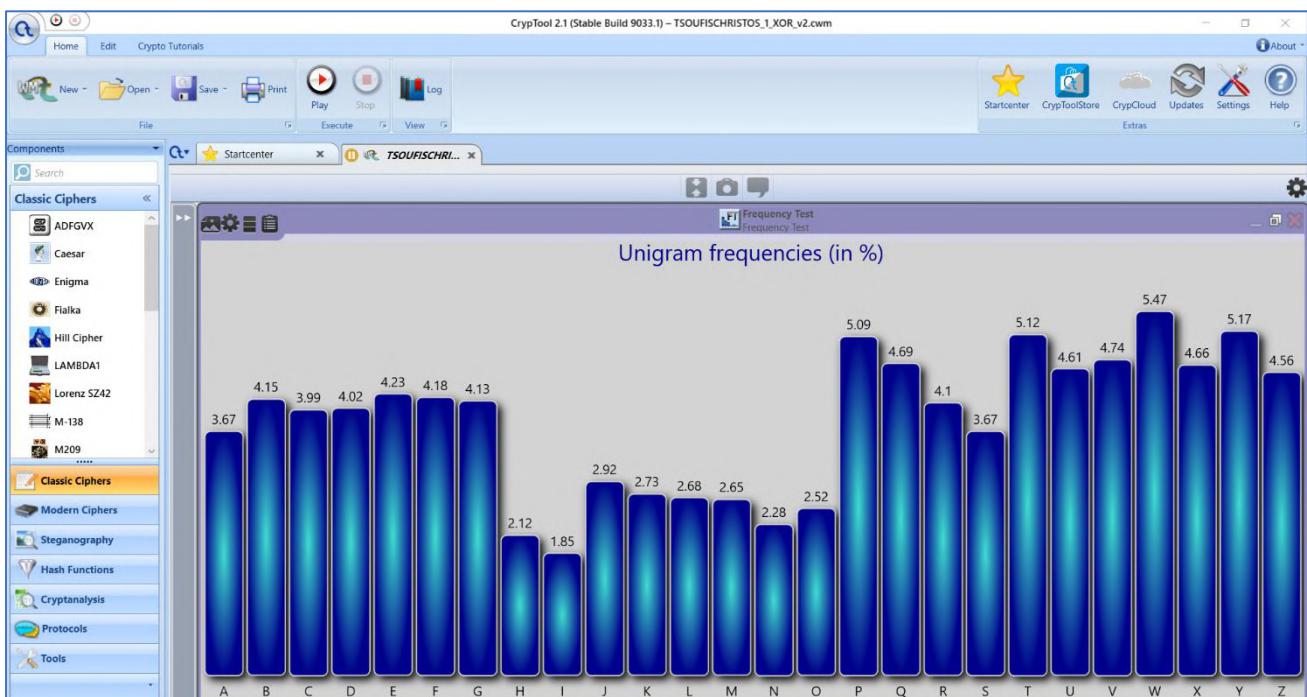
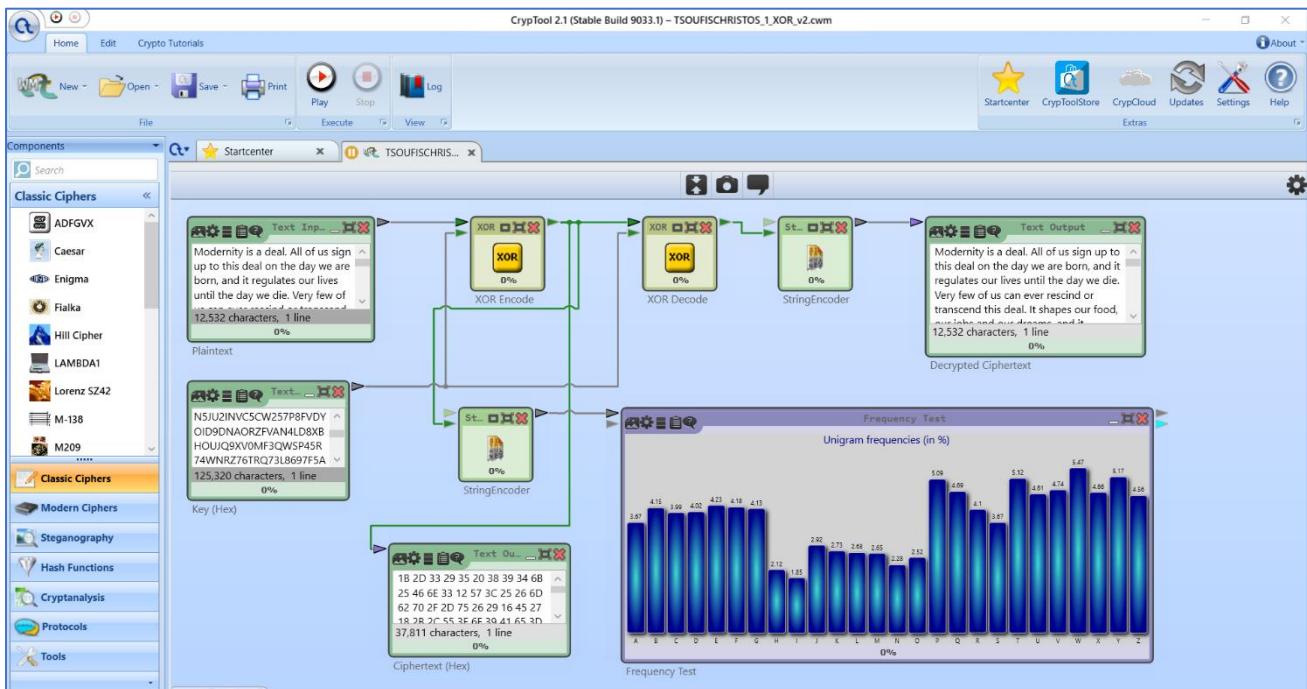
XOR Cipher

Ακολουθεί μια σύντομη αναφορά στον τρόπο λειτουργίας του αλγορίθμου κρυπτογράφησης XOR. Ο XOR Cipher είναι μια μέθοδος κρυπτογράφησης που χρησιμοποιείται για την κρυπτογράφηση δεδομένων και είναι δύσκολο να “σπάσει” με brute-force μέθοδο (π.χ. με την παραγωγή τυχαίων κλειδιών κρυπτογράφησης ώστε να ταιριάζουν με το σωστό). Η ιδέα της υλοποίησης είναι ότι πρώτα ορίζεται το XOR – κλειδί κρυπτογράφησης και έπειτα γίνεται XOR μεταξύ των χαρακτήρων στο στην συμβολοσειρά με το κλειδί κρυπτογράφησης. Για την αποκρυπτογράφηση, οι κρυπτογραφημένοι χαρακτήρες γίνονται XOR ξανά με το ορισμένο κλειδί.

Αρχικά, φαίνεται ο έτοιμος Cipher που δίνεται από το CrypTool.



Παρακάτω φαίνεται μια διαφορετική υλοποίηση του XOR Cipher.

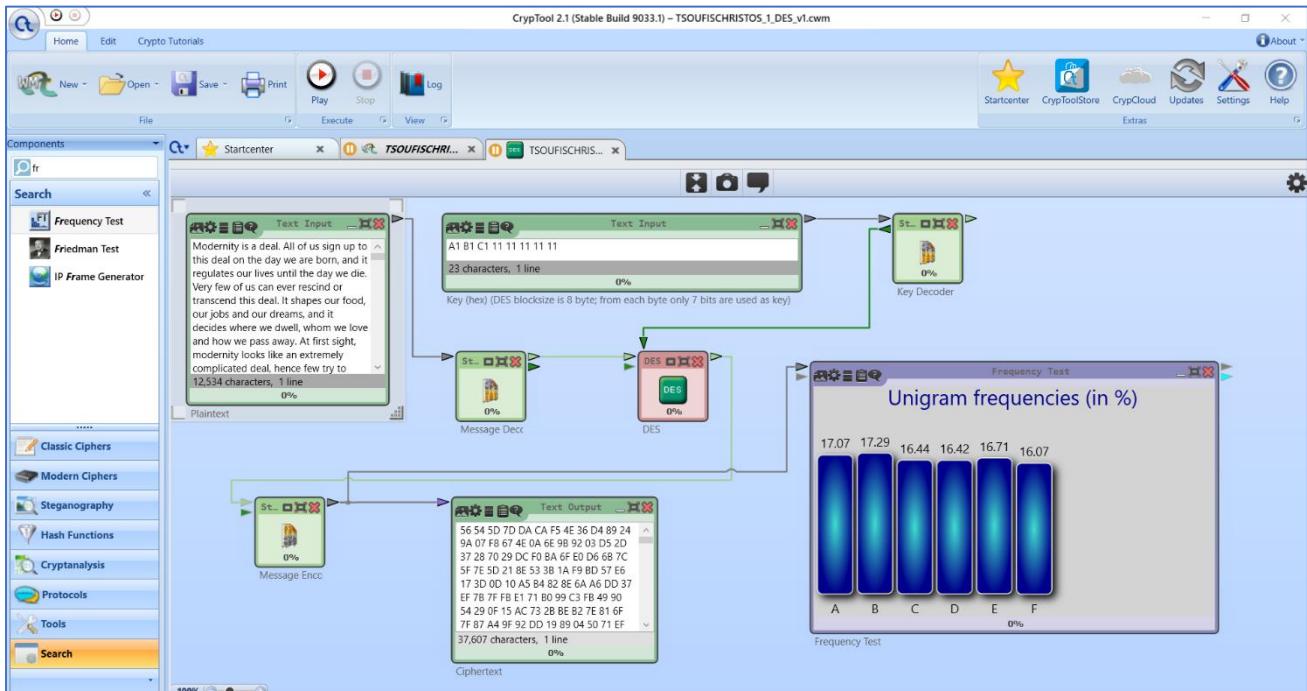


DES Cipher

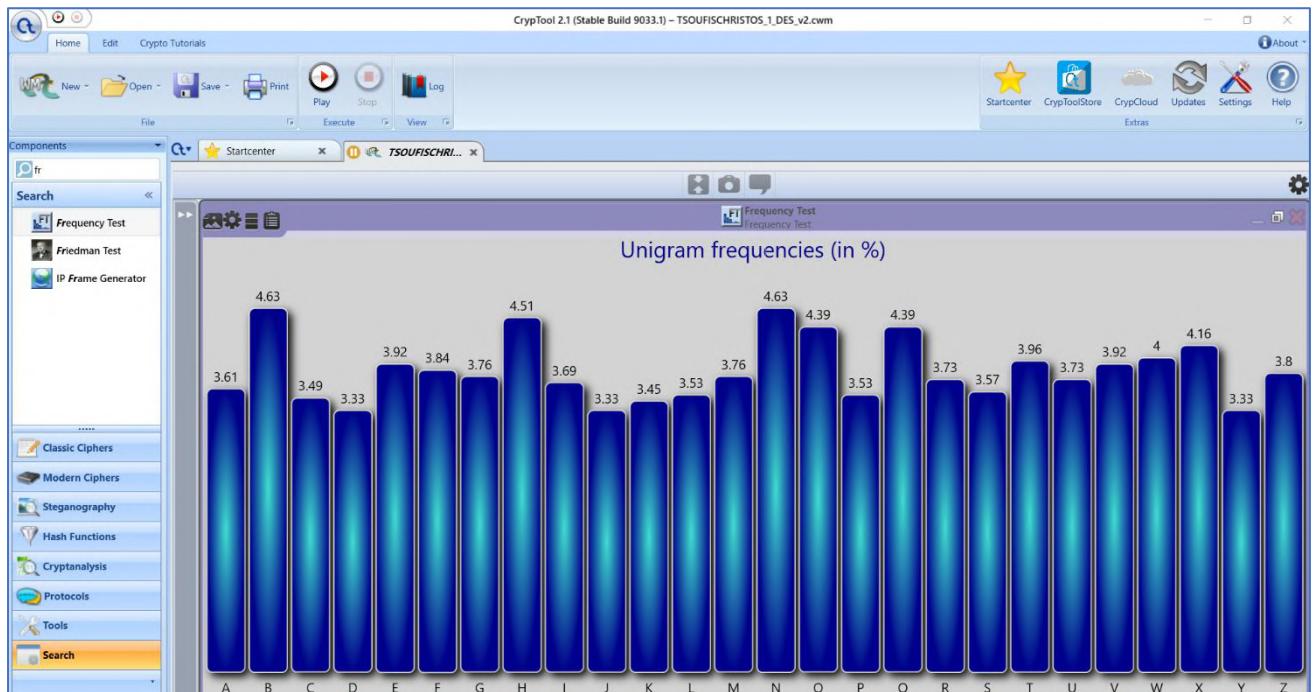
Ακολουθεί μια σύντομη αναφορά στον τρόπο λειτουργίας του αλγορίθμου κρυπτογράφησης DES. Ο DES Cipher είναι ένας block cipher και κρυπτογραφεί δεδομένα σε blocks μεγέθους 64 bits το καθένα. Δηλαδή, 64 bits plain text δίνονται ως input στον DES και παράγει 64 bits cipher text. Ο αλγόριθμος είναι σχεδόν ίδιος για κρυπτογράφηση και αποκρυπτογράφηση, μικρές διαφοροποιήσεις. Το μήκος του κλειδιού είναι 56 bits. Βασίζεται στην αντικατάσταση (substitution) και στην transposition. Αρχικά, το 64 bit plain text block δίνεται σε μια Initial Permutation (IP) συνάρτηση. Έπειτα, το IP παράγει δύο μέρη του permuted block, το δεξί (RPT) και το αριστερό (LPT). Μετά, κάθε LPT & RPT κάνει 16 γύρους για την κρυπτογράφηση. Στο τέλος, τα LPT & RPT ενώνονται και το Final Permutation (FP) σχηματίζεται στο combined block. Το αποτέλεσμα τελικά είναι ένα 64 bit cipher text. Σημειώνεται εδώ ότι ο DES έχει βρεθεί ευάλωτος σε πολύ δυνατές επιθέσεις και επομένως, η δημοτικότητά του φθίνει.

Με το template που δίνεται, μπορεί να χρησιμοποιηθεί το DES component για την κρυπτογράφηση ενός αυθαίρετου text που δίνεται στο component Plaintext στα αριστερά. Το κρυπτογραφημένο κείμενο που προκύπτει φαίνεται στο component Ciphertext στα δεξιά μετά το πάτημα του Play button. Το DES component λειτουργεί για δυαδικές τιμές (π.χ. bytes). Έτσι, το input text πρώτα μετατρέπεται σε bytes με το component Message decoder. Με αυτές τις συνθήκες, γίνεται interpreted σε ASCII. Τα bytes που προκύπτουν, κρυπτογραφούνται με το DES, και προκύπτει άλλη ακολουθία από bytes. Αυτά τα bytes απλώς τυπώνονται σαν δεκαεξαδικές τιμές με τη βοήθεια του component Message encoder. Σημειώνεται ότι μπορεί να γίνει και αποκρυπτογράφηση των μηνυμάτων με αυτό το template. Για να γίνει αυτό, πρώτα αντιγράφεται οι κρυπτογραφημένες δεκαεξαδικές τιμές στο Plaintext. Έπειτα, αλλάζοντας τα ακόλουθα: (1) το Input format του Message decoder tίθεται σε Hexadecimal, (2) το Action του DES σε Decrypt και (3) το Format του Message encoder σε Text και το Encoding σε ASCII. Επίσης, ο έλεγχος του TripleDES variant στο DES component απαιτεί αναπροσαρμογή και του κλειδιού. Αφού, ο DES algorithm χρειάζεται key μεγέθους 8 bytes, ενώ ο TripleDES algorithm χρειάζεται key είτε 16 είτε 24 bytes.

Έτσι, παρακάτω φαίνεται ο έτοιμος Cipher που δίνεται από το CrypTool.

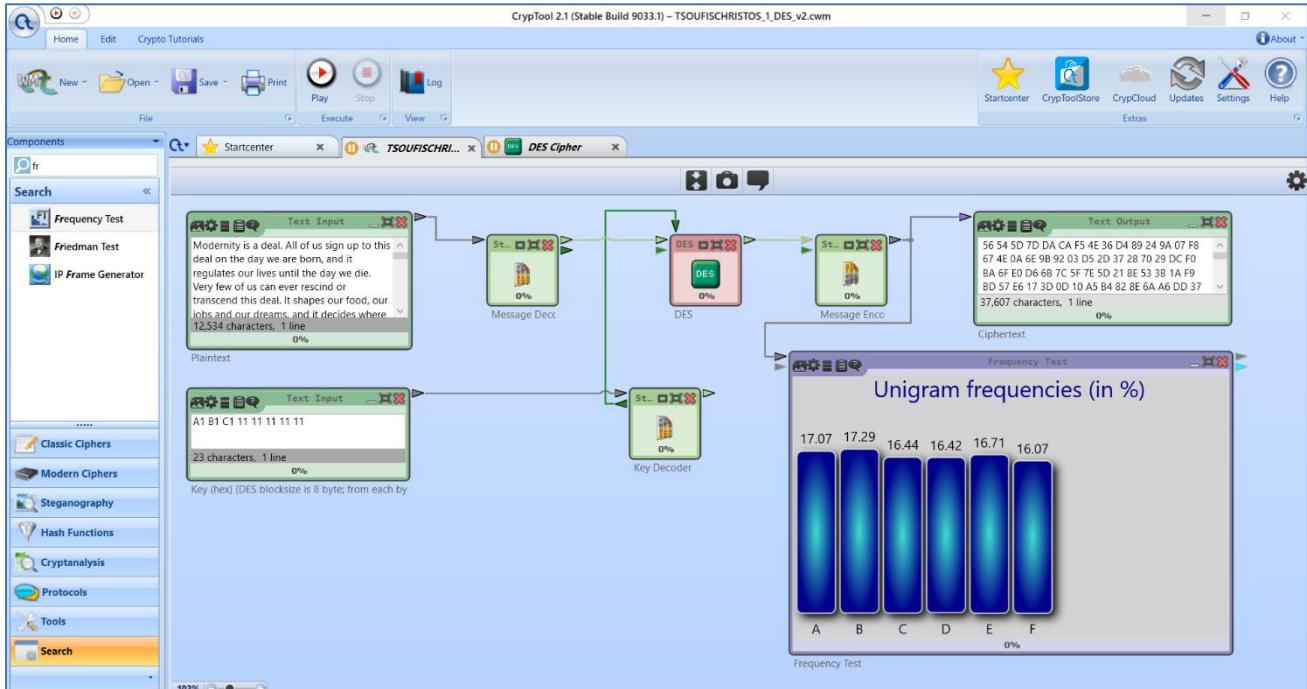


Παρακάτω φαίνεται μια ελαφρώς διαφορετική υλοποίηση του DES Cipher (πιθανώς να υπάρχει λάθος σε κάποιο component οπότε αποτυγχάνει η κρυπτογράφηση όπως φαίνεται στο output αλλά έχει προστεθεί ως παρατήρηση).

