ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΑΣΦΑΛΟΥΣ ΕΠΙΚΟΙΝΩΝΙΑΣ»

Δόλλας Νικόλαος Α.Μ.: 2113007

Επιβλέπων καθηγητής: Σταμούλης Γεώργιος

Συνεπιβλέπων καθηγητής: Μακρής Γεώργιος

Λαμία, Ιούνιος 2017

Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις (1), που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί χωρίς να τα περικλείω σε εισαγωγικά και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάσθηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.

2. Δέχομαι ότι η αυτολεξεί παράθεση χωρίς εισαγωγικά, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία 4/7/2017

Ο-Η Δηλών

Ν. Δ. Ιδιώτης

(Υπογραφή)

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

ΠΕΡΙΛΗΨΗ

Αυτή η πτυχιακή εργασία ασχολείται με την κρυπτογραφία και την ανάπτυξη μίας εφαρμογής ασφαλούς επικοινωνίας. Στο πρώτο κεφάλαιο, αναλύονται βασικές έννοιες, τεχνικοί όροι, κρυπτογραφικές υπηρεσίες και πρωτόκολλα. Επίσης, αναφορά γίνεται και στις αρχές μέτρησης της δύναμης, αλλά και στην αξιολόγηση της ασφάλειας των κρυπτοσυστημάτων. Στο δεύτερο, κεφάλαιο γίνεται μία ιστορική αναδρομή που περιλαμβάνει τις κλασσικές, αλλά και τις σύγχρονες τεχνικές κρυπτογράφησης. Στο τρίτο κεφάλαιο, γίνεται αναλυτική περιγραφή των σύγχρονων τεχνικών κρυπτογράφησης (συμμετρική και ασύμμετρη κρυπτογράφηση) και αναφέρεται ο τρόπος λειτουργίας κάποιων γνωστών αλγορίθμων. Να σημειωθεί ότι για τους αλγόριθμους AES και RSA δίνονται και παραδείγματα, ώστε να κατανοηθεί πλήρως η λειτουργία τους, γιατί θα χρησιμοποιηθούν στην εφαρμογή που θα παρουσιαστεί στη συνέχεια. Όλα τα παραπάνω είναι απαραίτητα για την ασφαλή μεταφορά δεδομένων στα δίκτυα, αλλά και στους ηλεκτρονικούς υπολογιστές. Τέλος, στο τέταρτο και τελευταίο κεφάλαιο, παρουσιάζεται η εφαρμογή που δημιουργήθηκε και έχει ως στόχο την επίτευξη ασφαλούς επικοινωνίας μεταξύ δύο χρηστών.

Πίνακας Περιεχομένων

1. ΕΙΣΑΓΩΓΗ

1.1 Ορολογία και βασικές έννοιες

Η κρυπτογραφία μελετά τρόπους με τους οποίους μπορούμε να μετασχηματίσουμε ένα μήνυμα σε μία μορφή, στην οποία θα αποκρύπτεται το περιεχόμενο του, έτσι ώστε να μην μπορεί να διαβαστεί από κανέναν άλλο, παρά μόνο από το παραλήπτη, για τον οποίο προορίζεται το μήνυμα. Ο ορισμός του Rivest (1990) εισάγει την έννοια του αντιπάλου (αυτός που προσπαθεί να κλέψει το αρχικό μήνυμα) και είναι ίσως ο πιο κατάλληλος ορισμός. Έτσι, μπορούμε να πούμε πως η κρυπτογραφία ασχολείται με την επικοινωνία και την ύπαρξη αντιπάλων. (Κάτος & Στεφανίδης, 2003: 2)

Επομένως, η ύπαρξη ενός αντιπάλου σε μία επικοινωνία είναι η βασική αιτία ύπαρξης της κρυπτογραφίας στο μήνυμα μας. Εκτός από την επιθυμία μας να κρύψουμε ένα μήνυμα από τους αντίπαλους, που θέλουν να το δουν, που μπορεί να είναι μέχρι και η οποιαδήποτε υπηρεσία η οποία υποστηρίζει την ανταλλαγή μηνυμάτων, ώστε να μας διευκολύνει να προωθήσουμε το μήνυμα μας. Συνεπώς θα πρέπει, με κάποιον τρόπο να μην αλλοιωθεί το μήνυμά μας, ή αν αλλοιωθεί, να μπορέσει να το καταλάβει ο παραλήπτης και να φτάσει στον πραγματικό του παραλήπτη και όχι σε κάποιον που τον υποδύεται. (Κάτος & Στεφανίδης, 2003: 2)

Την αρχική μορφή του μηνύματος την ονομάζουμε απλό κείμενο (plaintext), ενώ το κρυπτογραφημένο κείμενο το ονομάζουμε κρυπτοκείμενο (ciphertext). Ο μετασχηματισμός του απλού κειμένου σε κρυπτοκείμενο ονομάζεται κρυπτογράφηση (encryption), ενώ ο μετασχηματισμός του κρυπτοκειμένου σε απλό κείμενο (η οποία είναι και η αντίστροφη διαδικασία της κρυπτογράφησης), και επιτρέπει την ανάγνωση του αρχικού κειμένου από τον εξουσιοδοτημένο παραλήπτη, ονομάζεται αποκρυπτογράφηση (decryption). Οι διαδικασίες της κρυπτογράφησης και της αποκρυπτογράφησης πραγματοποιούνται με έναν αλγόριθμο κρυπτογράφησης και αποκρυπτογράφησης αντίστοιχα. Οι αλγόριθμοι για την κρυπτογράφηση και την αποκρυπτογράφηση αποτελούν τον κρυπταλγόριθμο (cipher). Τόσο για την διαδικασία της κρυπτογράφησης, όσο και της αποκρυπτογράφησης, απαιτείται μια ακόμα πληροφορία