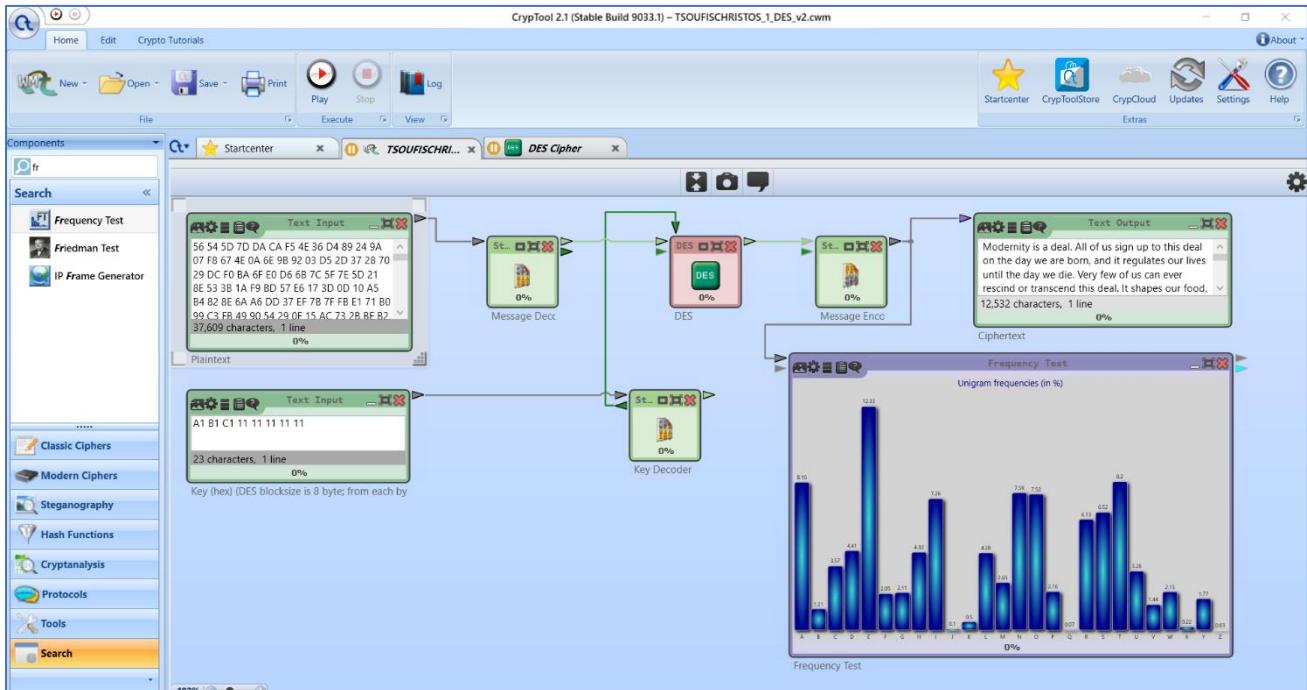


Για το ίδιο αρχείο, διορθώνοντας το λάθος, προκύπτει η παρακάτω εικόνα.

Για την κρυπτογράφηση:



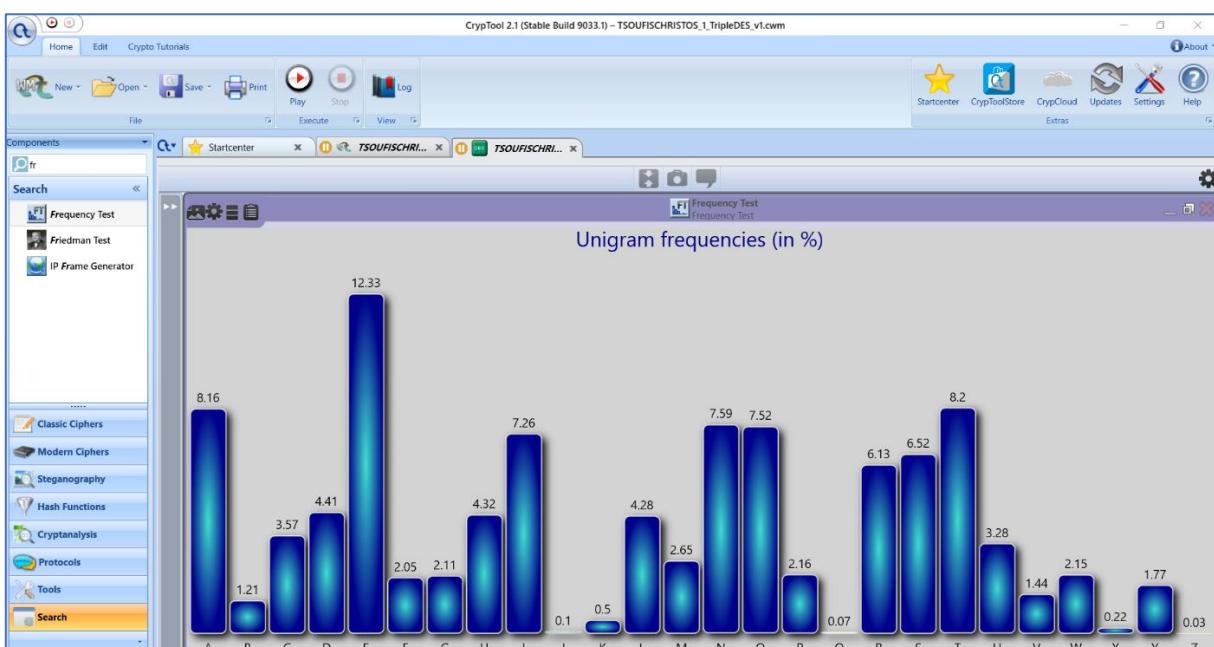
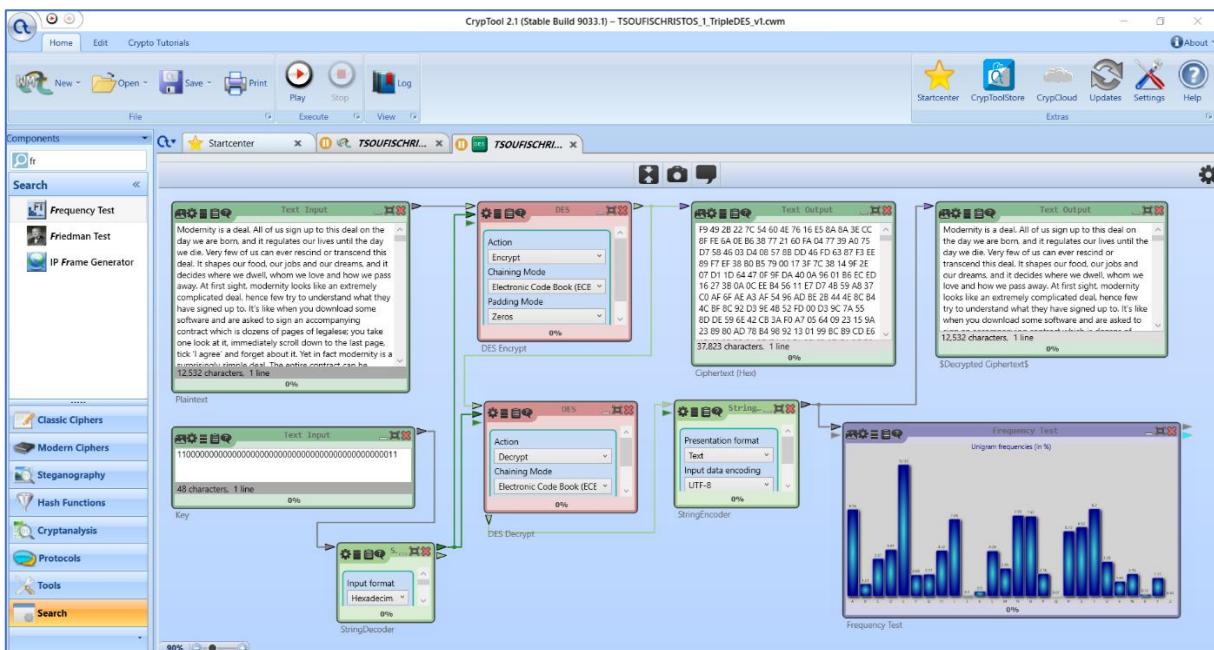
Για την αποκρυπτογράφηση:



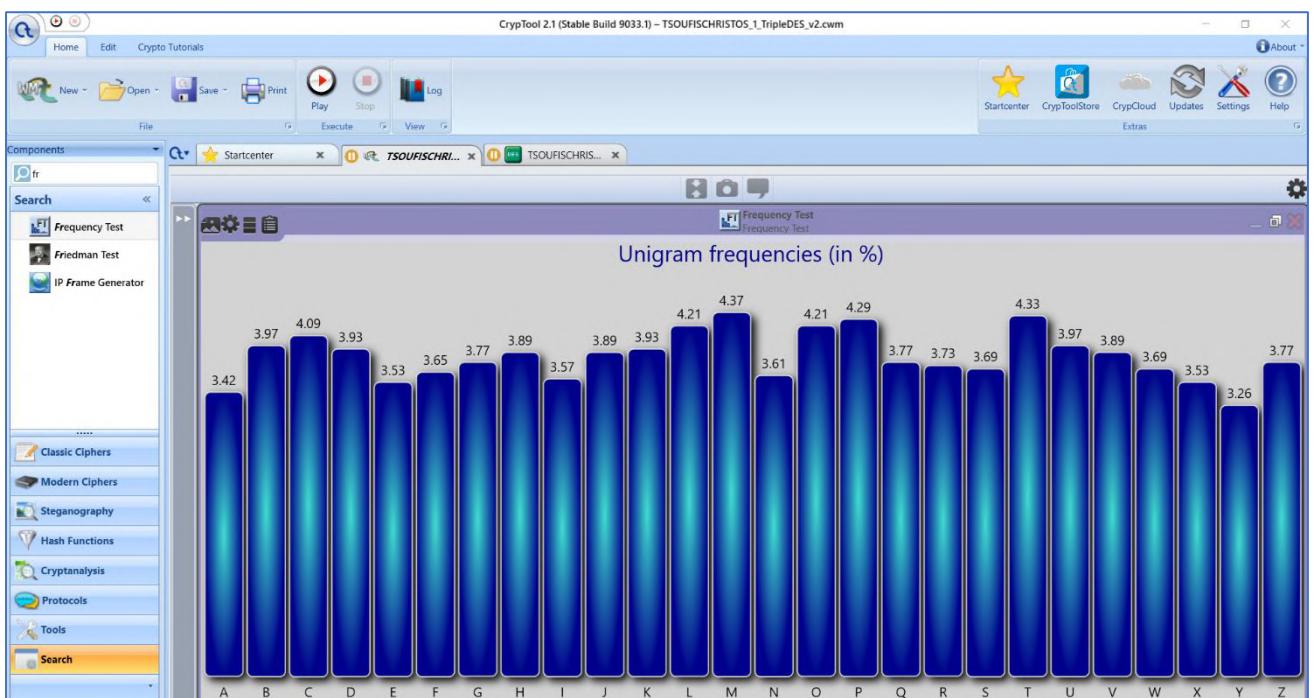
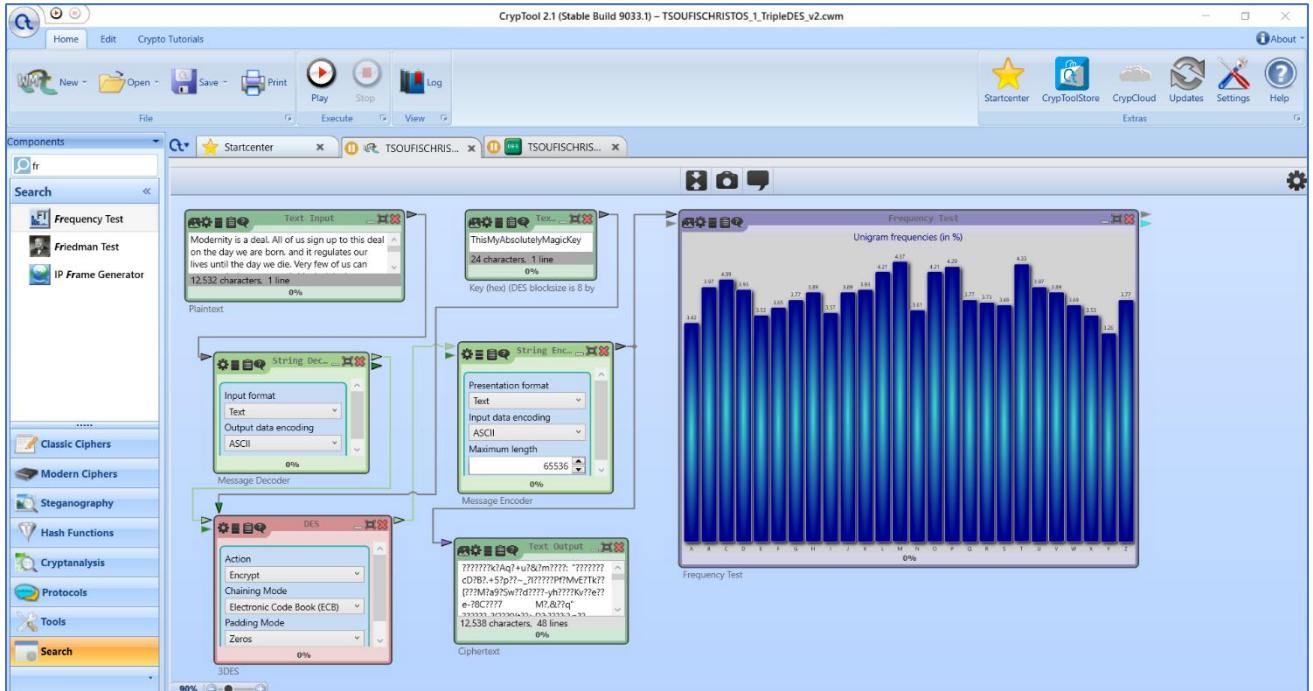
TripleDES Cipher

Ακολουθεί μια σύντομη αναφορά στον τρόπο λειτουργίας του αλγορίθμου κρυπτογράφησης TripleDES. Ο TripleDES Cipher είναι μια παραλλαγή του DES που στόχο έχει να είναι πιο ανθεκτική στις επιθέσεις. Χρησιμοποιεί κλειδί μήκους 168 bits και προσφέρει πολύ μεγαλύτερη ασφάλεια. Στην ουσία, χρησιμοποιεί 3 instances του DES στο ίδιο plain text. Μάλιστα, χρησιμοποιεί διαφορετικούς τύπους κλειδιών. Στην αρχή, όλα τα κλειδιά είναι διαφορετικά, έπειτα 2 κλειδιά είναι ίδια και 1 διαφορετικό και τέλος, όλα τα κλειδιά είναι ίδια. Σημειώνεται πως είναι ευάλωτος σε meet-in-the-middle attack επειδή δίνει ολική ασφάλεια της τάξης 2^{112} αντί να χρησιμοποιεί κλειδί 168 bit. Επιπλέον, μπορεί να δεχτεί block collision attack εξαιτίας του μικρού μεγέθους block και επειδή χρησιμοποιεί το ίδιο κλειδί για την κρυπτογράφηση μεγάλου κειμένου. Τέλος, αναφέρεται ότι είναι ευάλωτο και σε sweet32 attack.

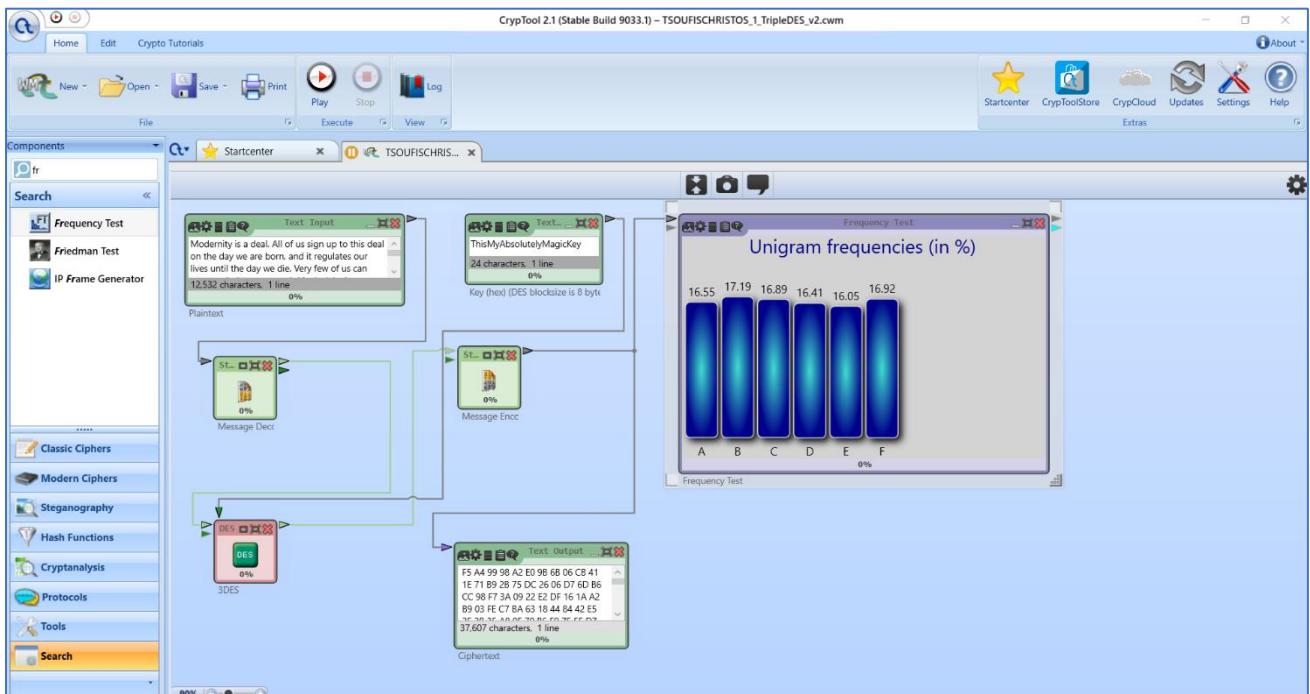
Έτσι, παρακάτω φαίνεται ο έτοιμος Cipher που δίνεται από το CrypTool.



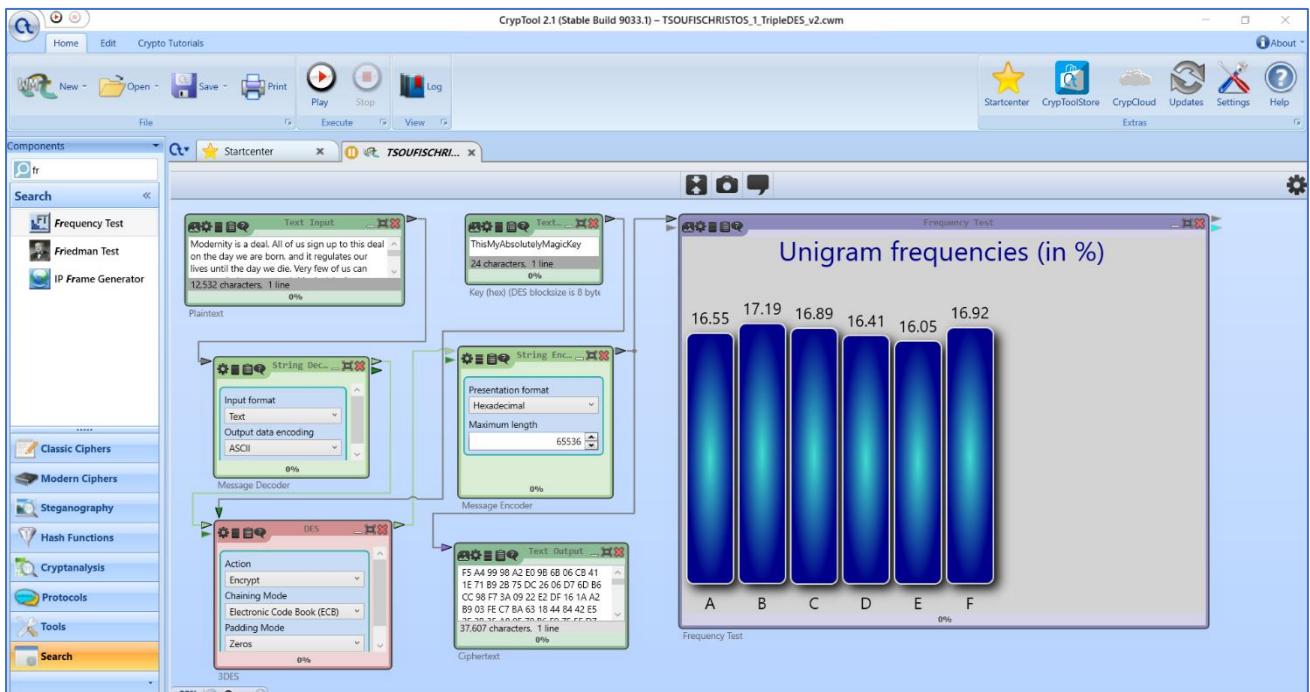
Παρακάτω φαίνεται μια ελαφρώς διαφορετική υλοποίηση του DES Cipher (πιθανώς να υπάρχει λάθος σε κάποιο component οπότε αποτυγχάνει η κρυπτογράφηση όπως φαίνεται στο output αλλά έχει προστεθεί ως παρατήρηση).



Για το ίδιο αρχείο, διορθώνοντας το λάθος, προκύπτει η παρακάτω εικόνα.



Και με ανεπτυγμένα τα components:



AES Cipher

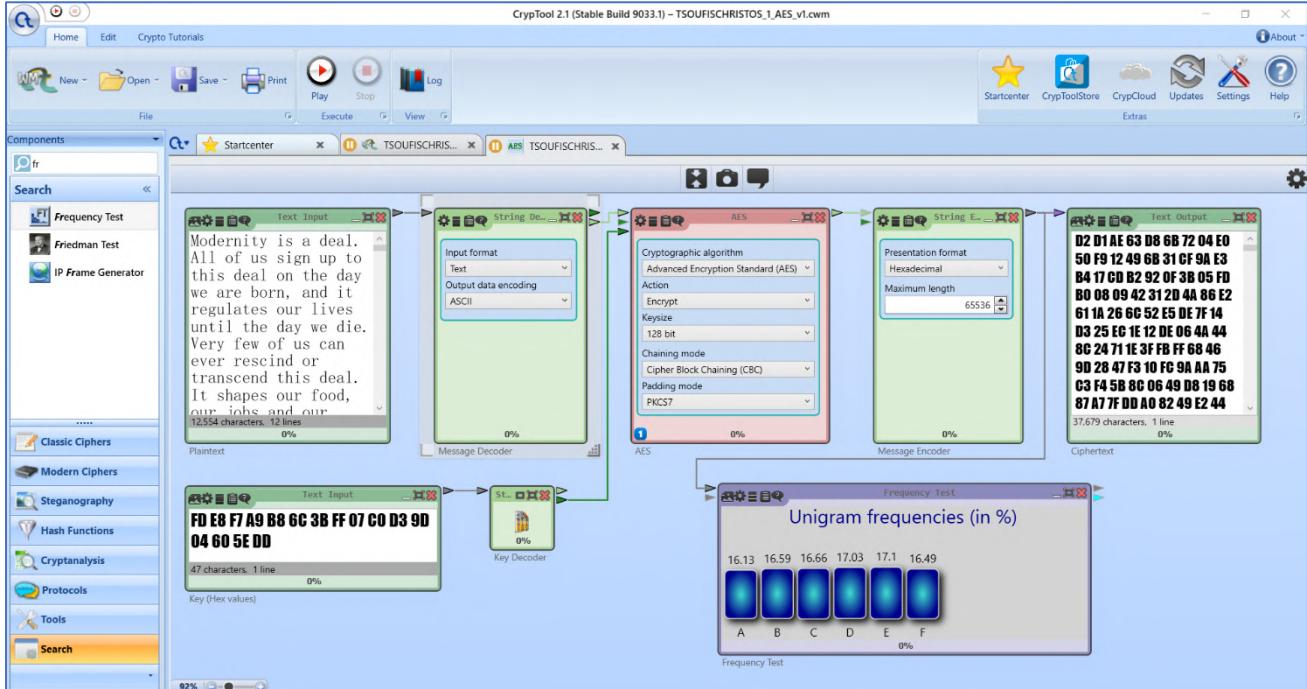
Ακολουθεί μια σύντομη αναφορά στον τρόπο λειτουργίας του αλγορίθμου κρυπτογράφησης AES. Ο AES Cipher είναι μια μέθοδος συμμετρικής κρυπτογράφησης. Χρησιμοποιείται κυρίως για κρυπτογράφηση και προστασία των δεδομένων. Χρησιμοποιείται έναντι του DES καθώς είναι πολύ γρηγορότερος και καλύτερος από τον DES. Αποτελείται από 3 block ciphers οι οποίοι χρησιμοποιούνται για την κρυπτογράφηση των δεδομένων. Αναπτύχθηκε στο NIST το 1997 και δημοσιεύτηκε το 2001 και έχει ως στόχο να ξεπεράσει τα κενά του DES. Τα χαρακτηριστικά του είναι ότι έχει keys 3 μηκών (128, 192, 256 bits). Είναι flexible και έχει υλοποίηση και για software αλλά και για hardware. Παρέχει υψηλή ασφάλεια και μπορεί να αποτρέψει πολλές επιθέσεις. Επιπλέον, δεν έχει copyright οπότε είναι open source και μπορεί να χρησιμοποιηθεί παγκοσμίως. Αποτελείται από 10 γύρους επεξεργασίας για 128 bit keys. Στα πλεονεκτήματά του, αξίζει να αναφερθεί ότι είναι ένας robust algorithm. Στα μειονεκτήματά του, συμπεριλαμβάνεται η δυσκολία στην υλοποίηση στο software αλλά ειδικά όταν απαιτείται υψηλό performance αλλά και οι πολλοί κύκλοι και η επεξεργασία που απαιτείται για την κρυπτογράφηση.

Με το template αυτό μπορεί να χρησιμοποιηθεί το AES component για την κρυπτογράφηση ενός αυθαίρετου text που δίνεται στο Plaintext component στα αριστερά. Το κρυπτογραφημένο text που προκύπτει, φαίνεται Ciphertext component στα δεξιά μετά το πάτημα του Play button. Το AES component λειτουργεί για δυαδικές τιμές (π.χ. bytes). Έτσι, το inputted text πρώτα μετατρέπεται σε bytes με το Message Decoder component. Υπό αυτές τις συνθήκες γίνεται interpreted σε ASCII. Τα bytes που προκύπτουν, κρυπτογραφούνται με AES, οπότε προκύπτει μια άλλη ακολουθία από bytes. Αυτά τα bytes μετά απλώς τυπώνονται ως δεκαεξαδικές τιμές με την βοήθεια του Message Encoder component.

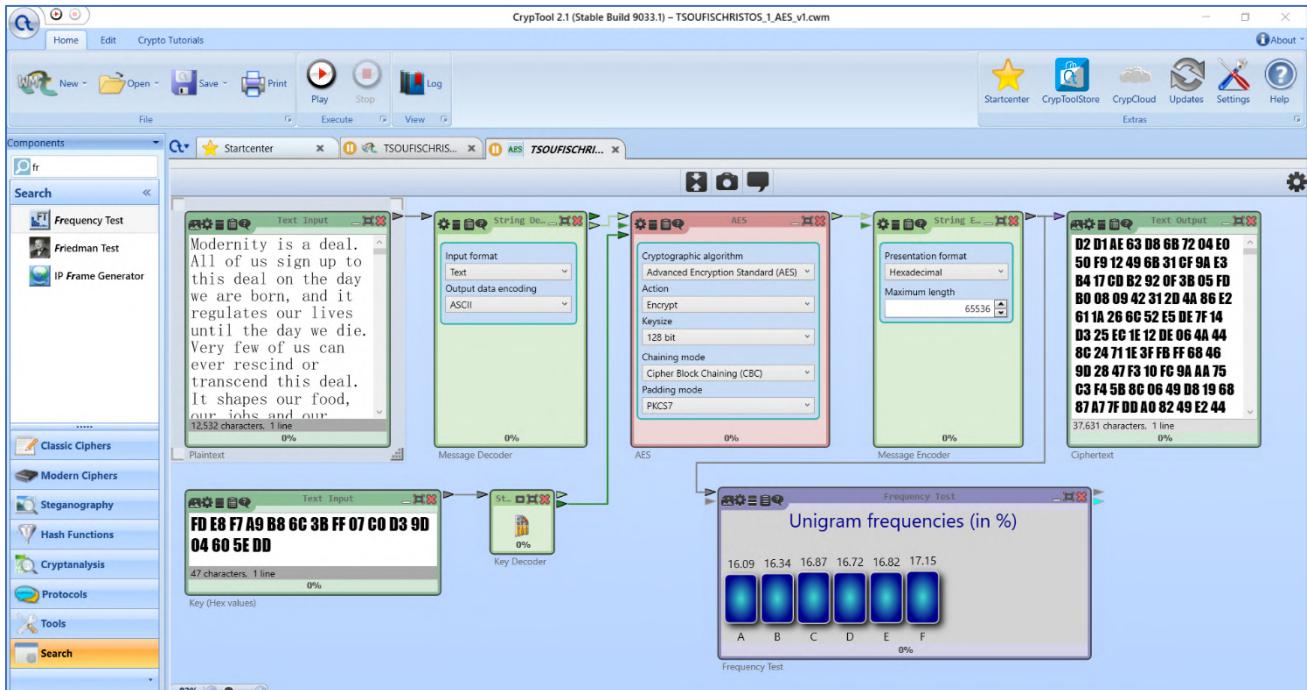
Σημειώνεται εδώ ότι, με αυτό το template γίνεται και η αποκρυπτογράφηση. Για να γίνει αυτό, πρώτα αντιγράφονται οι κρυπτογραφημένες δεκαεξαδικές τιμές στο Plaintext component. Υστερα, πρέπει να γίνουν οι εξής αλλαγές: (1) να τεθεί το Input format του Message Decoder σε Hexadecimal, (2) να τεθεί το Action του AES σε Decrypt και (3) να τεθεί το Format του Message Encoder σε Text και το encoding σε ASCII.

Έτσι, παρακάτω φαίνεται ο έτοιμος Cipher που δίνεται από το CrypTool.

Για το ίδιο input, χωρισμένο σε 12 lines.

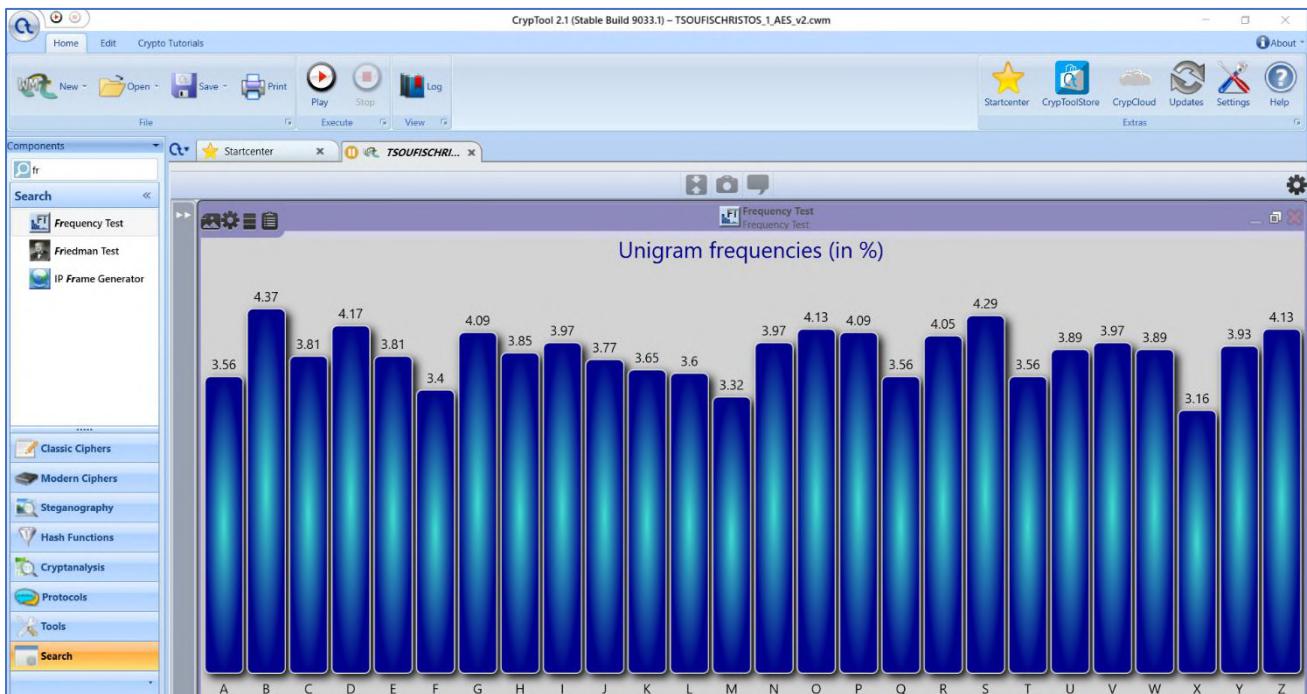
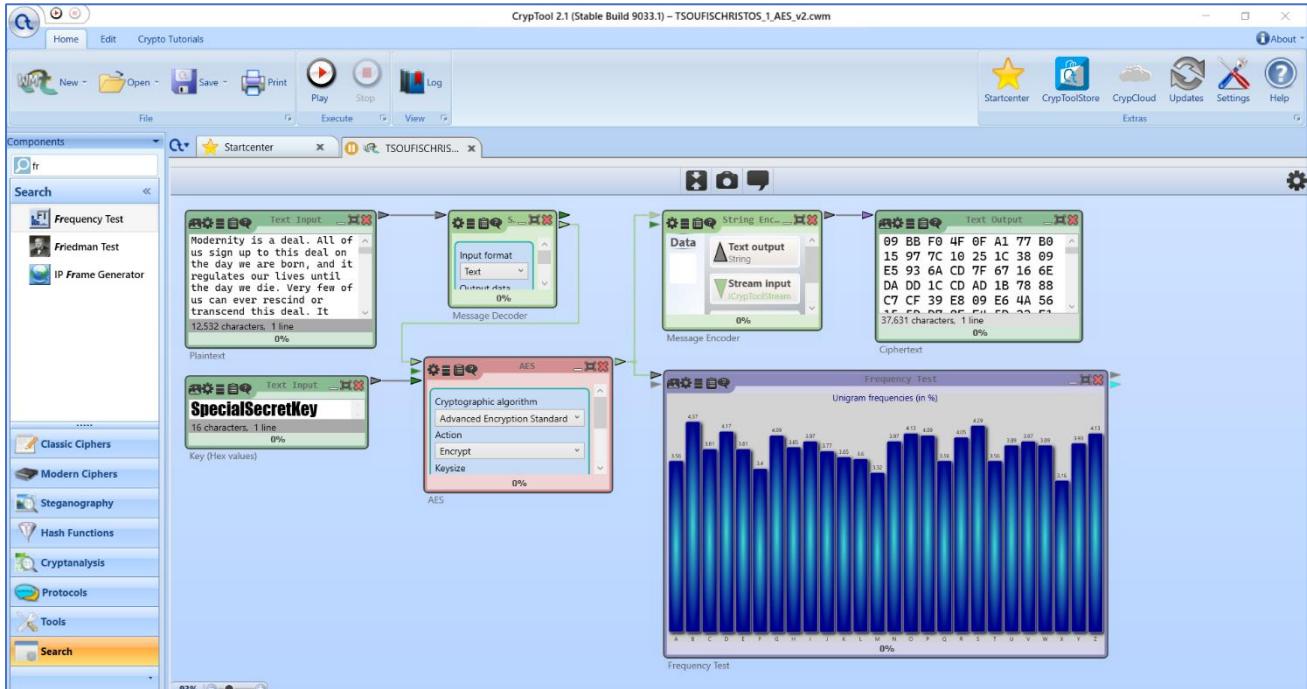


Για το ίδιο input, σε 1 line.