εισόδου που την ονομάζουμε κλειδί (key). Η ύπαρξη του κλειδιού είναι και η κύρια διαφορά μεταξύ της κρυπτογράφησης και της κωδικοποίησης (encoding). Αναλυτικότερα, η κρυπτογράφηση και αποκρυπτογράφηση ενός κειμένου μπορεί να εκτελεστεί με επιτυχία μόνον από τον κάτοχο του σωστού κλειδιού. (Κάτος & Στεφανίδης, 2003: 3)

Η περιγραφή των διαδικασιών κρυπτογράφησης και αποκρυπτογράφησης αποτελεί το κρυπτοσύστημα. Το κρυπτοσύστημα, μπορούμε να πούμε ότι είναι ένα σύνολο από κρυπτογραφικές τεχνικές που χρησιμοποιείται για να παρέχει υπηρεσίες ασφάλειας. Ένας αντίπαλος ενός κρυπτοσυστήματος θα εστιάσει στο να ανακαλύψει το σωστό κλειδί, δηλαδή το κλειδί εκείνο με το οποίο θα μπορέσει να διαβάσει το μήνυμα, χωρίς το μήνυμα να προορίζεται για αυτόν.

Κρυπτανάλυση είναι η επιστήμη που ασχολείται με την αποκρυπτογράφηση του κρυπτοκειμένου, χωρίς την κατοχή του κλειδιού. Δηλαδή, μπορούμε να πούμε, πως είναι η μελέτη μαθηματικών τεχνικών για τη ακύρωση των υπηρεσιών ασφάλειας, με απλά λόγια, η προσπάθεια για την εύρεση του μυστικού κλειδιού. (Κάτος & Στεφανίδης, 2003: 3)

Ο αντίπαλος μπορεί να ενδιαφέρεται περισσότερο στο να ανακαλύψει το μυστικό κλειδί, γιατί με την κατοχή του θα μπορεί να αποκρυπτογραφήσει όλα τα μηνύματα τα οποία στάλθηκαν με τη χρήση του συγκεκριμένου κλειδιού. Και πάλι όμως, στόχος του αντιπάλου είναι να ανακτήσει την πληροφορία που βρίσκεται στο κρυπτοκείμενο. Έτσι, ο αντίπαλος προσπαθεί να κλέψει τις πληροφορίες του κρυπτοκειμένου, χωρίς να έχει το μυστικό κλειδί. Αυτή η διαδικασία όπου ένας αντίπαλος καταφέρνει με κατάλληλους χειρισμούς ενός πρωτοκόλλου, να καταστήσει το πρωτόκολλο αδύναμο στο να προσφέρει την κρυπτογραφική υπηρεσία (δηλαδή καταφέρνει να εκτελέσει την αποκρυπτογράφηση σε ένα κρυπτοκείμενο το οποίο δεν του ανήκει), ονομάζεται αποτυχία πρωτοκόλλου (protocol failure). Ο αντίπαλος έτσι, μπορεί να μη γνωρίζει το μυστικό κλειδί, το οποίο είναι πολύ καλά προστατευμένο μέσα στο σύστημα, αλλά μπορεί να εκμεταλλευτεί την πρόσβασή του σε αυτό και να επιτύχει την αποκρυπτογράφηση. (Κάτος & Στεφανίδης, 2003: 3)

1.2 Κρυπτογραφικές Υπηρεσίες και Πρωτόκολλα

Οι κρυπτογραφικές υπηρεσίες χρησιμοποιούνται στην κρυπτογραφία με σκοπό την αντιμετώπιση συγκεκριμένων απειλών. Οι βασικές κρυπτογραφικές υπηρεσίες χωρίζονται στις ακόλουθες:

- **Εμπιστευτικότητα (Confidentiality)**: είναι η προστασία των πληροφοριών από τη μη εξουσιοδοτημένη πρόσβαση. Η εμπιστευτικότητα θα πρέπει να προσφέρεται με τέτοιον τρόπο, ώστε να διασφαλίσουμε την προσπελασιμότητα της πληροφορίας, μόνον από όσους έχουν τα απαραίτητα δικαιώματα
- **Ακεραιότητα (Integrity)**: είναι η προστασία τροποποίησης των πληροφοριών από μη εξουσιοδοτημένα άτομα. Η ακεραιότητα θα πρέπει να παρέχει στον κάτοχο του μηνύματος τη δυνατότητα να μπορεί να εντοπίσει πιθανές αλλαγές από μη εξουσιοδοτημένα άτομα. Δηλαδή, στοχεύει στη διαφύλαξη της ακρίβειας και της πληρότητας της πληροφορίας και των μεθόδων επεξεργασίας αυτής
- **Αυθεντικοποίηση (Authentication)**: είναι η εξασφάλιση της αναγνώρισης του χρήστη με τον οποίο επικοινωνούμε, ενώ αυθεντικοποίηση δεδομένων (data authentication) είναι ότι το μήνυμα προέρχεται πράγματι από τον αποστολέα, με τον οποίο θέλουμε να επικοινωνήσουμε
- **Αποποίηση Ευθύνης (Non-repudiation)**: είναι η υπηρεσία κατά την οποία ο παραλήπτης δεν μπορεί να αρνηθεί ότι έλαβε το μήνυμα ή η υπηρεσία κατά την οποία ο αποστολέας δεν μπορεί να αρνηθεί ότι έστειλε το μήνυμα

(βλ. Κάτος & Στεφανίδης, 2003: 9)

Επίσης έχουμε και κάποιες συμπληρωματικές κρυπτογραφικές υπηρεσίες οι οποίες είναι:

- **Διαθεσιμότητα (Availability)**: είναι η διασφάλιση της προσπελασιμότητας της πληροφορίας σε εξουσιοδοτημένους χρήστες όταν απαιτείται

- **Λογοδοσία (Accountability)**: είναι η υπευθυνότητα και αρμοδιότητες για την σωστή χρήση ενός αγαθού
- **Αξιοπιστία (Reliability)**: είναι για παράδειγμα η αξιοπιστία ενός πρωτοκόλλου επικοινωνίας

(βλ. Ντούσκας, 2017: 17-18)

Άρα, η ασφάλεια πληροφοριών (information security) είναι η διασφάλιση εμπιστευτικότητας, ακεραιότητας, αυθεντικοποίησης και αποποίησης ευθύνης των πληροφοριών, καθώς και της διαθεσιμότητας, λογοδοσίας και αξιοπιστίας για τους πόρους και τις πληροφορίες. (Ντούσκας, 2017: 19)

Κρυπτογραφικό πρωτόκολλο είναι μία διαδικασία με πολύ συγκεκριμένα βήματα, τα οποία πρέπει να ακολουθήσουν τα μέλη για να επικοινωνήσουν, με την χρήση μίας συγκεκριμένης κρυπτογραφικής υπηρεσίας. Έτσι, ως βασικό χαρακτηριστικό ενός κρυπτογραφικού πρωτοκόλλου είναι ότι το κάθε μέλος της επικοινωνίας πρέπει να γνωρίζει, σε κάθε χρονική στιγμή, ποιο βήμα πρέπει να εκτελεστεί και πως πρέπει να εκτελεστεί. Αν κάποιο από αυτά τα βήματα δεν εκτελεστεί σωστά, τότε το πρωτόκολλο καταρρέει, με αποτέλεσμα τον τερματισμό της επικοινωνίας ή του κρυπτογραφικού πρωτοκόλλου. (Κάτος & Στεφανίδης, 2003: 10)

1.3 Αρχές Μέτρησης Δύναμης και Μοντέλα Αξιολόγησης Ασφάλειας Κρυπτοσυστημάτων

Η αρχή του Kerchoff

Ένα κριτήριο στην αντικειμενική μέτρηση της δύναμης ενός κρυπτοσυστήματος είναι γνωστό ως η αρχή του Kerchoff, σύμφωνα με τον οποίο η ασφάλεια ενός κρυπτοσυστήματος δεν εξαρτάται από τη μυστικότητα του αλγόριθμου αλλά μόνο από τη φύλαξη του μυστικού κλειδιού. Επίσης,

ανέφερε ότι κάποιος που σχεδιάζει κρυπτοσυστήματα, πρέπει να υπολογίζει ότι ο εχθρός θα αποκτήσει αμέσως πλήρη εξοικείωση με αυτά. (Κάτος & Στεφανίδης, 2003: 11)

Το να είναι ο αλγόριθμος κρυπτογράφησης μυστικός δημιουργεί πολλούς κινδύνους και προβλήματα. Αρχικά, δεν θα μπορούσαμε να αξιολογήσουμε αντικειμενικά τον αλγόριθμο, με αποτέλεσμα να μην μπορεί να υπολογισθεί η πραγματική κρυπτογραφική του δύναμη. Επίσης με την αντίστροφη ανάλυση (reverse engineering) του αλγορίθμου, θα δινόταν η δυνατότητα στον αντίπαλο να ανακαλύψει τη δομή, αλλά και πολλές λεπτομέρειες για τον αλγόριθμο κρυπτογράφησης. Τέλος, μπορούμε εύκολα να καταλάβουμε πως, εάν ένα κλειδί γίνει γνωστό, η αντικατάστασή του είναι εύκολη, ενώ εάν ο αλγόριθμος κρυπτογράφησης γίνει γνωστός, τότε δεν είναι εύκολη. (Κάτος & Στεφανίδης, 2003: 11-12)

Τα μέτρα του Shannon

Ο Shannon, το 1949, διατύπωσε ένα σύνολο από μέτρα, τα οποία χαρακτηρίζουν έναν ορθά σχεδιασμένο αλγόριθμο κρυπτογράφησης και είναι τα εξής:

1. Βαθμός απαιτούμενης κρυπτογραφικής ασφάλειας
2. Μήκος του κλειδιού
3. Πρακτική εκτέλεση της κρυπτογράφησης και της αποκρυπτογράφησης
4. Διόγκωση του κρυπτοκειμένου
5. Διάδοση των σφαλμάτων κρυπτογράφησης

Η ύπαρξη των μέτρων σε ένα κρυπτοσύστημα είναι υποχρεωτική, αλλά στην πραγματικότητα δεν υπάρχει κρυπτοσύστημα το οποίο να ικανοποιεί όλα τα μέτρα στο μέγιστό τους.

(βλ. Κάτος & Στεφανίδης, 2003: 12)

Σύγχυση και Διάχυση

Δύο ιδιότητες που χρησιμοποιούνται στην αξιολόγηση της κρυπτογραφικής δύναμης είναι η σύγχυση (confusion) και η διάχυση (diffusion).