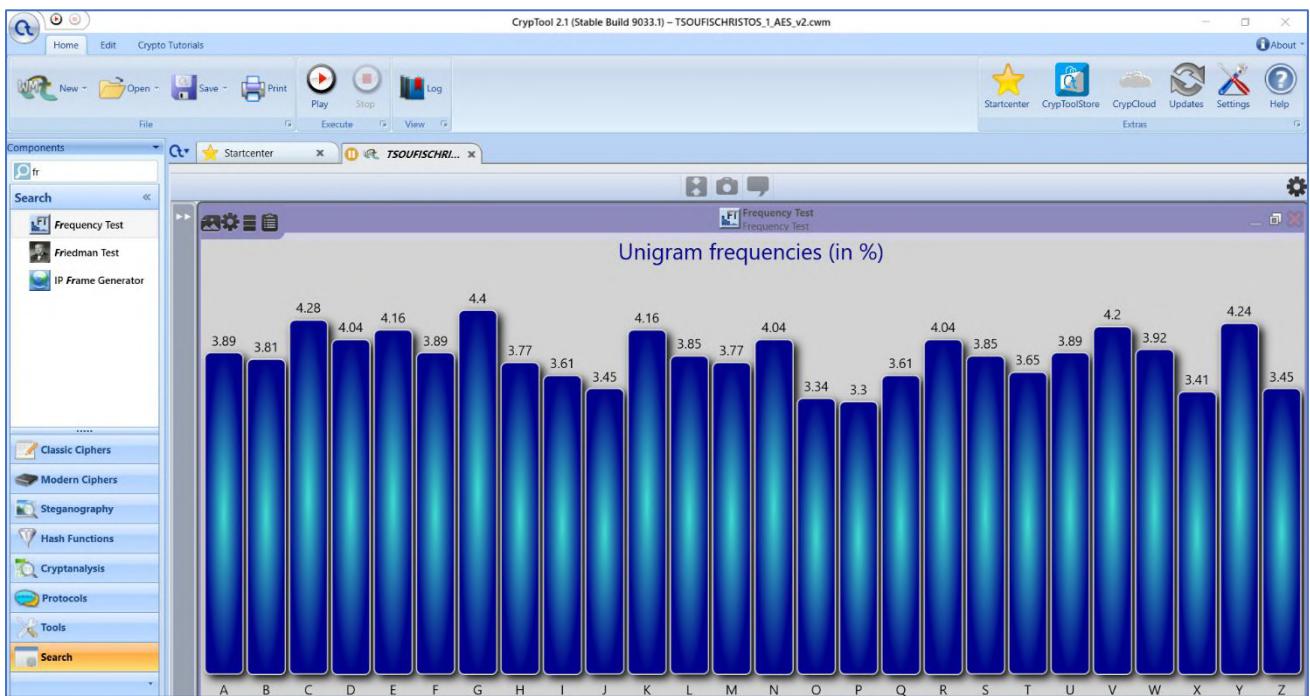
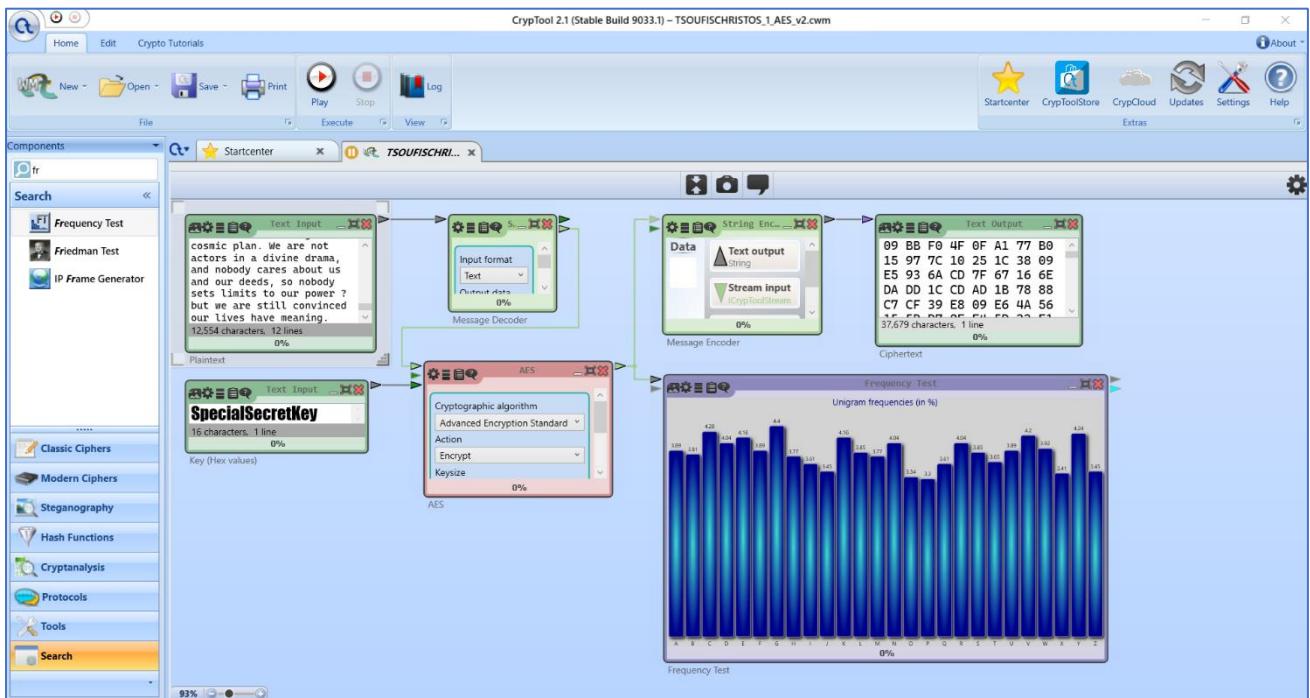


Παρακάτω φαίνεται μια ελαφρώς διαφορετική υλοποίηση του DES Cipher.

Για 1 line:



Για 12 lines:

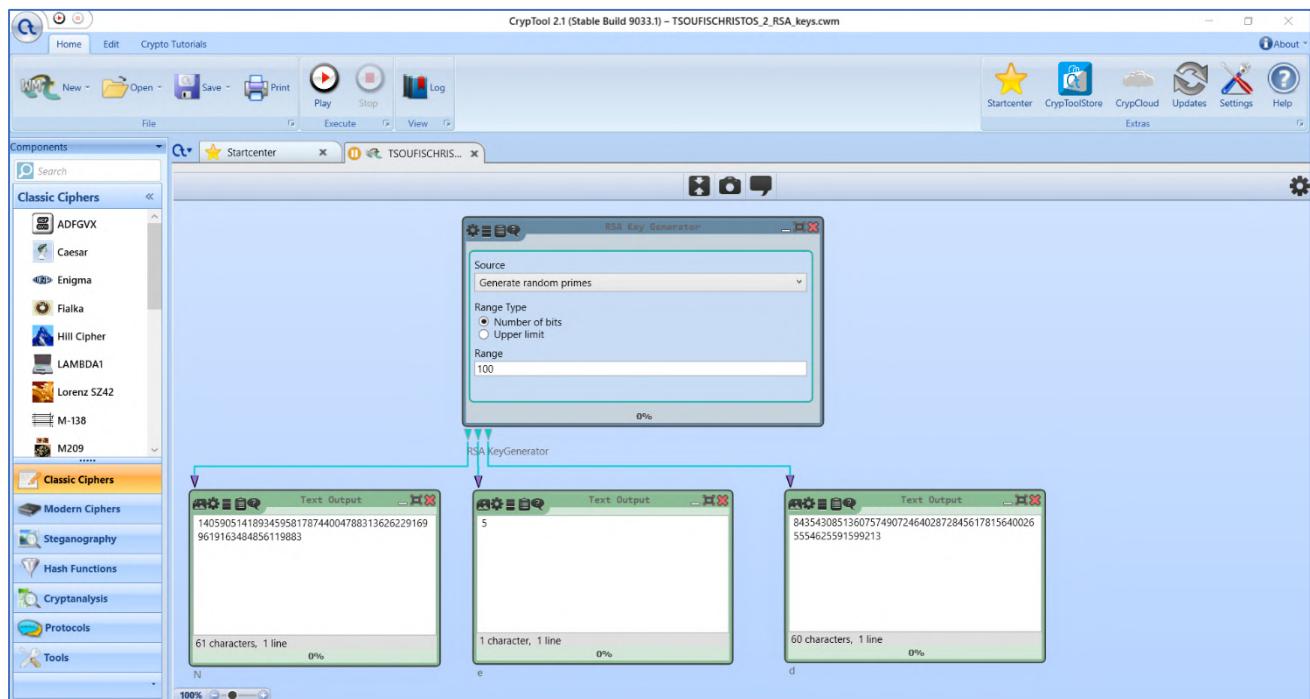


Σχολιασμός & Παρατηρήσεις:

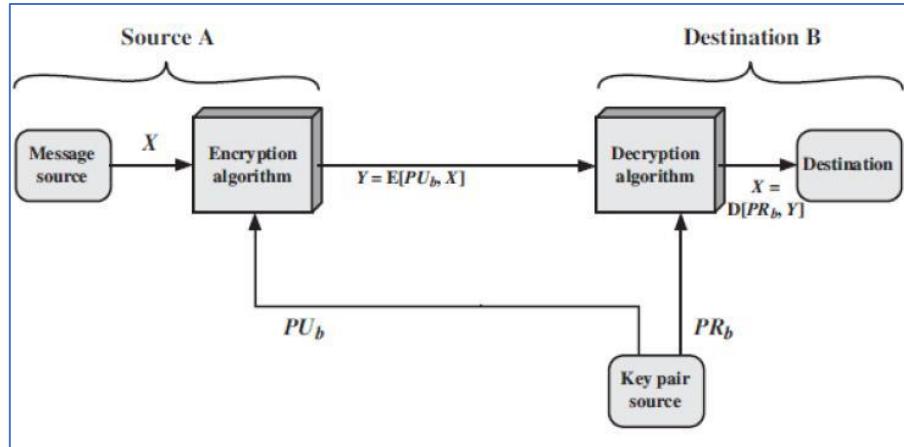
- Για τον **XOR Cipher**, παρατηρείται μια καλή κατανομή γραμμάτων, καθότι δεν παρατηρούνται μεγάλες διακυμάνσεις. Το αποτέλεσμα προκύπτει από XOR του plain text με ένα τυχαίο key (χρησιμοποιήθηκε μια ψευδο-τυχαία ακολουθία ώστε να ταιριάζει το μήκος του κλειδιού με το αρχικό κείμενο), και επειδή η κατανομή του τυχαίου κλειδιού είναι κανονική, το παραγόμενο κρυπτοκείμενο κανονικοποιείται. Συνεπώς, XOR Cipher έχει υψηλή αντοχή στην κρυπτανάλυση. Όμως, για την λειτουργία του απαιτείται ένα τεράστιο τυχαίο κλειδί, (μεγαλύτερο από το αρχικό κείμενο) γεγονός που αποτελεί πρόβλημα. Επιπλέον, λόγω της τυχαιότητας, είναι πολύ δύσκολο να παραχθεί τυχαίο κλειδί από ηλεκτρονικό υπολογιστή.
- Για τον **DES Cipher**, παρατηρείται μια πολύ ομοιόμορφη κατανομή, το οποίο σημαίνει ότι ο αλγόριθμος παρουσιάζει καλή αντοχή έναντι επιθέσεων κρυπτανάλυσης. Εδώ χρησιμοποιείται κάποιο τυχαίο key 8 Bytes, όμως μπορεί να “σπάσει” με σχετική ευκολία με brute force attack. Μια λύση σε αυτό το πρόβλημα θα ήταν η χρήση ενός μεγαλύτερου κλειδιού. Στα πλεονεκτήματα του αλγορίθμου, συγκαταλέγεται η αποδοτικότητά του για κρυπτογράφηση, αλλά και για αποκρυπτογράφηση. Αντιθέτως, στα μειονεκτήματα αξίζει να αναφερθεί το πρόβλημα του συμπληρώματος (δηλ. αν είναι γνωστό ένα κλειδί δεν αποκρυπτογραφεί το κείμενο τότε και το συμπλήρωμα του κλειδιού δεν το αποκρυπτογραφεί, γεγονός που μειώνει την ασφάλεια στο μισό). Επιπλέον, σημειώνεται πως υπάρχουν ειδικά κυκλώματα τα οποία μπορούν αποδοτικά να κάνουν brute force attack.
- Για τον **TripleDES Cipher**, παρατηρείται ότι παράγει κρυπτοκείμενο με πολύ καλή κατανομή και επομένως, έχει μεγάλη αντοχή στην κρυπτανάλυση. Εδώ χρησιμοποιείται ένα κλειδί 24 Bytes, ενώ υπάρχουν και κλειδιά των 16 ή 24 bits που είναι αρκετά για την αποφυγή μιας brute force attack. Σε γενικά πλαίσια, είναι ένας ασφαλής αλγόριθμος. Ωστόσο, υστερεί σε ταχύτητα καθώς απαιτεί αρκετούς υπολογιστικούς πόρους για την εκτέλεσή του.
- Τέλος, για τον **AES Cipher**, παρατηρείται μια επίσης καλή κατανομή μεταξύ γραμμάτων. Αυτό κάνει τον αλγόριθμο ιδιαίτερα ανθεκτικό σε επιθέσεις κρυπτανάλυσης. Ο AES Cipher αποτελεί έναν από τους ασφαλέστερους και αποδοτικότερους συμμετρικούς αλγορίθμους κρυπτογράφησης σήμερα και επειδή είναι και open source, χρησιμοποιείται παγκοσμίως. Εν τέλει, αξίζει να σημειωθεί ότι δεν είναι γνωστό κάποιο ενάλωτο σημείο του αλγορίθμου και μάλιστα, ενώ έχει αδιαμφισβήτητη αντοχή του σε brute force attacks.

2. Με χρήση του Cryptool και του RSA αλγορίθμου που διαθέτει, να παραχθεί ζεύγος κλειδιών για τους χρήστες *a* και *b* κάνοντας χρήση του RSA Key Generator block.

Ο RSA algorithm είναι ένας asymmetric αλγόριθμος κρυπτογράφησης. Asymmetric σημαίνει ότι λειτουργεί για δύο διαφορετικά keys (π.χ. public key (που φαίνεται σε όλους) & private key (που είναι ιδιωτικό)). Η ιδέα του RSA βασίζεται στο γεγονός ότι είναι δύσκολη η παραγοντοποίηση ενός μεγάλου ακεραίου. Το public key, αποτελείται από 2 αριθμούς όπου ο 1^{ος} είναι πολλαπλάσιο δυο μεγάλων πρώτων αριθμών. Έτσι, αν κάποιος παραγοντοποιήσει τον μεγάλο αριθμό, το private key αποκαλύπτεται. Επομένως, η δύναμη της κρυπτογράφησης βασίζεται στο μέγεθος του key και αν διπλασιαστεί ή τριπλασιαστεί το μέγεθος, η δύναμη της κρυπτογράφησης αυξάνεται εκθετικά. Τα RSA keys μπορεί να είναι 1024 ή 2048 bits αλλά οι ειδικοί θεωρούν ότι τα 1024 bits keys είναι πιθανό να “σπάσουν” στο μέλλον.

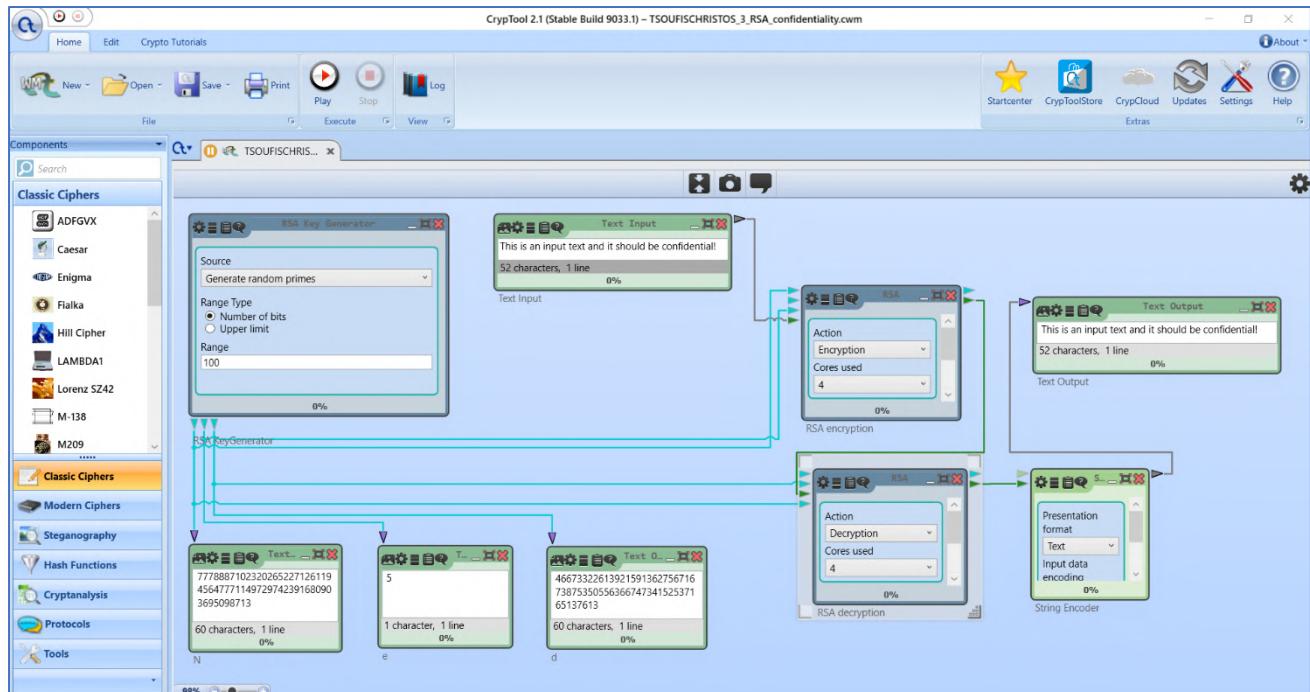


3. Να υλοποιηθεί με το CrypTool διάταξη για την προστασία της εμπιστευτικότητας (confidentiality) μηνύματος της επιλογής σας σύμφωνα με το ακόλουθο σχήμα:



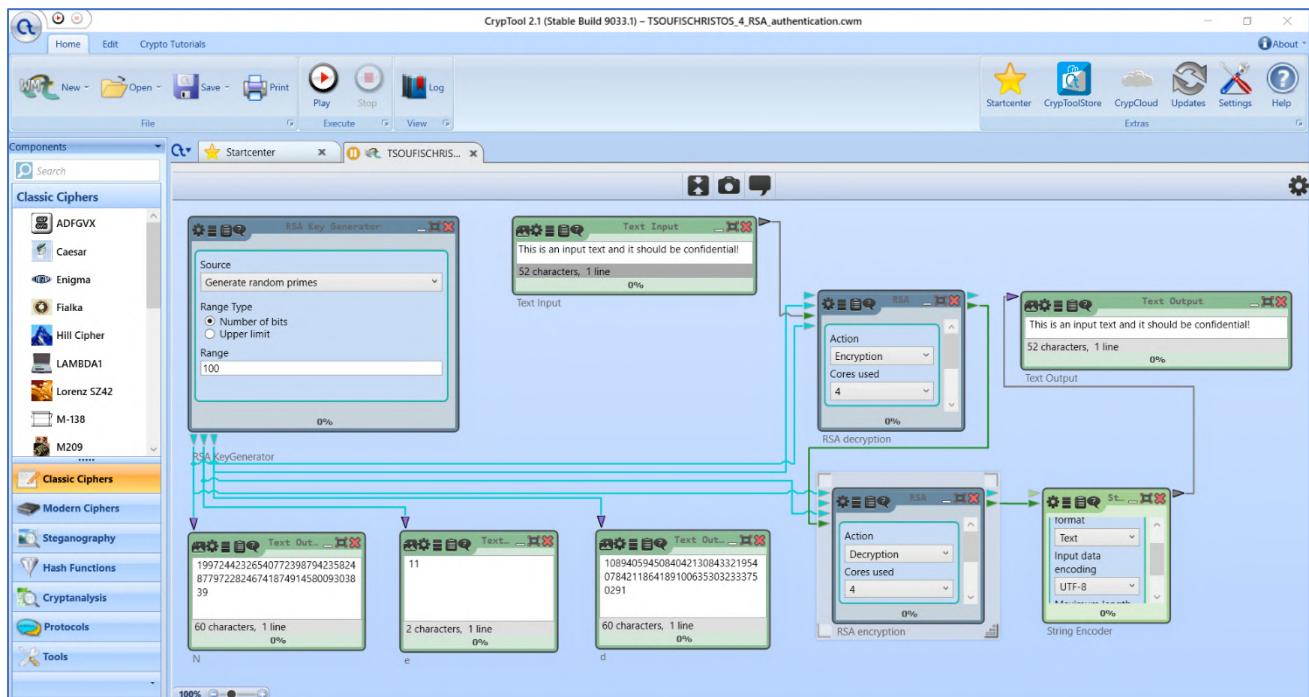
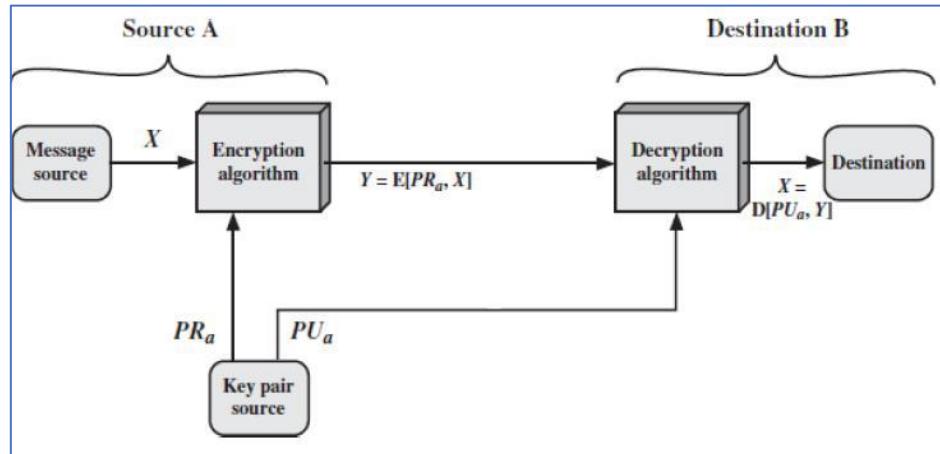
Ο χρήστης α κρυπτογραφεί το μήνυμα X με το *public key* του παραλήπτη b . Στη συνέχεια, ο παραλήπτης χρησιμοποιεί το *private key* του για την αποκρυπτογράφηση του μηνύματος.

$$Y = E(PUb, X), X = D(PRb, Y)$$



4. Να υλοποιηθεί με το *CrypTool* διάταξη για την πιστοποίηση της ταυτότητας (*authentication*) κάποιας οντότητας (π.χ., χρήστης) αλλά και για την διασφάλιση ότι το μήνυμα δεν έχει υποστεί αλλαγές (*integrity*), όπου ο χρήστης α κρυπτογραφεί το μήνυμα X με το *private key* του. Στη συνέχεια, ο παραλήπτης χρησιμοποιεί το *public key* του α για την αποκρυπτογράφηση του. Επιτυχημένη αποκρυπτογράφηση πιστοποιεί ότι όντως έχει κρυπτογραφηθεί με το *private key* που γνωρίζει μόνο ο χρήστης α.

$$Y = E(PR_a, X), X = D(PU_a, Y)$$



Σχολιασμός & Παρατηρήσεις:

Παρατηρείται πως παραπάνω διάταξη για authentication είναι πολύ παρόμοια με την διάταξη του confidentiality με μόνη αλλαγή την είσοδο του private key από τον παραλήπτη στον αποστολέα.

5. Συγκέντρωση Πληροφοριών και Ανίχνευση Αδυναμιών σε Δίκτυα Υπολογιστών

Το κομμάτι αυτό της εργασίας ασχολείται με εργαλεία συγκέντρωσης πληροφοριών και ανίχνευσης αδυναμιών σε δίκτυα υπολογιστών. Θα χρειαστεί να εγκατασταθούν τα εργαλεία: Wireshark, Nmap.

Ζητούνται τα ακόλουθα:

1. Σε ένα παράθυρο εντολών και με χρήση κατάλληλης σύνταξης της εντολής nslookup να αναζητηθούν οι ακόλουθες πληροφορίες για κάποιο domain name (π.χ. ntua.gr): οι εξυπηρετητές ονοματολογίας (NS εγγραφές) και εξυπηρετητές ηλεκτρονικού ταχυδρομείου (MX εγγραφές) και να καταγραφούν οι IP διευθύνσεις αυτών. Στο αρχείο που θα υποβληθεί, να υπάρχουν screenshots όπου θα φαίνονται οι εντολές και οι απαντήσεις.

Χρησιμοποιώντας την εντολή nslookup -querytype=ns nasa.gov προκύπτει για εξυπηρετητή ονοματολογίας (ενδεικτικά μαζί με μερικές IP addresses):

```
C:\Users\Chris Tsoufis>nslookup -querytype=ns nasa.gov
Server: speedport-entry-2i.ote.gr
Address: fe80::1

Non-authoritative answer:
nasa.gov      nameserver = a8-66.akam.net
nasa.gov      nameserver = a1-32.akam.net
nasa.gov      nameserver = a9-64.akam.net
nasa.gov      nameserver = a12-64.akam.net
nasa.gov      nameserver = a5-66.akam.net
nasa.gov      nameserver = a14-67.akam.net

C:\Users\Chris Tsoufis>nslookup a8-66.akam.net
Server: speedport-entry-2i.ote.gr
Address: fe80::1

Non-authoritative answer:
Name:   a8-66.akam.net
Addresses: 2600:1403:a::42
          2.16.40.66

C:\Users\Chris Tsoufis>nslookup a1-32.akam.net
Server: speedport-entry-2i.ote.gr
Address: fe80::1

Non-authoritative answer:
Name:   a1-32.akam.net
Addresses: 2600:1401:2::20
          193.108.91.32
```

Τρέχοντας την ίδια εντολή από διαφορετικό PC προκύπτει η ακόλουθη εικόνα.

```
C:\Users\      nslookup -querytype=ns ece.ntua.gr
Server:
Address: 192.168.2.1

Non-authoritative answer:
ece.ntua.gr    nameserver = achilles.noc.ntua.gr
ece.ntua.gr    nameserver = ulysses.noc.ntua.gr
ece.ntua.gr    nameserver = diomedes.noc.ntua.gr

ulysses.noc.ntua.gr    internet address = 147.102.222.230
ulysses.noc.ntua.gr    AAAA IPv6 address = 2001:648:2000:de::230
achilles.noc.ntua.gr   internet address = 147.102.222.210
achilles.noc.ntua.gr   AAAA IPv6 address = 2001:648:2000:de::210
diomedes.noc.ntua.gr  internet address = 147.102.222.220
diomedes.noc.ntua.gr  AAAA IPv6 address = 2001:648:2000:de::220
```

Χρησιμοποιώντας την εντολή nslookup -querytype=mx nasa.gov προκύπτει για εξυπηρετητή ηλεκτρονικού ταχυδρομείου (ενδεικτικά μαζί με μερικές IP addresses):

```
Select Command Prompt

C:\Users\Chris Tsoufis>nslookup -querytype=mx nasa.gov
Server: speedport-entry-2i.ote.gr
Address: fe80::1

Non-authoritative answer:
nasa.gov      MX preference = 10, mail exchanger = ndmsvnpf103.ndc.nasa.gov
nasa.gov      MX preference = 10, mail exchanger = ndjsvnpf103.ndc.nasa.gov
nasa.gov      MX preference = 10, mail exchanger = ndmsvnpf102.ndc.nasa.gov
nasa.gov      MX preference = 10, mail exchanger = ndmsvnpf104.ndc.nasa.gov
nasa.gov      MX preference = 10, mail exchanger = ndmsvnpf101.ndc.nasa.gov
nasa.gov      MX preference = 10, mail exchanger = ndjsvnpf104.ndc.nasa.gov
nasa.gov      MX preference = 10, mail exchanger = ndjsvnpf101.ndc.nasa.gov
nasa.gov      MX preference = 10, mail exchanger = ndjsvnpf102.ndc.nasa.gov

C:\Users\Chris Tsoufis>nslookup ndmsvnpf103.ndc.nasa.gov
Server: speedport-entry-2i.ote.gr
Address: fe80::1

Non-authoritative answer:
Name: ndmsvnpf103.ndc.nasa.gov
Addresses: 2001:4d0:8302:1100::153
          198.117.0.153

C:\Users\Chris Tsoufis>nslookup ndjsvnpf103.ndc.nasa.gov
Server: speedport-entry-2i.ote.gr
Address: fe80::1

Non-authoritative answer:
Name: ndjsvnpf103.ndc.nasa.gov
Addresses: 2001:4d0:a302:1100::153
          198.117.1.153
```

Τρέχοντας την ίδια εντολή από διαφορετικό PC προκύπτει η ακόλουθη εικόνα.

```
C:\Users\      nslookup -querytype=mx ece.ntua.gr
Server:
Address: 192.168.2.1

Non-authoritative answer:
ece.ntua.gr    MX preference = 100, mail exchanger = achilles.noc.ntua.gr
ece.ntua.gr    MX preference = 100, mail exchanger = diomedes.noc.ntua.gr
ece.ntua.gr    MX preference = 100, mail exchanger = ulysses.noc.ntua.gr

ece.ntua.gr    nameserver = ulysses.noc.ntua.gr
ece.ntua.gr    nameserver = diomedes.noc.ntua.gr
ece.ntua.gr    nameserver = achilles.noc.ntua.gr
ulysses.noc.ntua.gr   internet address = 147.102.222.230
ulysses.noc.ntua.gr   AAAA IPv6 address = 2001:648:2000:de::230
achilles.noc.ntua.gr   internet address = 147.102.222.210
achilles.noc.ntua.gr   AAAA IPv6 address = 2001:648:2000:de::210
diomedes.noc.ntua.gr   internet address = 147.102.222.220
```

2. Να γίνει αρχή μιας καταγραφής μέσα από το Wireshark και να εκτελεστούν διαδοχικά ping και tracert/traceroute σε κάποιο προορισμό. Να γίνει περιγραφή των τρόπου λειτουργίας των εντολών αυτών. Στο αρχείο που θα υποβληθεί, να υπάρχουν τα screenshots από το Wireshark.

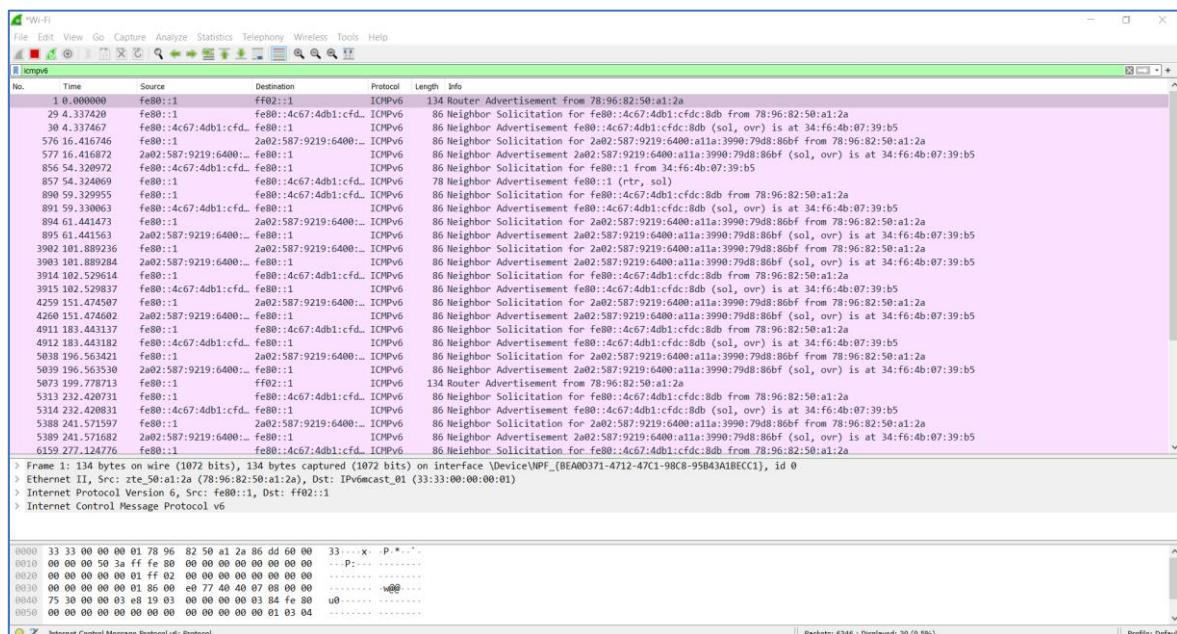
ping: Είναι μέθοδος για τον εντοπισμό της διαθεσιμότητας και της απόδοσης ενός απομακρυσμένου πόρου του δικτύου. Αποτελεί διαδικασία κατά την οποία επιβεβαιώνεται η σύνδεση με έναν απομακρυσμένο υπολογιστή στο διαδίκτυο. Με το ping αποστέλλεται στον απομακρυσμένο υπολογιστή ένα πακέτο δεδομένων και στη συνέχεια ο υπολογιστής που έστειλε το πακέτο, περιμένει για μία echo reply, δηλαδή την απάντηση στο πακέτο δεδομένων του ping (πολλοί το αποκαλούν και pong). Το πακέτο που αποστέλλεται με το ping ονομάζεται ICMP (Internet Control Message Protocol) echo packet. Το διάστημα μεταξύ του ping και του echo reply επιβεβαιώνει την ποιότητα της σύνδεσης και αποκαλείται lag (καθυστέρηση). Αν ο αποστολέας δε βρεθεί είτε θα σταλεί κάποιο μήνυμα σφάλματος από κάποιον ενδιάμεσο κόμβο, ή το χρονόμετρο που θέτει ο αποστολέας θα λήξει, και το πακέτο θα θεωρείται άκυρο (ακόμα και αν ληφθεί εκ των υστέρων). Συνήθως πάντως δε στέλνεται μόνο ένα πακέτο, αλλά μία δέσμη από αυτά.

```
C:\Users\Chris Tsoufis>ping ntua.gr

Pinging ntua.gr [2001:648:2000:de::210] with 32 bytes of data:
Reply from 2001:648:2000:de::210: time=10ms
Reply from 2001:648:2000:de::210: time=11ms
Reply from 2001:648:2000:de::210: time=32ms
Reply from 2001:648:2000:de::210: time=10ms

Ping statistics for 2001:648:2000:de::210:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 10ms, Maximum = 32ms, Average = 15ms

C:\Users\Chris Tsoufis>
```



Σχολιασμός: Για την καταγραφή, χρησιμοποιήθηκε το φίλτρο icmpv6 και έτσι επιβεβαιώνεται ότι στάλθηκαν 4 πακέτα ping και λήφθηκαν αντίστοιχες απαντήσεις, γεγονός που είναι σε συμφωνία και με την απάντηση στο cmd.

Τρέχοντας από διαφορετικό PC, προκύπτει ότι:

```
C:\Users\      ping ece.ntua.gr

Pinging ece.ntua.gr [147.102.222.195] with 32 bytes of data:
Reply from 147.102.222.195: bytes=32 time=18ms TTL=58
Reply from 147.102.222.195: bytes=32 time=22ms TTL=58
Reply from 147.102.222.195: bytes=32 time=391ms TTL=58
Reply from 147.102.222.195: bytes=32 time=19ms TTL=58

Ping statistics for 147.102.222.195:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 18ms, Maximum = 391ms, Average = 112ms
```

The screenshot shows the Wireshark interface with a single captured frame selected. The frame details are as follows:

No.	Time	Source	Destination	Protocol	Length	Info
29	3.844219	192.168.2.9	147.102.222.195	ICMP	74	Echo (ping) request id=0x0001, seq=11/2816, ttl=128 (reply in 30)
30	3.862305	147.102.222.195	192.168.2.9	ICMP	74	Echo (ping) reply id=0x0001, seq=11/2816, ttl=58 (request in 29)
33	4.864435	192.168.2.9	147.102.222.195	ICMP	74	Echo (ping) request id=0x0001, seq=12/3072, ttl=128 (reply in 35)
35	4.886830	147.102.222.195	192.168.2.9	ICMP	74	Echo (ping) reply id=0x0001, seq=12/3072, ttl=58 (request in 33)
44	5.889405	192.168.2.9	147.102.222.195	ICMP	74	Echo (ping) request id=0x0001, seq=13/3328, ttl=128 (reply in 46)
46	6.288523	147.102.222.195	192.168.2.9	ICMP	74	Echo (ping) reply id=0x0001, seq=13/3328, ttl=58 (request in 44)
62	6.904293	192.168.2.9	147.102.222.195	ICMP	74	Echo (ping) request id=0x0001, seq=14/3584, ttl=128 (reply in 63)
63	6.923131	147.102.222.195	192.168.2.9	ICMP	74	Echo (ping) reply id=0x0001, seq=14/3584, ttl=58 (request in 62)

Frame details:

- > Frame 29: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{3917DD20-3003-43F7-A409-7CAEC8BC6A9F}, id 0
- > Ethernet II, Src: IntelCor_a4:4f:3e (78:0c:b8:a4:4f:3e), Dst: DlwnetTec_10:74:d0 (e4:26:86:10:74:d0)
- > Internet Protocol Version 4, Src: 192.168.2.9, Dst: 147.102.222.195
- > Internet Control Message Protocol

traceroute/tracert: Είναι διαγνωστικές εντολές για Δίκτυα Υπολογιστών και δείχνουν πιθανά routes (paths) και μετράνε transit καθυστερήσεις πακέτων πάνω σε IP δίκτυα. Ουσιαστικά, πρόκειται για ένα πιο “σύνθετο” ping. Με το tracert εντοπίζονται όλα τα ενδιάμεσα hops ως τον προορισμό. Με άλλα λόγια, βρίσκει τη διαδρομή που θα ακολουθήσει ο υπολογιστής μας μέσα στο διαδίκτυο για να συνδεθεί σε κάποιον άλλον υπολογιστή. Μάλιστα, αυτό γίνεται με τη διαδοχική αποστολή πακέτων με ανξανόμενο TTL (time to live), αφού το πρώτο πακέτο που στέλνεται έχει TTL = 1, οπότε θα φτάσει μέχρι τον πρώτο κόμβο και θα σταλεί πίσω στον αποστολέα ένα πακέτο «time to live exceeded» κ.ο.κ. μέχρι τον τελικό προορισμό.

```
Windows Command Prompt
C:\Users\Chris Tsoufis>tracert ntua.gr

Tracing route to ntua.gr [2001:648:2000:de::210]
over a maximum of 30 hops:

  1  *           *           *      Request timed out.
  2  *           *           *      Request timed out.
  3  35 ms       8 ms        7 ms  2a02:580:50da:437:: Request timed out.
  4  *           *           *      Request timed out.
  5  9 ms        10 ms       9 ms  grnet.gr-ix.gr [2001:7f8:6e::1]
  6  15 ms       10 ms       10 ms kolettir-eier-AE.backbone.grnet.gr [2001:648:2ff2:101::1]
  7  17 ms       12 ms       15 ms ntua-zogr-2.kolettir.access-link.grnet.gr [2001:648:2ffd:3323:2::2]
  8  12 ms       11 ms       12 ms achilles.noc.ntua.gr [2001:648:2000:de::210]

Trace complete.

C:\Users\Chris Tsoufis>
```