- **Σύγχυση**: είναι η ικανότητα του αλγόριθμου κρυπτογράφησης, ώστε όταν συμβαίνει μία αλλαγή στο απλό κείμενο, ο αντίπαλος δεν μπορεί να προβλέψει ποιες μεταβολές θα συμβούν στο κρυπτοκείμενο
- **Διάχυση**: είναι η ικανότητα του αλγόριθμου κρυπτογράφησης ώστε ένα τμήμα του απλού κειμένου, κατά την κρυπτογράφηση, να επηρεάζει όσο το δυνατόν περισσότερα τμήματα του κρυπτοκειμένου

(βλ. Ντούσκας, 2017: 2)

Η δύναμη ενός κρυπτοσυστήματος να αντιστέκεται στις επιθέσεις ενός αντιπάλου, δημιούργησε την ανάγκη καθορισμού μερικών αντικειμενικών μέτρων για τη μέτρηση της κρυπτογραφικής δύναμης. Αυτό είχε ως αποτέλεσμα, τη δημιουργία των εξής μαθηματικών μοντέλων:

- **Ασφάλεια άνευ όρων (unconditionally secure)**: ένα κρυπτοσύστημα είναι άνευ όρων ασφαλές, όταν ο αντίπαλος δεν μπορεί να πάρει καμία πληροφορία για το απλό κείμενο από το κρυπτοκείμενο. Η υπόθεση βασίζεται στο ότι ο αντίπαλος έχει άπειρη υπολογιστική ισχύ στη διάθεσή του
- **Υπολογιστική ασφάλεια (computationally secure)**: ένα κρυπτοσύστημα είναι υπολογιστικά ασφαλές, ώστε όταν ο αντίπαλος θέλει να το “σπάσει”, απαιτείται υπολογιστική ισχύς, πέραν των διαθέσιμων που έχει. Ο υπολογισμός γίνεται με βάση τον καλύτερο αλγόριθμο που γνωρίζει ο αντίπαλος, προκειμένου να “σπάσει” το κρυπτοσύστημα. Ο πιο συχνός αλγόριθμος που χρησιμοποιείται, για να σπάσει ένα κρυπτοσύστημα και είναι γνωστός πάντα από όλους, είναι αυτός της εξαντλητικής αναζήτησης (exhaustive search), όπου ο αντίπαλος δοκιμάζει ένα-ένα τα κλειδιά, μέχρι να βρει το σωστό

- **Ασφάλεια θεωρητικής πολυπλοκότητας (complexity theoretic)**: θεωρείται ότι ο αντίπαλος μπορεί να πραγματοποιήσει επίθεση στο κρυπτοσύστημα, η οποία όμως απαιτεί πολυωνυμική υπολογιστική ισχύ. Δηλαδή, οι παράμετροι ασφάλειας του κρυπτοσυστήματος μπορούν να εκφραστούν πολυωνυμικά ως προς το χώρο και το χρόνο
- **Αποδείξιμη ασφάλεια (provable security)**: ένα κρυπτοσύστημα είναι αποδείξιμα ασφαλές, όταν μπορούμε να αποδείξουμε ότι η ασφάλειά του είναι ισοδύναμη με κάποιο πρόβλημα που θεωρείται «δύσκολο». Τέτοιου είδους προβλήματα βρίσκουμε στη θεωρία αριθμών, όπως για παράδειγμα, η παραγοντοποίηση ενός μεγάλου σύνθετου αριθμού στους πρώτους παράγοντές του (αλγόριθμος RSA)

(βλ. Κάτος & Στεφανίδης, 2003: 13-14)

2. ΙΣΤΟΡΙΑ ΤΗΣ ΚΡΥΠΤΟΓΡΑΦΙΑΣ

Η χρήση της κρυπτογραφίας δεν αποτελεί μόνο γεγονός των ημερών μας, αλλά υπήρχε χιλιάδες χρόνια πριν. Σε σχέση λοιπόν με τις τεχνικές που χρησιμοποιούσαν παλαιότερα για κάποιες μεθόδους κρυπτογράφησης, η τεχνικές που γνωρίζουμε σήμερα έχουν εξελιχθεί κατά πολύ. Σκοπός αυτού του κεφαλαίου είναι η κατασκευή ενός χρονοδιαγράμματος των πιο σημαντικών γεγονότων κρυπτογράφησης που έχουν συμβεί με το πέρασμα των αιώνων.

Υπάρχουν διάφορα χρονοδιαγράμματα κρυπτογράφησης που είναι διαθέσιμα στο Internet, πολλά από τα οποία είναι πολύ λεπτομερή και αναλύουν όλα τα γεγονότα που σχετίζονται με την κρυπτογραφία. Σε αυτό το κεφάλαιο παρουσιάζονται τα πιο βασικά γεγονότων που έχουν συμβεί, ώστε να αποκτήσουμε μια σφαιρική άποψη των γεγονότων.

Οι τεχνικές κρυπτογράφησης, όπως θα δούμε και στην συνέχεια, μπορούν να χωριστούν σε δύο κατηγορίες: κλασσικές τεχνικές κρυπτογράφησης και σύγχρονες τεχνικές κρυπτογράφησης.

Οι ημερομηνίες που θα αναφερθούν είναι μία εκτίμηση των ημερομηνιών που έλαβαν χώρα τα γεγονότα κρυπτογράφησης, εφόσον ορισμένες πηγές αναφέρουν την ημερομηνία, κατά την οποία αναπτύχθηκε μια ιδέα και άλλες την ημερομηνία δημοσίευσης για πρώτη φορά.

(Πηγές: https://en.wikipedia.org/wiki/History_of_cryptography
http://world.std.com/~cme/html/timeline.html)

2.1 Κλασσικές Τεχνικές Κρυπτογράφησης

Οι κλασσικές τεχνικές κρυπτογράφησης είναι τεχνικές βασισμένες σε στυλό και χαρτί οι οποίες αναπτύχθηκαν σε μία εποχή που δεν υπήρχαν υπολογιστές, παρόλο που ορισμένες από αυτές τις τεχνικές χρησιμοποιούνται μέχρι και σήμερα σαν αλγόριθμοι, με την βοήθεια του υπολογιστή. Γενικά, οι κλασσικές τεχνικές κρυπτογράφησης βασίζονται στην αντικατάσταση των γραμμάτων και στη χρήση διαφορετικών συμβόλων.