No.	Time	Source	Destination	Protocol	Length	Info
21,535,691	192.168.1.26	239.255.255.250	UDP	77	53063 + 15600 Len:35	
33 4.505,150	192.168.1.26	192.168.1.255	UDP	77	38136 + 15600 Len:35	
34 5.609,940	fe80::4c67:4db1:cfd.. fe80::1	DNS	91	Standard query 0x670f A www.ntua.gr		
35 5.601,014	fe80::4c67:4db1:cfd.. fe80::1	DNS	97	Standard query 0x072e A fonts.gstatic.com		
36 5.601,026	fe80::4c67:4db1:cfd.. fe80::1	DNS	100	Standard query 0x066a A fonts.googleapis.com		
37 5.601,432	fe80::4c67:4db1:cfd.. fe80::1	DNS	97	Standard query 0x43fa AAAA fonts.gstatic.com		
38 5.601,448	fe80::4c67:4db1:cfd.. fe80::1	DNS	91	Standard query 0x3225 AAAA www.ntua.gr		
39 5.601,497	fe80::4c67:4db1:cfd.. fe80::1	DNS	100	Standard query 0x490f AAAA fonts.googleapis.com		
41 5.603,178	fe80::4c67:4db1:cfd.. fe80::1	DNS	103	Standard query 0x0cb0 A safebrowsing.google.com		
42 5.603,632	fe80::4c67:4db1:cfd.. fe80::1	DNS	103	Standard query 0x2336 AAAA safebrowsing.google.com		
43 5.612,707	fe80::1	fe80::4c67:4db1:cfd.. DNS	107	Standard query response 0x670f A www.ntua.gr A 147.102.224.101		
44 5.617,333	fe80::1	fe80::4c67:4db1:cfd.. DNS	149	Standard query response 0x072e A fonts.gstatic.com CNAME gstaticssl.l.google.com A 172.217.16.131		
45 5.621,253	fe80::1	fe80::4c67:4db1:cfd.. DNS	116	Standard query response 0x066a A fonts.googleapis.com A 172.217.18.106		
46 5.627,322	fe80::1	fe80::4c67:4db1:cfd.. DNS	119	Standard query response 0x3225 AAAA www.ntua.gr AAAA 2001:648:2000:329::101		
49 5.631,179	fe80::1	fe80::4c67:4db1:cfd.. DNS	175	Standard query response 0x43fa AAAA fonts.gstatic.com CNAME gstaticcdsl.l.google.com AAAA 2a00:1450:4001:809::2003		
50 5.633,546	2a02:587:9219:6400..	2a00:1450:4001:809.. QUIC	1392	Initial, DCID=0c6747ec77d6fa43, PKN: 1, CRYPTO, PADDING		
51 5.637,413	fe80::1	fe80::4c67:4db1:cfd.. DNS	134	Standard query response 0x490f AAAA fonts.googleapis.com AAAA 2a00:1450:4001:813::2004		
52 5.638,695	2a02:587:9219:6400..	2a00:1450:4001:813.. QUIC	1392	Initial, DCID=59548Ff210602811, PKN: 1, CRYPTO, PADDING		
53 5.640,706	fe80::1	fe80::4c67:4db1:cfd.. DNS	151	Standard query response 0x0cb0 A safebrowsing.google.com CNAME sb.l.google.com A 216.58.212.142		
54 5.640,706	fe80::1	fe80::4c67:4db1:cfd.. DNS	150	Standard query response 0x2336 AAAA safebrowsing.google.com CNAME sb.l.google.com AAAA 2a00:1450:4001:829::200e		
55 5.641,545	2a02:587:9219:6400..	2a00:1450:4001:829.. QUIC	1392	Initial, DCID=5fa2754481072635, PKN: 1, CRYPTO, PADDING		
82 5.688,077	2a00:1450:4001:813..	2a02:587:9219:6400.. QUIC	1392	Initial, SCID=59548Ff210602811, PKN: 1, ACK, CRYPTO, PADDING		
83 5.690,067	2a02:587:9219:6400..	2a00:1450:4001:813.. QUIC	1392	Initial, DCID=59548Ff210602811, PKN: 2, ACK, PADDING		
84 5.690,319	2a00:1450:4001:809..	2a02:587:9219:6400.. QUIC	1392	Protected Payload (KPO)		
85 5.691,202	2a02:587:9219:6400..	2a00:1450:4001:809.. QUIC	220	Protected Payload (KPO), DCID=0c6747ec77d6fa43		
86 5.693,411	2a00:1450:4001:829..	2a02:587:9219:6400.. QUIC	1392	Protected Payload (KPO)		
87 5.694,154	2a02:587:9219:6400..	2a00:1450:4001:829.. QUIC	216	Protected Payload (KPO).. DCID=5fa2754481072635		

Frame 2: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface 'DeviceNPF_{BEAD0371-4712-47C1-98C8-95B43A1BECC1}', id 0

> Ethernet II, Src: Samsung_E3:eb:12 (00:7c:2d:f3:eb:12), Dst: IPv4 broadcast (ff:ff:ff:ff:ff:ff) (01:00:5e:ff:ff:ff)

> Internet Protocol Version 4, Src: 192.168.1.26, Dst: 239.255.255.250

> User Datagram Protocol, Src Port: 53063, Dst Port: 15600

> Data (35 bytes)

Σχολιασμός: Για την καταγραφή, χρησιμοποιήθηκε το φίλτρο udp και έτσι επιβεβαιώνονται τα αποτελέσματα τόσο από το cmd όσο και από το Wireshark.

Τρέχοντας από διαφορετικό PC, προκύπτει ότι:

```
C:\Users\      >tracert ece.ntua.gr

Tracing route to ece.ntua.gr [147.102.222.196]
over a maximum of 30 hops:

 1      1 ms      1 ms      1 ms  vodafone.station [192.168.2.1]
 2      *      52 ms     21 ms  loopback2004.med01.dsl.hol.gr [62.38.0.170]
 3     19 ms     17 ms     16 ms  ppp062038028201.dsl.hol.gr [62.38.28.201]
 4     17 ms     19 ms     17 ms  62.38.96.150
 5   1204 ms     77 ms     22 ms  grnet.gr-ix.gr [176.126.38.1]
 6     22 ms     19 ms     18 ms  ntua-zogr-3.eier.access-link.grnet.gr [62.217.96.169]
 7     40 ms     20 ms     18 ms  f1.mail.ntua.gr [147.102.222.196]

Trace complete.
```

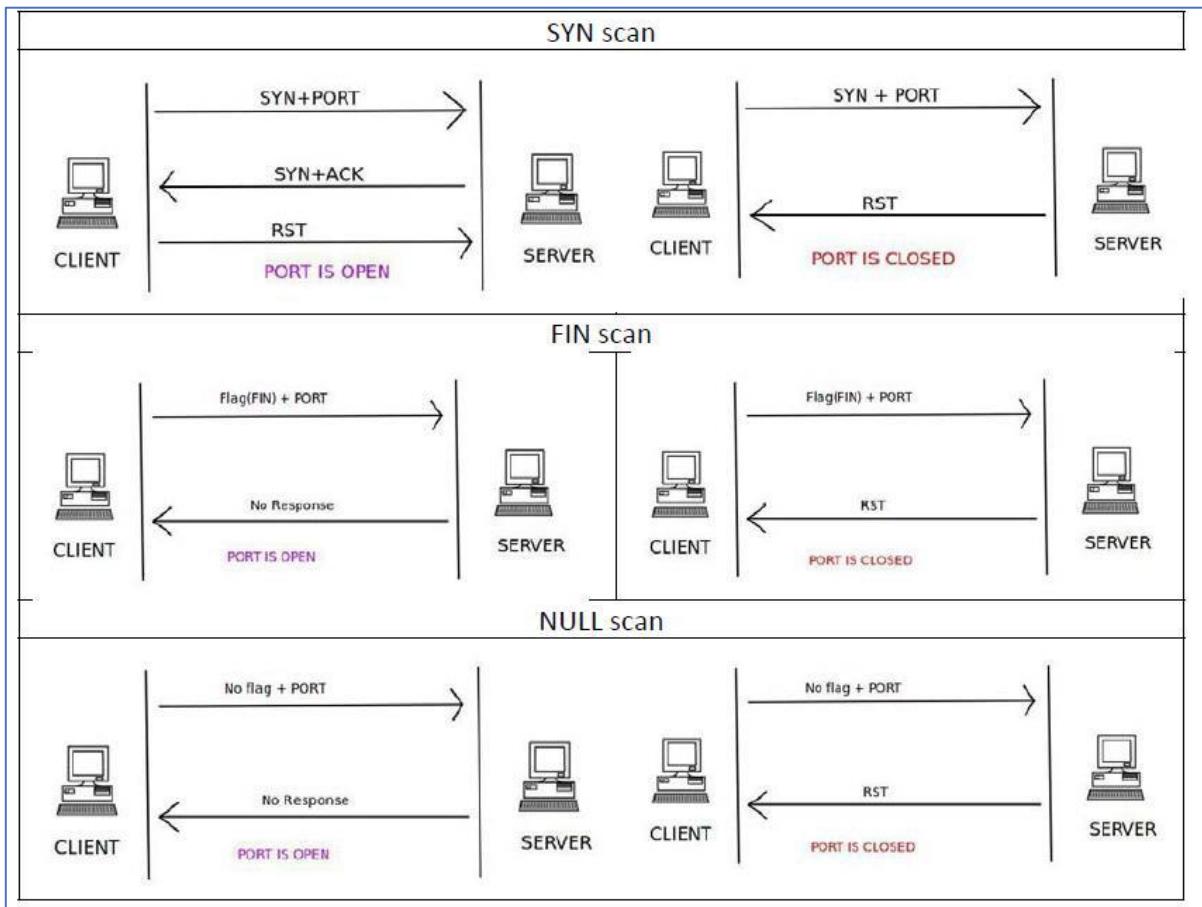
Wi-Fi						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
icmp						
No.	Time	Source	Destination	Protocol	Length	Info
148	15.668582	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=15/3840, ttl=1 (no response found!)
149	15.670378	192.168.2.1	192.168.2.9	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
150	15.671285	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=16/4096, ttl=1 (no response found!)
151	15.672221	192.168.2.1	192.168.2.9	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
152	15.673161	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=17/4352, ttl=1 (no response found!)
153	15.674169	192.168.2.1	192.168.2.9	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
175	16.705666	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=18/4608, ttl=2 (no response found!)
219	20.247054	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=19/4864, ttl=2 (no response found!)
222	20.298761	62.38.0.170	192.168.2.9	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
223	20.301041	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=20/5120, ttl=2 (no response found!)
225	20.322537	62.38.0.170	192.168.2.9	ICMP	134	Time-to-live exceeded (Time to live exceeded in transit)
230	21.341106	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=21/5376, ttl=3 (no response found!)
231	21.360002	62.38.28.201	192.168.2.9	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
232	21.362891	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=22/5632, ttl=3 (no response found!)
233	21.379618	62.38.28.201	192.168.2.9	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
234	21.383120	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=23/5888, ttl=3 (no response found!)
235	21.399675	62.38.28.201	192.168.2.9	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
240	22.434268	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=24/6144, ttl=4 (no response found!)
241	22.451259	62.38.96.150	192.168.2.9	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
242	22.457446	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=25/6400, ttl=4 (no response found!)
243	22.473980	62.38.96.150	192.168.2.9	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
244	22.476955	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=26/6656, ttl=4 (no response found!)
245	22.494386	62.38.96.150	192.168.2.9	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
255	22.086464	62.38.96.150	192.168.2.9	ICMP	110	Destination unreachable (Port unreachable)
286	25.596944	62.38.96.150	192.168.2.9	ICMP	110	Destination unreachable (Port unreachable)
						> Frame 148: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface \Device\NPF_{3917DD00-3003-43F7-A409-7CAEC8BC6A9F}, id 0
						✓ Ethernet II, Src: IntelCor_a4:f3:e (78:0c:b8:a4:f3:e), Dst: DpNetTec_10:74:d0 (e4:26:86:10:74:d0)

Wi-Fi						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
icmp						
No.	Time	Source	Destination	Protocol	Length	Info
242	22.454746	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=25/6400, ttl=4 (no response found!)
243	22.473980	62.38.96.150	192.168.2.9	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
244	22.476955	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=26/6656, ttl=4 (no response found!)
245	22.494386	62.38.96.150	192.168.2.9	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
255	22.086464	62.38.96.150	192.168.2.9	ICMP	110	Destination unreachable (Port unreachable)
286	25.596944	62.38.96.150	192.168.2.9	ICMP	110	Destination unreachable (Port unreachable)
296	28.086861	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=27/6912, ttl=5 (no response found!)
302	29.291284	176.126.38.1	192.168.2.9	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
323	29.305537	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=28/7168, ttl=5 (no response found!)
327	29.382554	176.126.38.1	192.168.2.9	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
334	29.385222	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=29/7244, ttl=5 (no response found!)
335	29.408825	176.126.38.1	192.168.2.9	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
381	30.454227	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=30/7680, ttl=6 (no response found!)
382	30.476837	62.217.96.169	192.168.2.9	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
383	30.479010	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=31/7936, ttl=6 (no response found!)
385	30.498363	62.217.96.169	192.168.2.9	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
386	30.501587	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=32/8192, ttl=6 (no response found!)
387	30.520143	62.217.96.169	192.168.2.9	ICMP	110	Time-to-live exceeded (Time to live exceeded in transit)
392	31.580001	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=33/8448, ttl=7 (reply in 393)
393	31.620092	147.102.222.196	192.168.2.9	ICMP	106	Echo (ping) reply id=0x0001, seq=33/8448, ttl=58 (request in 392)
394	31.621386	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=34/8704, ttl=7 (reply in 395)
395	31.642069	147.102.222.196	192.168.2.9	ICMP	106	Echo (ping) reply id=0x0001, seq=34/8704, ttl=58 (request in 394)
396	31.643411	192.168.2.9	147.102.222.196	ICMP	106	Echo (ping) request id=0x0001, seq=35/8960, ttl=7 (reply in 397)
397	31.661886	147.102.222.196	192.168.2.9	ICMP	106	Echo (ping) reply id=0x0001, seq=35/8960, ttl=58 (request in 396)
						> Frame 148: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface \Device\NPF_{3917DD00-3003-43F7-A409-7CAEC8BC6A9F}, id 0
						✓ Ethernet II, Src: IntelCor_a4:f3:e (78:0c:b8:a4:f3:e), Dst: DpNetTec_10:74:d0 (e4:26:86:10:74:d0)

Το nmap είναι εργαλείο που χρησιμοποιείται για την δικτυακή σάρωση ενός υπολογιστή και την αναγνώριση κάποιων χαρακτηριστικών του, με πιο σημαντική λειτουργία τη σάρωση θυρών (port scanning).

Για πληροφορίες σχετικά με το nmap δείτε τη σελίδα <https://nmap.org/book/man.html>.

Τρεις από τις κυριότερες τεχνικές σάρωσης θυρών TCP είναι το SYN scan, το FIN scan και το NULL scan, οι οποίες περιγράφονται σχηματικά στα παρακάτω διαγράμματα.



3. Με το nmap να εντοπιστούν οι IP διευθύνσεις των υπολογιστών που είναι ενεργοί στο τοπικό δίκτυο του σπιτιού. Στο αρχείο που θα υποβληθεί, να υπάρχουν και τα σχετικά screenshots.

Στο nmap, εκτελώντας την εντολή nmap -T4 -A -v 192.168.1.3, εμφανίζεται όλες οι συσκευές που είναι συνδεδεμένες στο οικιακό δίκτυο, όπως φαίνεται και στην εικόνα.

```

Zenmap
Scan Tools Profile Help
Target: 192.168.1.3
Command: nmap -T4 -A -v 192.168.1.3

Hosts Services Nmap Output Ports / Hosts Topology Host Details Scans
OS Host
desktop-eumlcma (192.168.1.3)
192.168.56.1

Completed service scan at 22:34, 0.2ms elapsed (4 services on 1 host)
Initiating OS detection (try #1) against desktop-eumlcma (192.168.1.3)
NSE: Script scanning 192.168.1.3.
Initiating NSE at 22:34
Completed NSE at 22:34, 14.23s elapsed
Initiating NSE at 22:34
Completed NSE at 22:34, 0.02s elapsed
Initiating NSE at 22:34
Completed NSE at 22:34, 0.00s elapsed
Nmap scan report for desktop-eumlcma (192.168.1.3)
Host is up (0.0002s latency).
Not shown: 1000 closed ports
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
3306/tcp  open  mysql        MySQL (unauthorized)
Device type: general purpose
Running: Microsoft Windows 10
OS details: Microsoft Windows 10 1809 - 1909
Network Distance: 0 hops
TCP Sequence Prediction: Difficulty=258 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb-security-mode:
|_ 2.02:
|_ Message signing enabled but not required
| smb2-time:
|_ start-date: 2021-07-25T19:34:14
| end-date: N/A

NSE: Script Post-scanning:
Initiating NSE at 22:34
Completed NSE at 22:34, 0.00s elapsed
Initiating NSE at 22:34
Completed NSE at 22:34, 0.00s elapsed
Initiating NSE at 22:34
Completed NSE at 22:34, 0.00s elapsed
Read data files from: C:\Program Files (x86)\Nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.91 seconds
Raw packets sent: 1016 (45.41KB) | Rcvd: 2042 (86.82KB)

```

Από την αναζήτηση προέκυψε ότι το τοπικό δίκτυο του σπιτιού έχει 2 hosts και ο 1 εξ αυτών είναι ενεργός (όπως φαίνεται και στην τελευταία γραμμή όπου αναφέρεται 1 host up).

Τρέχοντας σε διαφορετικό PC, προκύπτει:

```

Zenmap
Scan Tools Profile Help
Target: 192.168.2.1
Command: nmap -T4 -A -v 192.168.2.1

Hosts Services Nmap Output Ports / Hosts Topology Host Details Scans
OS Host
vodafone.station (192.168.2.1)
DESKTOP-HRHQIOH
DESKTOP-B8C46J6
192.168.2.51

Completed service scan at 22:14, 0.0ms elapsed (4 services on 4 hosts)
NSE: Script Post-scanning...
Initiating NSE at 22:14
Completed NSE at 22:14, 0.00s elapsed
Initiating NSE at 22:14
Completed NSE at 22:14, 0.00s elapsed
Initiating NSE at 22:14
Completed NSE at 22:14, 0.00s elapsed
Read data files from: C:\Program Files (x86)\Nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 4 IP addresses (1 host up) scanned in 30.40 seconds
Raw packets sent: 2019 (89.67KB) | Rcvd: 29 (1.846KB)

Host vodafone.station (192.168.2.1):
NSE: Script Post-scanning...
Initiating NSE at 22:14
Completed NSE at 22:14, 0.00s elapsed
Initiating NSE at 22:14
Completed NSE at 22:14, 0.00s elapsed
Initiating NSE at 22:14
Completed NSE at 22:14, 0.00s elapsed
Read data files from: C:\Program Files (x86)\Nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 30.40 seconds
Raw packets sent: 2019 (89.67KB) | Rcvd: 29 (1.846KB)

TRACEROUTE
HOST RTT ADDRESS
1 6.99 ms vodafone.station (192.168.2.1)

NSE: Script Post-scanning...
Initiating NSE at 22:14
Completed NSE at 22:14, 0.00s elapsed
Initiating NSE at 22:14
Completed NSE at 22:14, 0.00s elapsed
Initiating NSE at 22:14
Completed NSE at 22:14, 0.00s elapsed
Read data files from: C:\Program Files (x86)\Nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 30.40 seconds
Raw packets sent: 2019 (89.67KB) | Rcvd: 29 (1.846KB)

```

Από την αναζήτηση προέκυψε ότι το τοπικό δίκτυο του σπιτιού έχει 4 hosts και ο 1 εξ αυτών είναι ενεργός (όπως φαίνεται και στην τελευταία γραμμή όπου αναφέρεται 1 host up).

4. Να χρησιμοποιηθεί το nmap για την εύρεση των υπηρεσιών που τρέχουν στις πασίγνωστες (well known) θύρες ενός υπολογιστή στο τοπικό δίκτυο του σπιτιού σας ενεργοποιώντας ταυτόχρονα την επιλογή για ανίχνευση του λειτουργικού συστήματος του στόχου (OS Fingerprinting). Στο αρχείο που θα νοποβληθεί, να υπάρχουν και τα σχετικά screenshots.

Επιλέγοντας την IP 192.168.1.3 από την παραπάνω λίστα και εκτελώντας την ακόλουθη εντολή:

nmap -O -p -1023 -O 192.168.1.3 προκύπτει η εικόνα που φαίνεται παρακάτω.

```

Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-25 22:37 GTB Daylight Time
Nmap scan report for desktop-eumlcma (192.168.1.3)
Host is up (0.00022s latency).
Not shown: 999 closed ports
PORT      STATE    SERVICE
135/tcp   open     msrpc
137/tcp   filtered netbios-ns
139/tcp   open     netbios-ssn
445/tcp   open     microsoft-ds
Device type: general-purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10
OS details: Microsoft Windows 10 1809 - 1909
Network Distance: 0 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.06 seconds
  
```

Σχολιασμός: Παρατηρείται ότι υπάρχουν ανοιχτά 3 tcp ports & 1 filtered tcp port που αφορούν το (135) msrpc, το (139) netbios-ssn και το (445) microsoft-ds & (137) netbios-ns για την filtered.

- (135): Σημειώνεται ότι η Remote Procedure Call (RPC) port 135 χρησιμοποιείται σε client/server applications όπως Exchange clients (ο πρόσφατα exploited messenger service) καθώς και άλλα Windows NT/2K/XP software.
- (137): Το NetBIOS είναι ένα protocol που χρησιμοποιείται για File και Print Sharing για όλες τις versions των Windows (NETBIOS Name Service (TCP/UDP: 137)). By default, όταν τα File και Print Sharing ενεργοποιούνται, γίνεται bind με τα πάντα, συμπεριλαμβανομένων των TCP/IP (The Internet Protocol), αντί για μόνο το local network, που σημαίνει ότι οι shared resources είναι διαθέσιμοι πάνω σε ολόκληρο το Internet για reading και deletion, εκτός και αν γίνουν configured κατάλληλα.
- (139): Όμοια με 137.
- (445): Το TCP port 445 χρησιμοποιείται για direct TCP/IP MS Networking access χωρίς το NetBIOS layer. Αυτή η υπηρεσία υλοποιείται στις πιο πρόσφατες εκδόσεις των Windows. Το SMB (Server Message Block) protocol χρησιμοποιείται μεταξύ άλλων για file sharing σε Windows NT/2K/XP.

Με κάποιες τροποποιήσεις, η αναμενόμενη εικόνα είναι η ακόλουθη:

```
nmap -p -1023 -O 192.168.1.3 -O
Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-25 22:37 GTB Daylight Time
Nmap scan report for desktop-eumlcma (192.168.1.3)
Host is up (0.00022s latency).
Not shown: 1019 closed ports
PORT      STATE    SERVICE
80/tcp    open     http
8080/tcp  open     domain
443/tcp   open     https
Device type: general purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10
OS details: Microsoft Windows 10 1809 - 1909
Network Distance: 0 hops
OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.06 seconds
```

Σχολιασμός:

Παρατηρείται λοιπόν ότι υπάρχουν ανοιχτά 3 tcp ports που σχετίζονται με το domain name, το http και το https κάτι που είναι αναμενόμενο αφού η παραπάνω συσκευή είναι access point και θα πρέπει να εξυπηρετεί DNS αιτήματα, ενώ ταυτόχρονα κάποιος χρήστης μπορεί να επικοινωνήσει μέσω http & https αιτημάτων.

Τρέχοντας σε διαφορετικό PC προκύπτει:

```
nmap -p -1023 -O 192.168.2.1 -O
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-01 12:33 GTB Daylight Time
Nmap scan report for vodafone.station (192.168.2.1)
Host is up (0.0042s latency).
Not shown: 1017 filtered ports
PORT      STATE    SERVICE
80/tcp    open     http
443/tcp   open     https
511/tcp   closed   bsdwan-rpc
851/tcp   closed   unknown
856/tcp   closed   unknown
865/tcp   closed   unknown
MAC Address: E4:26:86:10:74:D8 (DLink Technologies(Suzhou))
Device type: general purpose|WAP
Running: Linux 3.X|4.X, Asus embedded
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel cpe:/h:asus:rt-ac66u
OS details: Linux 3.10 - 4.11, Asus RT-AC66U WAP
Network Distance: 1 hop
OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.06 seconds
```

Παρατηρείται λοιπόν ότι υπάρχουν ανοιχτά 3 tcp ports που σχετίζονται με το domain name, το http και το https

6. Ανίχνευση εισβολής

Στο κομμάτι αυτό της εργασίας θα μελετηθεί η ανίχνευση εισβολής με χρήση του εργαλείου Snort το οποίο θα πρέπει να εγκατασταθεί (<https://www.snort.org/downloads>).

Να ανοιχθεί ένα παράθυρο εντολών και να γίνει μετάβαση στον κατάλογο C:\snort\bin. Πληκτρολογώντας “snort -h” και μετά πατώντας “Enter” μπορεί να δει κανείς το snort help. Να εξεταστεί η χρήση των ακόλουθων εντολών:

```
snort -W  
snort -i <if>  
snort -v  
snort -d  
snort -e  
snort -b  
snort -l <ld>  
snort -K <mode>  
snort -c <rules>  
snort -r <tf>  
snort -x  
snort -X
```

Ζητούνται τα ακόλουθα:

1. Να εντοπιστεί η διεπαφή του υπολογιστή μέσω της οποίας έχετε δικτυακή κίνηση. Στη συνέχεια, να γίνει αρχή με την εντολή snort -v -i?, όπου ? ο αριθμός της προηγούμενης διεπαφής, καταγραφής στη διεπαφή αυτή. Σε νέο παράθυρο εντολών να εκτελεστεί η εντολή ping προς κάποιο προορισμό. Στο αρχείο απαντήσεων να υπάρχουν τα σχετικά screenshots της καταγραφής του snort.

Αρχικά, εκτελείται η εντολή snort -W για την εύρεση της διεπαφής που χρησιμοποιείται.

Επειδή με το Wifi η σύνδεση δεν ήταν εφικτή, έγινε σύνδεση με Ethernet, και η διεπαφή είναι:

Επομένως εκτελείται η εντολή snort -v -i.

Επειδή, έπειτα από πολλές προσπάθειες και αναζήτηση για την εύρεση μιας λύσης, εν τέλει κατέστη αδύνατη η εκτέλεση του Snort από το ίδιο PC, τα παρακάτω screenshots είναι από διαφορετικό.

Ακολούθως, παρουσιάζεται η καταγραφή του snort για τα πακέτα που ανταλλάχθηκαν:

```
C:\Snort\bin>snort -W

      -> Snort! <-
o`-- Version 2.9.18-WIN64 GRE (Build 169)
     By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2010-2021 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using PCRE version: 8.10 2018-06-25
Using ZLIB version: 1.2.11

Index Physical Address      IP Address    Device Name      Description
-----  -----
1  00:00:00:00:00:00      disabled       \Device\NPF_{C83E0432-CEE7-424A-8B87-32352DEF05F}   NdisWan Adapter
2  00:00:00:00:00:00      0000:00:fe:80:00:00:00:00:1869:7273 \Device\NPF_{75480D12-4F84-42F2-A3F5-72FE0D7A89F} Microsoft
3  00:00:00:00:00:00      0000:00:fe:80:00:00:00:00:20F8:7ecf \Device\NPF_{39170D20-3003-43F7-A409-7CAEC8BC6A9F} Microsoft
4  00:00:00:00:00:00      disabled       \Device\NPF_{3A780A01-6AA4-4346-9DAE-F63593C04BC1}   NdisWan Adapter
5  00:00:00:00:00:00      0000:00:fe:80:00:00:00:00:b811:1817 \Device\NPF_{5A86B8F8-2A0F-4F9D-9987-83AC21C35D04} Microsoft
6  00:00:00:00:00:00      0000:00:fe:80:00:00:00:00:6199:2aa0 \Device\NPF_{DAC7B007-5704-47E5-9F98-20872C8D04B62} Microsoft
7  00:00:00:00:00:00      disabled       \Device\NPF_{EFD445F7-2544-4410-834E-6CB13150591}   NdisWan Adapter
8  00:00:00:00:00:00      disabled       \Device\NPF_Loopback Adapter for loopback traffic capture
9  00:FF:BA:BF:D4:05      0000:00:fe:80:00:00:00:00:00:00:154a \Device\NPF_{BABFDA04-7108-4003-9FB2-A447B6784EC} Kaspersky Security Data Escort Adapter
10  F8:CA:B8:16:D6:33     0000:00:fe:80:00:00:00:00:91bd:f28a \Device\NPF_{906D3289-3548-402D-AFA5-69315B05E7D2} Realtek PCIe FE Family Controller
11  00:FF:90:97:1E:52      0000:00:fe:80:00:00:00:00:00:8d4:ea5f \Device\NPF_{90971E52-F7D1-48AD-929C-7C3669465709} TAP-Windows Adapter V9

C:\Snort\bin>
```

Η 5^η διεπαφή είναι εκείνη με το Ethernet.

Ενδεικτικά, φαίνονται τα 2 echo requests και 2 echo replies από τα 4 και 4 που ανταλλάχθηκαν αντίστοιχα

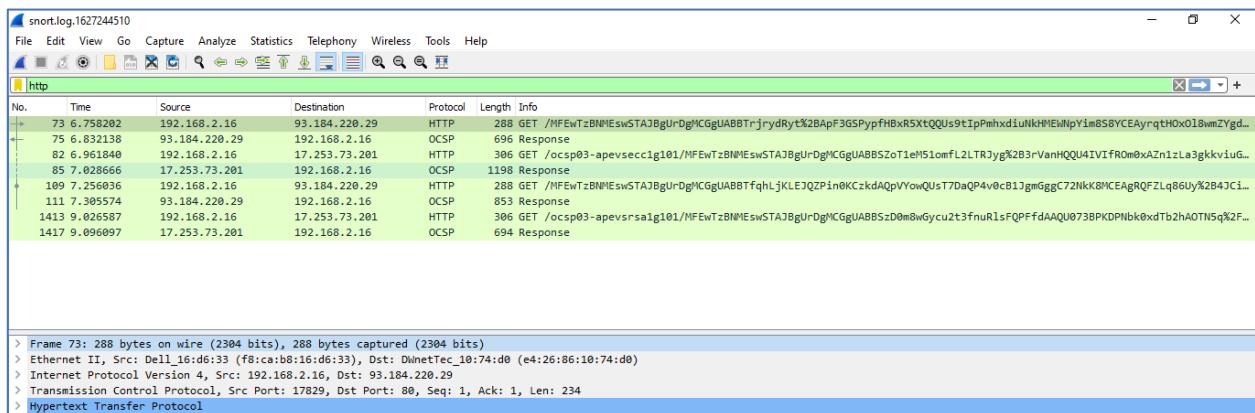
2. Snort Packet Logger Mode: Να χρησιμοποιηθεί το Snort για την καταγραφή πακέτων σε αρχείο ορίζοντας συγκεκριμένο φάκελο καταγραφής χρησιμοποιώντας την παράμετρο -l. Η εντολή αυτή καταγράφει όλα τα ανιχνευόμενα πακέτα σε log file.

Να χρησιμοποιηθεί ένας internet browser και να γίνει επίσκεψη σε μια ιστοσελίδα. Με το Wireshark μπορεί να διαβάσει κανείς το log αρχείο που δημιουργήθηκε. Στο αρχείο απαντήσεων να υπάρχει σχετικό screenshot από το Wireshark. Το log αρχείο μπορεί επίσης να διαβαστεί με το Snort χρησιμοποιώντας την επιλογή -r, η οποία ζεκινά λειτουργία αναπαραγωγής των καταγεγραμμένων πακέτων (playback mode). Στο αρχείο απαντήσεων να υπάρχει σχετικό screenshot.

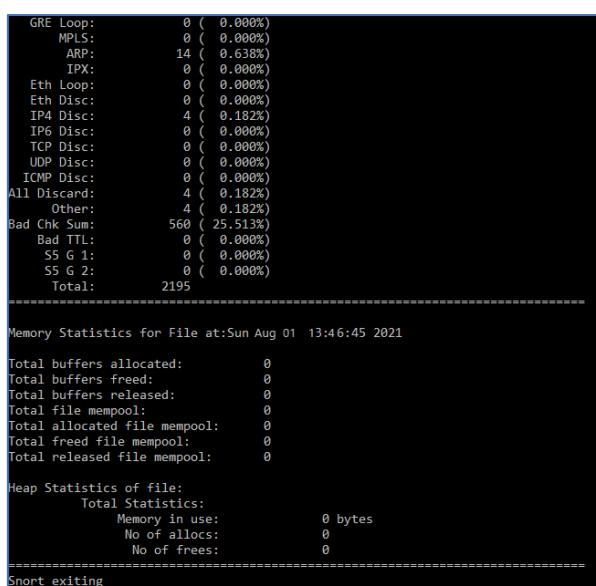
Με την εκτέλεση της εντολή snort -v -i5 -l c:\Snort\log μπορεί κανείς να καταγράψει την κίνηση σε ένα αρχείο.

Αρχικά, γίνεται επίσκεψη στην ιστοσελίδα <http://apple.com>. Έπειτα, γίνεται το άνοιγμα του παραγόμενου αρχείου με το Wireshark. Μετά, προστίθεται το φίλτρο εμφάνισης http για να παραμείνουν μόνο τα μηνύματα προς την ιστοσελίδα.

Ακολούθως, φαίνεται ένα στιγμιότυπο του αποτελέσματος.



Εναλλακτικά, μπορεί να διαβαστεί το αρχείο καταγραφής και μέσα από το Snort, με την εντολή snort -r c:\snort\log\snort.log.1627244510.



3. Θα γίνει χρήση της δυνατότητας του snort να χρησιμοποιεί κανόνες κατά την καταγραφή. Οι κανόνες είναι αυτής της μορφής:

```
action protocol address port direction address port (rule option)
```

Το πεδίο *action* προσδιορίζει τη βασική ενέργεια που πρέπει να εκτελεστεί, εάν οι τιμές των υπολοίπων πεδίων συμφωνούν με τα αντίστοιχα πεδία πακέτου που ανιχνεύτηκε. Η βασική τιμή του πεδίου αυτού, όταν το Snort λειτουργεί ως *Packet Logger* είναι "log", η οποία οδηγεί σε καταγραφή πακέτων σε *log file*, η τυπική θέση του οποίου είναι στο φάκελο *C:\snort\log*. Το πεδίο *protocol* μπορεί να είναι "TCP", "UDP" ή "ICMP". Η τιμή "Any" δεν υποστηρίζεται. Το πεδίο *address* προσδιορίζεται με τυπική CIDR σημειογραφία διευθύνσεων. Το πεδίο *port* αποδίδεται ως ένας φυσικός αριθμός (που αντιστοιχεί σε θύρα) ή ως διάστημα αριθμών, ενώ μπορεί να χρησιμοποιηθεί και ο τελεστής "!" για να εξαιρέσει θύρες. Το πεδίο *direction* είναι ίσο με "->" ή "<-", για μονόδρομη κίνηση, ή "<>" για αμφίδρομη κίνηση μεταξύ δύο διευθύνσεων. Το πεδίο *rule option* προσδιορίζει επιπλέον ενέργειες/συνθήκες που πρέπει να εκτελεστούν/ελεγχθούν, σε περίπτωση που οι καθορισμένες διευθύνσεις και τα πρωτόκολλα ανιχνευτούν. Σημειώνεται ότι στις παρενθέσεις μπορεί να ορίζονται περισσότερες από μία παράμετροι. Η έκφραση κάθε μιας πρέπει να καταλήγει σε ";", ακόμη κι αν είναι η μοναδική παράμετρος του κανόνα. Κάθε κανόνας πρέπει να έχει μοναδική ταυτότητα, η οποία ορίζεται από την παράμετρο "sid". Άλλες χρήσιμες παράμετροι είναι: "msg", "itype", "content", "flags", "length", "dsize", "ttl", κ.α. αναλυτικός κατάλογος των Rule Options είναι διαθέσιμος στο εγχειρίδιο χρήσης του Snort (http://manual.snort.org/snort_manual.html).

Να συνταχθεί κανόνας που θα αφορά κίνηση προς την πόρτα 80 (http) ή 443 (https) της ιστοσελίδας που θα επιλέχθηκε προηγουμένως και να αποθηκευτεί σε αρχείο txt κάτω από το φάκελο *C:\snort\rules*. Στη συνέχεια με την παράμετρο -r διαβαστεί το αρχείο καταγραφής του προηγούμενου ερωτήματος κάνοντας χρήση του κανόνα που φτιάχτηκε με χρήση της παραμέτρου -c. Στο αρχείο απαντήσεων να αντιγραφεί ο κανόνας και το αποτέλεσμα του snort.

Ο κανόνας για την ιστοσελίδα *apple.com* είναι ο εξής:

```
log tcp any any 17.253.144.10 [80,443];
```

Επιπλέον, δημιουργείται ένα αρχείο *file.txt* στο rules και αποθηκεύεται μέσα του η παραπάνω γραμμή. Έπειτα, εκτελείται η εντολή:

```
snort -c C:\Snort\Rules\file.text -r C:\Snort\log\snort.log.1627244510
```

4. Να συνταχθούν κανόνες που θα απομονώνουν κίνηση http, SMTP, ftp και telnet. Στο αρχείο απαντήσεων να αντιγραφούν οι κανόνες.