Οι ιστορικοί πιστεύουν ότι η πρώτη απόπειρα κρυπτογράφησης συνέβη στην αρχαία Κίνα επειδή χρησιμοποιούσαν την ίδια τους την γραπτή γλώσσα ως τεχνική κρυπτογράφησης. Έτσι, μόνο οι πολίτες από ανώτερες κοινωνικές τάξεις, είχαν τη δυνατότητα να μάθουν να διαβάζουν και να γράφουν, πετυχαίνοντας με αυτόν τον τρόπο την μεταφορά μυστικών μηνυμάτων μεταξύ τους, χωρίς οι πολίτες κατώτερων κοινωνικών τάξεων (π.χ. αγρότες) να μπορούν να αποκρυπτογραφούν τα μηνύματα. Η πρώτη τεκμηριωμένη χρήση της κρυπτογραφίας, χρονολογείται περίπου το 1900 π.Χ. στην Αίγυπτο, όπου βρέθηκαν επιγραφές που δεν περιείχαν διαφορετικό σύνολο ιερογλυφικών, αλλά ένα σύστημα μερικών μη τυποποιημένων ιερογλυφικών. Το συμπέρασμα είναι ότι ο γραφέας των επιγραφών χρησιμοποίησε κάποιο είδος τεχνικής κρυπτογράφησης, για να κρύψει το αληθινό μήνυμα που περιείχε το ιερογλυφικό. (Morkel & Eloff, 2004: 5)

Στη Μεσοποταμία, το 1500 π.Χ. βρέθηκε ένα πλακίδιο που περιείχε μια κρυπτογραφημένη συνταγή για το κεραμικό λούστρο. Μεταξύ 500 και 600 π.Χ., οι Εβραίοι γραφείς χρησιμοποίησαν τον κώδικα ATBASH, όταν έγραψαν το βιβλίο του Ιερεμία. Ο κρυπτογράφος ATBASH χρησιμοποιούσε την λογική της αντικατάστασης, όπου έκανε αντικατάσταση το τελευταίο

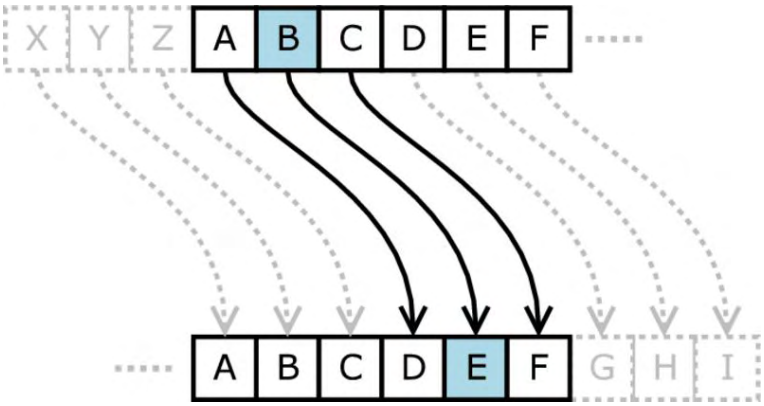
γράμμα του αλφάβητου ως πρώτο και αντίστροφα. Αυτός ο κρυπτογραφημένος κώδικας ήταν, όπως καταλαβαίνουμε, πολύ απλός, καθώς υπήρχε μόνο μία πιθανή απάντηση, για να σπάσει ο κώδικας. (Morkel & Eloff, 2004: 5)

א ב ג ד ה ו ז ח ט י כ ל מ נ ס ע פ צ ק ר ש ת
ת ש ר ק צ פ ע ס נ מ ל כ י ט ח ז ו ה ד ג ב א

Το 486 π.Χ., αναπτύχθηκε ως τεχνική στρατιωτικής κρυπτογράφησης μία μέθοδος κρυπτογράφησης, που ονομάζεται Greek Scytale. Οι στρατιώτες τύλιγαν μια λωρίδα παπύρου γύρω από ένα κομμάτι ξύλου. Το μήνυμα ήταν γραμμένο πάνω στον πάπυρο και, όταν αφαιρούνταν από το ξύλο, διάφορα μέρη του μηνύματος βρισκόταν σε διαφορετικά μέρη του παπύρου. Μόνο όταν ο πάπυρος ήταν τυλιγμένος γύρω από ένα ταιριαστό κομμάτι ξύλου, θα μπορούσε να συνταχθεί το αρχικά σωστό μήνυμα. Ωστόσο, υπάρχουν πρόσφατοι ισχυρισμοί ότι ο Greek Scytale είναι απλώς ένας μύθος. (Morkel & Eloff, 2004: 5)

Η πιο γνωστή τεχνική κλασσικής κρυπτογράφησης είναι ίσως ο Caesar Cipher, που αναπτύχθηκε από τον Ιούλιο Καίσαρα (Julius Caesar) στα τέλη του 50 και 60 π.Χ.. Ο Caesar Cipher χρησιμοποιούσε την μέθοδο της αντικατάστασης, όπου κάθε γράμμα στο αλφάβητο αντικαθίσταται από κάποιο άλλο γράμμα της αλφαβήτου. Αυτό γινόταν με μία μέθοδο ολίσθησης, δηλαδή κάθε γράμμα του αρχικού μηνύματος μετακινιόταν k θέσεις δεξιά (π.χ. για k=3 το γράμμα a θα αντικαθιστούνταν με το γράμμα d). Ο κωδικός του Caesar είναι μία απλή τεχνική κρυπτογράφησης, αλλά ήταν πολύ αποτελεσματικός και επιτυχημένος για την εποχή