Οι κανόνες για την απομόνωση κίνησης του είναι οι εξής:

- http: log tcp any any <> any 80
- SMTP: log tcp any any <> any [25, 465, 587]
- ftp: log tcp any any <> any 21
- telnet: log tcp any any <> any 23

5. Snort Network Intrusion Detection System Mode: Σε αυτό μέρος της άσκησης θα μελετηθεί το Network Intrusion Detection System – NIDS Mode του Snort. Όταν το Snort λειτουργεί σε NIDS mode, υπάρχουν διάφορες ενέργειες (actions), που χρησιμοποιούνται στους κανόνες, όπως οι ακόλουθοι:

- alert: δημιουργεί ειδοποίηση χρησιμοποιώντας την επιλεγμένη μέθοδο ειδοποίησης, και στη συνέχεια καταγράφει το πακέτο σε αρχείο (Προσοχή: όλες οι ειδοποιήσεις αποθηκεύονται σε ένα κοινό αρχείο alert, η συνήθης θέση του οποίου είναι στο φάκελο C:\Snort\log.)
- log: καταγράφει το πακέτο σε αρχείο (Προσοχή: κάθε διεύθυνση IP αποκτά το δικό της φάκελο με τα δικά της αρχεία καταγραφής για μεταγενέστερη ανάλυση, η συνήθης θέση των οποίων είναι στο φάκελο C:\Snort\log.)
- pass: αγνοεί το πακέτο σιωπηρά και δεν το προωθεί στον προορισμό του
- activate: δημιουργεί ειδοποίηση και, στη συνέχεια, ενεργοποιεί ένα άλλο δυναμικό κανόνα
- dynamic: παραμένει αδρανής έως ότου ενεργοποιηθεί από έναν κανόνα activate, οπότε λειτουργεί ως κανόνας log
- drop: μπλοκάρει το πακέτο και το καταγράφει σε αρχείο
- reject: μπλοκάρει το πακέτο, το καταγράφει σε αρχείο και στη συνέχεια αποστέλλει μήνυμα “TCP reset” (αν το πρωτόκολλο είναι TCP) ή “ICMP port unreachable” (εάν το πρωτόκολλο είναι UDP).
- sdrop: μπλοκάρει το πακέτο, αλλά δεν το καταγράφει σε αρχείο

Όταν το Snort λειτουργεί σε NIDS mode, δεν καταγράφει κανονικά όλα τα πακέτα που έχουν συλληφθεί, όπως γίνεται όταν λειτουργεί σε network sniffer mode. Αντ' αυτού, εφαρμόζει τους κανόνες που έχει σε όλα τα πακέτα που συλλαμβάνει. Εάν ένα πακέτο ταιριάζει με ένα κανόνα, μόνον τότε είναι καταγράφεται σε αρχείο ή παράγεται ειδοποίηση. Εάν ένα πακέτο δεν ταιριάζει με κανένα κανόνα, το πακέτο απορρίπτεται σιωπηλά και δεν καταχωρείται στο αρχείο καταγραφής. Όταν το Snort χρησιμοποιείται σε NIDS mode, συνήθως παραμετροποιείται από το διαχειριστή από ένα αρχείο ρυθμίσεων (configuration file), το οποίο ενεργοποιείται με τη γραμμή εντολών. Αυτό το αρχείο διαμόρφωσης περιέχει κανόνες Snort ή αναφορά σε άλλα αρχεία που περιέχουν κανόνες Snort. Εκτός από τους κανόνες, το αρχείο ρυθμίσεων περιέχει επίσης πληροφορίες σχετικά με plug-ins εισόδου και εξόδου. Το τυπικό όνομα του αρχείου ρυθμίσεων του Snort είναι snort.conf.

Ζητείται να αναζητηθούν στο διαδίκτυο malware trace files (pcap αρχεία), τα οποία θα διαβαστούν με το Wireshark, να μελετηθούν και να εντοπιστούν τα πακέτα που οφείλονται στο συγκεκριμένο malware. Να γραφτεί ο τρόπος λειτουργίας του malware και στη συνέχεια να συνταχθούν κανόνες στο snort που θα στέλνουν και θα καταγράφουν alert μόλις ανιχνευτεί κάποιο από τα ύποπτα πακέτα, με μήνυμα: “Malware detected by <το ονοματεπώνυμο με λατινικούς χαρακτήρες>! ! !” και να αναπαραχθούν τα δεδομένα trace files με το Snort, υπό τους κανόνες που συντάχθηκαν για ανίχνευση της εισβολής αυτής. Παραδείγματα δικτυακών τόπων διάθεσης pcap αρχείων είναι τα ακόλουθα (στα οποία μπορεί να καταφύγει κανείς εάν δεν εντοπίσει εναλλακτικές):

- <https://www.malware-traffic-analysis.net/>
- <https://www.pcapanalysis.com/download-malware-samples/>

Αρχικά, σημειώνεται ότι malware trace files υπάρχουν στην ιστοσελίδα που δίνεται και που επισυνάπτεται και στα Resources. Ο εντοπισμός απειλής στο Wireshark δεν είναι εύκολος. Όμως, παρατηρήθηκαν μερικά προβληματικά πακέτα αλλά η απομόνωσή τους είναι δυσκολότερη.

177 0.010000	192.168.202.68	192.168.24.100	TCP	1438 http-alt(8080) → mtcp(1038) [ACK] Seq=150745 Ack=1 Win=14600 Len=1380[Malformed Packet]
181 0.000000	192.168.202.68	192.168.24.100	TCP	1438 http-alt(8080) → mtcp(1038) [ACK] Seq=154885 Ack=1 Win=14600 Len=1380
217 0.000000	192.168.202.68	192.168.24.100	TCP	1438 http-alt(8080) → mtcp(1038) [ACK] Seq=189385 Ack=1 Win=14600 Len=1380
220 0.000000	192.168.202.68	192.168.24.100	TCP	1438 http-alt(8080) → mtcp(1038) [ACK] Seq=192145 Ack=1 Win=14600 Len=1380
283 0.000000	192.168.202.68	192.168.24.100	TCP	1438 http-alt(8080) → mtcp(1038) [ACK] Seq=248373 Ack=1 Win=14600 Len=1380
302 0.010000	192.168.202.68	192.168.24.100	TCP	1438 http-alt(8080) → mtcp(1038) [ACK] Seq=264933 Ack=1 Win=14600 Len=1380
41 0.000000	192.168.202.68	192.168.24.100	TCP	1438 http-alt(8080) → mtcp(1038) [ACK] Seq=31741 Ack=1 Win=14600 Len=1380
384 0.000000	192.168.202.68	192.168.24.100	TCP	1438 http-alt(8080) → mtcp(1038) [ACK] Seq=335989 Ack=1 Win=14600 Len=1380[Malformed Packet]
448 0.010000	192.168.202.68	192.168.24.100	TCP	1438 http-alt(8080) → mtcp(1038) [ACK] Seq=393597 Ack=1 Win=14600 Len=1380
47 0.000000	192.168.202.68	192.168.24.100	TCP	1438 http-alt(8080) → mtcp(1038) [ACK] Seq=40821 Ack=1 Win=14600 Len=1380
491 0.000000	192.168.202.68	192.168.24.100	TCP	1438 http-alt(8080) → mtcp(1038) [ACK] Seq=437053 Ack=1 Win=14600 Len=1380
498 0.000000	192.168.202.68	192.168.24.100	TCP	1438 http-alt(8080) → mtcp(1038) [ACK] Seq=445333 Ack=1 Win=14600 Len=1380
501 0.000000	192.168.202.68	192.168.24.100	TCP	1438 http-alt(8080) → mtcp(1038) [ACK] Seq=449473 Ack=1 Win=14600 Len=1380
513 0.000000	192.168.202.68	192.168.24.100	TCP	1438 http-alt(8080) → mtcp(1038) [ACK] Seq=460513 Ack=1 Win=14600 Len=1380
583 0.010000	192.168.202.68	192.168.24.100	TCP	1438 http-alt(8080) → mtcp(1038) [ACK] Seq=531921 Ack=1 Win=14600 Len=1380
585 0.000000	192.168.202.68	192.168.24.100	TCP	1438 http-alt(8080) → mtcp(1038) [ACK] Seq=533381 Ack=1 Win=14600 Len=1380

Ένας ενδεικτικός κανόνας για το alert είναι ο εξής:

```
alert tcp 192.168.2.16 any -> any 8080 (msg: "Malware detected by Christos Tsoufis!!!";)
```

Παράρτημα:

To plain text είναι απόσπασμα από το βιβλίο “Homo Deus: A Brief History of Tomorrow” του Yuval Noah Harari και περιέχει 2.125 λέξεις. Παρατίθεται παρακάτω:

Modernity is a deal. All of us sign up to this deal on the day we are born, and it regulates our lives until the day we die. Very few of us can ever rescind or transcend this deal. It shapes our food, our jobs and our dreams, and it decides where we dwell, whom we love and how we pass away. At first sight, modernity looks like an extremely complicated deal, hence few try to understand what they have signed up to. It's like when you download some software and are asked to sign an accompanying contract which is dozens of pages of legalese; you take one look at it, immediately scroll down to the last page, tick 'I agree' and forget about it. Yet in fact modernity is a surprisingly simple deal. The entire contract can be summarised in a single phrase: humans agree to give up meaning in exchange for power. Up until modern times, most cultures believed that humans play a part in some great cosmic plan. The plan was devised by the omnipotent gods, or by the eternal laws of nature, and humankind could not change it. The cosmic plan gave meaning to human life, but also restricted human power. Humans were much like actors on a stage. The script gave meaning to their every word, tear and gesture – but placed strict limits on their performance. Hamlet cannot murder Claudius in Act I, or leave Denmark and go to an ashram in India. Shakespeare won't allow it. Similarly, humans cannot live for ever, they cannot escape all diseases, and they cannot do as they please. It's not in the script. In exchange for giving up power, premodern humans believed that their lives gained meaning. It really mattered whether they fought bravely on the battlefield, whether they supported the lawful king, whether they ate forbidden foods for breakfast or whether they had an affair with the next-door neighbour. This created some inconveniences, of course, but it gave humans psychological protection against disasters. If something terrible happened – such as war, plague or drought – people consoled themselves that 'We all play a role in some great cosmic drama, devised by the gods, or by the laws of nature. We are not privy to the script, but we can rest assured that everything happens for a purpose. Even this terrible war, plague and drought have their place in the greater scheme of things. Furthermore, we can count on the playwright that the story surely has a good ending. So even the war, plague and drought will work out for the best – if not here and now, then in the afterlife.' Modern culture rejects this belief in a great cosmic plan. We are not actors in any larger-than-life drama. Life has no script, no playwright, no director, no producer – and no meaning. To the best of our scientific understanding, the universe is a blind and purposeless process, full of sound and fury but signifying nothing. During our infinitesimally brief stay on our tiny speck of a planet, we fret and strut this way and that, and then are heard of no more. Since there is no script, and since humans fulfil no role in any great drama, terrible things might befall us and no power will come to save us, or give meaning to our suffering. There won't be a happy ending, or a bad ending, or any ending at all. Things just happen, one after the other. The modern world does not believe in purpose, only in cause. If modernity has a motto, it is 'shit happens'. On the other hand, if shit just happens, without any binding script or purpose, then humans too are not limited to any predetermined role. We can do anything we want – provided we can find a way. We are constrained by nothing except our own ignorance. Plagues and droughts have no cosmic meaning – but we can eradicate them. Wars are not a necessary evil on the way to a better future – but we can make peace. No paradise awaits us after death – but we can create paradise here on earth, and live in it for ever, if we just manage to overcome some technical difficulties. If we invest money in research, then scientific breakthroughs will accelerate technological progress. New technologies will fuel economic growth, and a growing economy could dedicate even more money to

research. With each passing decade we will enjoy more food, faster vehicles and better medicines. One day our knowledge will be so vast and our technology so advanced that we could distil the elixir of eternal youth, the elixir of true happiness, and any other drug we might possibly desire – and no god will stop us. The modern deal thus offers humans an enormous temptation, coupled with a colossal threat. Omnipotence is in front of us, almost within our reach, but below us yawns the abyss of complete nothingness. On the practical level, modern life consists of a constant pursuit of power within a universe devoid of meaning. Modern culture is the most powerful in history, and it is ceaselessly researching, inventing, discovering and growing. At the same time, it is plagued by more existential angst than any previous culture. This chapter discusses the modern pursuit of power. The next chapter will examine how humankind has used its growing power to somehow sneak meaning back into the infinite emptiness of the cosmos. Yes, we moderns have promised to renounce meaning in exchange for power; but there's nobody out there to hold us to our promise. We think we are smart enough to enjoy the full benefits of the modern deal, without paying its price. Even if we go on running fast enough and manage to fend off both economic collapse and ecological meltdown, the race itself creates huge problems. On the individual level, it results in high levels of stress and tension. After centuries of economic growth and scientific progress, life should have become calm and peaceful, at least in the most advanced countries. If our ancestors knew what tools and resources stand ready at our command, they would have surmised we must be enjoying celestial tranquillity, free of all cares and worries. The truth is very different. Despite all our achievements, we feel a constant pressure to do and produce even more. We blame ourselves, our boss, the mortgage, the government, the school system. But it's not really their fault. It's the modern deal, which we have all signed up to on the day we were born. In the premodern world, people were akin to lowly clerks in a socialist bureaucracy. They punched their card, and then waited for somebody else to do something. In the modern world, we humans run the business. So we are under constant pressure day and night. On the collective level, the race manifests itself in ceaseless upheavals. Whereas previously social and political systems endured for centuries, today every generation destroys the old world and builds a new one in its place. As the Communist Manifesto brilliantly put it, the modern world positively requires uncertainty and disturbance. All fixed relations and ancient prejudices are swept away, and new structures become antiquated before they can ossify. All that is solid melts into air. It isn't easy to live in such a chaotic world, and it is even harder to govern it. Hence modernity needs to work hard to ensure that neither human individuals nor the human collective will try to retire from the race, despite all the tension and chaos it creates. For that purpose, modernity upholds growth as a supreme value for whose sake we should make every sacrifice and risk every danger. On the collective level, governments, firms and organisations are encouraged to measure their success in terms of growth, and to fear equilibrium as if it were the Devil. On the individual level, we are inspired to constantly increase our income and our standard of living. Even if you are quite satisfied with your current conditions, you should strive for more. Yesterday's luxuries become today's necessities. If once you could live well in a three-bedroom apartment with one car and a single desktop, today you need a five-bedroom house with two cars and a host of iPods, tablets and smartphones. It wasn't very hard to convince individuals to want more. Greed comes easily to humans. The big problem was to convince collective institutions such as states and churches to go along with the new ideal. For millennia, societies strove to curb individual desires and bring them into some kind of balance. It was well known that people wanted more and more for themselves, but when the pie was of a fixed size, social harmony depended on restraint. Avarice was bad. Modernity turned the world upside down. It convinced human collectives that equilibrium is far more frightening than chaos, and because avarice fuels growth, it is a force for good. Modernity accordingly inspired people to want more, and dismantled the age-old disciplines that curbed greed. The resulting anxieties were assuaged to a large extent by free-market capitalism, which is one reason why this particular ideology

has become so popular. Capitalist thinkers repeatedly calm us: ‘Don’t worry, it will be okay. Provided the economy grows, the invisible hand of the market will take care of everything else.’ Capitalism has thus sanctified a voracious and chaotic system that grows by leaps and bounds, without anyone understanding what is happening and where we are rushing. (Communism, which also believed in growth, thought it could prevent chaos and orchestrate growth through state planning. After initial successes, it eventually fell far behind the messy freemarket cavalcade.) Bashing free-market capitalism is high on the intellectual agenda nowadays. Since capitalism dominates our world, we should indeed make every effort to understand its shortcomings, before they cause apocalyptic catastrophes. Yet criticising capitalism should not blind us to its advantages and attainments. So far, it’s been an amazing success – at least if you ignore the potential for future ecological meltdown, and if you measure success by the yardstick of production and growth. In 2016 we may be living in a stressful and chaotic world, but the doomsday prophecies of collapse and violence have not materialised, whereas the scandalous promises of perpetual growth and global cooperation are fulfilled. Although we experience occasional economic crises and international wars, in the long run capitalism has not only managed to prevail, but also to overcome famine, plague and war. For thousands of years priests, rabbis and muftis explained that humans cannot overcome famine, plague and war by their own efforts. Then along came the bankers, investors and industrialists, and within 200 years managed to do exactly that. So the modern deal promised us unprecedented power – and the promise has been kept. Now what about the price? In exchange for power, the modern deal expects us to give up meaning. How did humans handle this chilling demand? Complying with it could easily have resulted in a dark world, devoid of ethics, aesthetics and compassion. Yet the fact remains that humankind is today not only far more powerful than ever, it is also far more peaceful and cooperative. How did humans manage that? How did morality, beauty and even compassion survive and flourish in a world devoid of gods, of heaven and of hell? Capitalists are, again, quick to give all the credit to the invisible hand of the market. Yet the market’s hand is blind as well as invisible, and by itself could never have saved human society. Indeed, not even a country fair can maintain itself without the helping hand of some god, king or church. If everything is for sale, including the courts and the police, trust evaporates, credit vanishes and business withers. 6 What, then, rescued modern society from collapse? Humankind was salvaged not by the law of supply and demand, but rather by the rise of a new revolutionary religion – humanism. The modern deal offers us power, on condition that we renounce our belief in a great cosmic plan that gives meaning to life. Yet when you examine the deal closely, you find a cunning escape clause. If humans somehow manage to find meaning without deriving it from a great cosmic plan, this is not considered a breach of contract. This escape clause has been the salvation of modern society, for it is impossible to sustain order without meaning. The great political, artistic and religious project of modernity has been to find a meaning to life that is not rooted in some great cosmic plan. We are not actors in a divine drama, and nobody cares about us and our deeds, so nobody sets limits to our power – but we are still convinced our lives have meaning.

Resources:

<https://www.youtube.com/c/CrypTool2/featured>

<https://www.geeksforgeeks.org/caesar-cipher-in-cryptography/>

<https://www.geeksforgeeks.org/playfair-cipher-with-examples/>

<https://www.geeksforgeeks.org/hill-cipher/>

<https://www.geeksforgeeks.org/vigenere-cipher/>

<https://www.geeksforgeeks.org/vernam-cipher-in-cryptography/>

<https://en.wikipedia.org/wiki/Steganography>

<https://www.geeksforgeeks.org/image-steganography-in-cryptography/>

<https://www.geeksforgeeks.org/lsb-based-image-steganography-using-matlab/>

https://en.wikipedia.org/wiki/Digital_watermarking

<https://en.wikipedia.org/wiki/BPCS-steganography>

<https://github.com/nonameable/steganography-with-midi>

https://en.wikipedia.org/wiki/XOR_cipher

<https://www.geeksforgeeks.org/xor-cipher/>

https://en.wikipedia.org/wiki/Data_Encryption_Standard

<https://www.geeksforgeeks.org/data-encryption-standard-des-set-1/>

https://en.wikipedia.org/wiki/Triple_DES

<https://www.geeksforgeeks.org/double-des-and-triple-des/>

https://en.wikipedia.org/wiki/Advanced_Encryption_Standard

<https://www.geeksforgeeks.org/aes-full-form/>

[https://en.wikipedia.org/wiki/RSA_\(cryptosystem\)](https://en.wikipedia.org/wiki/RSA_(cryptosystem))

<https://www.geeksforgeeks.org/rsa-algorithm-cryptography/>

<https://en.wikipedia.org/wiki/Ping>

<https://en.wikipedia.org/wiki/Traceroute>

<https://www.speedguide.net/port.php?port=135>

<https://www.speedguide.net/port.php?port=137>

<https://www.speedguide.net/port.php?port=139>

<https://www.speedguide.net/port.php?port=445>

<https://www.netresec.com/?page=MACCDC>