του, επειδή πολύ λίγοι άνθρωποι μπορούσαν να διαβάσουν και να γράψουν. Το κύριο μειονέκτημα του Caesar Cipher ήταν το γεγονός, ότι για την αποκρυπτογράφηση του μηνύματος χρειαζόταν λίγος χρόνος και υπομονή, επειδή για την κρυπτογράφηση του χρησιμοποιούνταν ένα πολύ προφανές μοτίβο. (Morkel & Eloff, 2004: 5-6)



Στην Ευρώπη, η περίοδος μεταξύ 500 και 1400 η κρυπτογράφηση θεωρήθηκε ως μαύρη μαγεία και συνεπώς απαγορευόταν. Έτσι, δεν γνωρίζουμε σχεδόν τίποτα για αυτήν την περίοδο και ένα μεγάλο μέρος της γνώσης σχετικά με την κρυπτογράφηση χάθηκε ή καταστράφηκε.

Το 1553 ο Giovan Belaso ανέφερε για πρώτη φορά την ιδέα ενός κωδικού πρόσβασης. Πρότεινε έναν τύπο κρυπτογράφησης όπου απαιτείται ο σωστός κωδικός πρόσβασης για την αποκρυπτογράφηση του κρυπτογραφημένου μηνύματος. Αυτός ο κωδικός πρόσβασης είναι ο ίδιος με το «μυστικό κλειδί» που χρησιμοποιείται σήμερα στην συμμετρική κρυπτογράφηση. (Morkel & Eloff, 2004: 6)

Το 1585 ο Blaise de Vigenère έγραψε ένα βιβλίο σχετικά με τους κρυπτογράφους που μπορεί να θεωρηθεί ως μετάβαση από τις κλασσικές στις σύγχρονες τεχνικές κρυπτογράφησης. Σε αυτό το βιβλίο εξήγησε το πρώτο σύστημα κλειδιού κρυπτογράφησης. Για να κρυπτογραφήσει ένα μήνυμα, χρειάζεται ένα κλειδί το οποίο είναι κοινό χαρακτηριστικό των σύγχρονων τεχνικών κρυπτογράφησης, όπως είναι ο Vigenère Square που χρησιμοποιεί το αλφάβητο ως κοινό χαρακτηριστικό των κλασσικών τεχνικών κρυπτογράφησης. Για να κρυπτογραφήσει ένα μήνυμα, ο αποστολέας πρέπει να λάβει το πρώτο και στην συνέχεια με την ίδια διαδικασία τα υπόλοιπα γράμματα του plaintext και να βρει την αντίστοιχη στήλη στον Vigenère Square. Στη

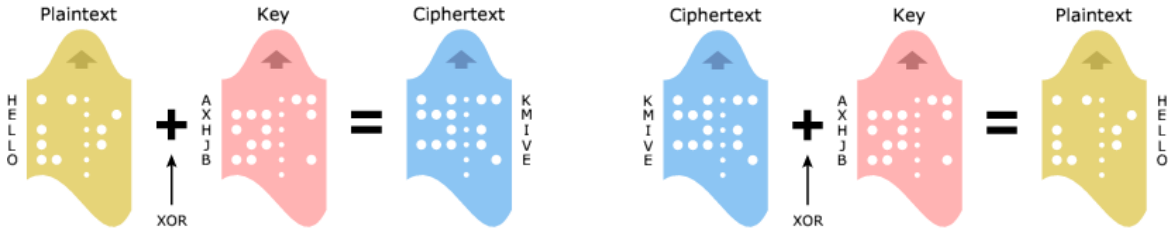
συνέχεια, λαμβάνεται το αντίστοιχο γράμμα του κλειδιού για να βρεθεί η αντίστοιχη σειρά στον Vigenère Square. Το γράμμα που βρίσκεται σε αυτή τη σειρά και τη στήλη είναι το κρυπτογραφημένο γράμμα για τον αρχικό χαρακτήρα (π.χ. plaintext: ATTACK, key: LEMONL το πρώτο γράμμα του ciphertext θα βρίσκεται στην στήλη A και στην γραμμή L που είναι το γράμμα L. Με την ίδια διαδικασία και για τα υπόλοιπα έχουμε τελικά ciphertext: LXFOPV). Παρόλο που ο Vigenère Square παραβιάστηκε το 1863, θεωρήθηκε ως μία από τις ασφαλέστερες μεθόδους επικοινωνίας. (Morkel & Eloff, 2004: 6 & https://en.wikipedia.org/wiki/Vigen%C3%A8re_cipher)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Ο Thomas Jefferson το 1790 εφηύρε τον τροχό κρυπτογράφησης. Δυστυχώς, τα χαρτιά του χάθηκαν και ανακαλύφθηκαν ξανά το 1922. Έτσι, ο Charles Babbage επανεφηύρε τον τροχό κρυπτογράφησης το 1854, χωρίς να ξέρει ότι έχει ήδη εφευρεθεί από κάποιον άλλο. Ο τροχός αποτελούνταν από έναν κύλινδρο ξύλου με 26 δίσκους που θα μπορούσαν να περιστραφούν γύρω από έναν άξονα. Τα γράμματα της αλφαβήτου είχαν εγγραφεί σε κάθε δίσκο, με τυχαία σειρά και οι δίσκοι μπορούσαν να δημιουργήσουν ανακατεμένες λέξεις και στην συνέχεια με την αντίστροφη διαδικασία να φτιάξουν πάλι την αρχική λέξη. Όταν οι δίσκοι είχαν περιστραφεί, για να απεικονίσουν ένα μήνυμα, τότε έπαιρναν το ciphertext από ένα άλλο σημείο του τροχού. Για να αποκρυπτογραφήσει το μήνυμα, ο παραλήπτης έπρεπε να έχει έναν ίδιο τροχό, με τους δίσκους διατεταγμένους με τον ίδιο τρόπο. Παρόμοιες συσκευές χρησιμοποιήθηκαν από τον αμερικανικό στρατό στον Α' Παγκόσμιο Πόλεμο. (Morkel & Eloff, 2004: 6)

Ο παλαιότερος αλγόριθμος κρυπτογράφησης που χρησιμοποιείται ακόμα και σήμερα αναπτύχθηκε από τον Gilbert Vernam το 1917 και ονομάζεται Vernam Cipher. Ο Vernam Cipher είναι μια έκδοση χρήσης μοναδικού μπλοκ (one-time pad), δηλαδή ένας αλγόριθμος που χρησιμοποιεί την μέθοδο της αντικατάστασης, όπου δεν μπορεί να προκύψει κανένα μοτίβο. Ο αποστολέας κρυπτογραφεί ένα μήνυμα χρησιμοποιώντας ένα τυχαία παραγόμενο κλειδί και προσθέτει κάθε κομμάτι του κλειδιού στο αντίστοιχο bit του μηνύματος. Στη συνέχεια, ο δέκτης αποκρυπτογραφεί το μήνυμα αφαιρώντας το ίδιο κλειδί. (Morkel & Eloff, 2004: 7)

Ο Vernam Cipher είναι η μόνη κλασσική τεχνική κρυπτογράφησης που παρέχει τέλεια μυστικότητα, δυστυχώς όμως, έχει κάποια μειονεκτήματα. Επειδή και το plaintext και το κλειδί βρίσκονται μαζί, ένας αντίπαλος θα μπορούσε να συλλέξει κάποιες πληροφορίες από μοτίβο του κωδικοποιημένου μηνύματος, εάν είχε χρησιμοποιηθεί το ίδιο κλειδί πριν. Ένα επιπλέον πρόβλημα αποτελεί, ότι το κλειδί πρέπει να είναι ακριβώς όσο το μήνυμα (ίδιο αριθμό γραμμάτων), το οποίο καθιστά πιο δύσκολη την ασφαλή διανομή του κλειδιού. (Morkel & Eloff, 2004: 7 & http://www.cryptomuseum.com/crypto/vernam.htm)



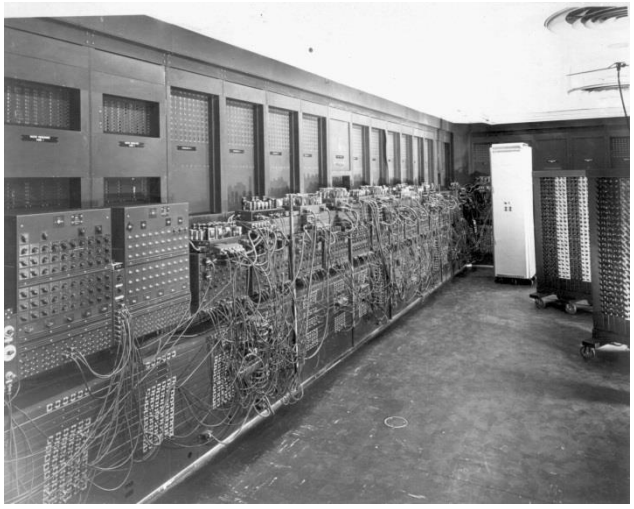
Το 1923 ο Arthur Scerbius δημιούργησε την μηχανή Enigma. Στην συνέχεια, η γερμανική κυβέρνηση ανέλαβε την μηχανή και βελτιώνοντάς την δημιούργησε τη μηχανή TYPEX που

χρησιμοποιήθηκε κατά τον Β’ Παγκόσμιο Πόλεμο. Η μηχανή αποτελείται από πέντε τροχούς που άλλαζαν τα γράμματα της αλφαβήτου. Η αντιστροφή της διαδικασίας θα μπορούσε να αποκρυπτογραφήσει το μήνυμα. Η κωδικοποίηση της μηχανής Enigma “έσπασε” στη δεκαετία του 1930 από τον πολωνό μαθηματικό Marian Rejewski. Η μηχανή Enigma που λειτουργούσε με τροχούς χρησιμοποιήθηκε ως βάση για πολλές μηχανές κρυπτογράφησης, αλλά όλες αυτές έχουν παραβιαστεί. (Morkel & Eloff, 2004: 7)

Ένα παράδειγμα κατά το οποίο προφορική και γραπτή γλώσσα χρησιμοποιήθηκε ως συσκευή κρυπτογράφησης, είναι οι Navajo windtalkers που χρησιμοποίησε ο αμερικανικός στρατός το 1942 κατά τον Β’ Παγκόσμιο Πόλεμο. Αυτοί ήταν Αμερικανοί στρατιώτες που είχαν μετατρέψει μηνύματα της μητρικής τους γλώσσας, σε μια γλώσσα τόσο περίπλοκη, που ο εχθρός δεν μπορούσε να καταλάβει το μήνυμα. (Morkel & Eloff, 2004: 7)

Letter	Navajo Word	Meaning
A	Wol-la-chee	Ant
B	Shush	Bear
C	Moasi	cat
D	Be	Deer
E	Dzeh	Elk
F	Ma-e	Fox
G	Klizzie	Goat
H	Lin	Horse
I	Tkin	Ice
J	Tkele-cho-gi	Jackass
K	Klizzie-yazzie	Kid
L	Dibeh-yazzie	Lamb
M	Na-as-tso-si	Mouse
N	Nesh-chee	Nut
O	Ne-ahs-jah	Owl
P	Bi-so-dih	Pig
Q	Ca-yeilth	Quiver
R	Gah	Rabbit
S	Dibeh	Sheep
T	Than-zie	Turkey
U	No-da-ih	Ute
V	A-keh-di-glini	Victor
W	Gloe-ih	Weasel
X	Al-an-as-dzoh	Cross
Y	Tsah-as-zih	Yucca
Z	Besh-do-gliz	Zinc

Μεταξύ 1943 και 1945, κατασκευάστηκε ο πρώτος ηλεκτρονικός υπολογιστής γενικής χρήσης, που αναφέρεται ως Electronic Numerical Integrator and Computer (ENIAC). Οι άνθρωποι που σχεδίασαν αυτή την πρωτοποριακή τεχνολογία ήταν ο John Mauchly, ο J. Presper Eckert και ο υπολοχαγός Herman Goldstine. Ο ENIAC σχεδιάστηκε αρχικά, για να βοηθήσει με σύνθετες μαθηματικές λειτουργίες στον Β' Παγκόσμιο Πόλεμο και μπορούσε να εκτελεί υπολογισμούς μέχρι και χίλιες φορές ταχύτερα από τον προκάτοχό του (μηχανικός υπολογιστής) και όρισε την έναρξη της σύγχρονης εποχής των υπολογιστών. (Morkel & Eloff, 2004: 7)



2.2 Σύγχρονες Τεχνικές Κρυπτογράφησης

Με την δημιουργία του πρώτου υπολογιστή, θα δούμε τεράστιες αλλαγές στις τεχνικές κρυπτογράφησης. Οι σύγχρονες τεχνικές κρυπτογράφησης σχεδιάστηκαν ειδικά για τη χρήση τους από ηλεκτρονικούς υπολογιστές και χρησιμοποίησαν τα bits, αντί για το αλφάβητο που είχε κάθε γλώσσα όπως είδαμε στις κλασσικές τεχνικές κρυπτογράφησης. Έτσι, δεν ήταν πλέον εφικτό να αναπτυχθούν κλασσικοί αλγόριθμοι κρυπτογράφησης βασισμένοι σε στυλό και χαρτί, καθώς αλγόριθμοι που κάποτε ήταν δύσκολο να λυθούν, μπορούσαν να "σπάσουν" μέσα σε σύντομο χρονικό διάστημα χρησιμοποιώντας υπολογιστές.

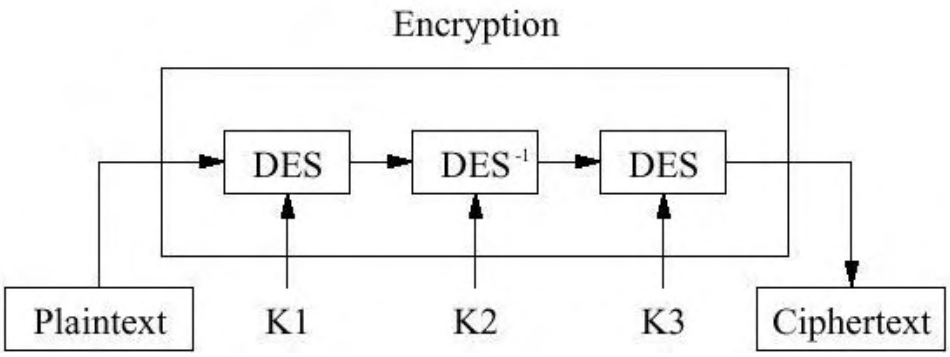
Ένα από τα κύρια προβλήματα των κλασσικών τεχνικών κρυπτογράφησης ήταν το γεγονός ότι αν θέλαμε να επικοινωνήσουμε κρυφά με περισσότερα από ένα άτομα, θα έπρεπε να έχουμε ξεχωριστή μυστική γλώσσα για κάθε άτομο. Αυτό δεν θα ήταν πολύ πρακτικό και οι επιστήμονες αναγκάστηκαν να σχεδιάσουν νέους αλγορίθμους. Η λύση θα ήταν τυποποιημένοι αλγόριθμοι και ο τρόπος λειτουργίας του κάθε αλγορίθμου θα ανακοινωνόταν δημόσια και η μυστικότητα του μηνύματος θα βασιζόταν σε έναν άλλο παράγοντα. Έτσι, σχεδιάστηκαν τα κρυπτογραφικά κλειδιά (cipher keys). Κάθε μήνυμα έχει και ένα κρυπτογραφικό κλειδί, που μοιραζόταν ο αποστολέας και ο παραλήπτης. Αυτό το κλειδί το χρησιμοποιούσαν όταν κρυπτογραφούσαν και αποκρυπτογραφούσαν το μήνυμα και χωρίς αυτό κανείς δεν θα μπορούσε να αποκρυπτογραφήσει το μήνυμα. Το κρυπτογραφικό κλειδί είναι ένα σημαντικό χαρακτηριστικό των σύγχρονων τεχνικών κρυπτογράφησης. Οι σύγχρονες τεχνικές κρυπτογράφησης μπορούν να χωριστούν σε δύο κατηγορίες: την συμμετρική κρυπτογράφηση και την ασύμμετρη κρυπτογράφηση. (Morkel & Eloff, 2004: 8)

Συμμετρική Κρυπτογράφηση

Τα συμμετρικά κρυπτοσυστήματα απαιτούν από τον αποστολέα και τον παραλήπτη να έχουν το ίδιο μυστικό κλειδί. Αυτό το κλειδί απαιτείται τόσο για την κρυπτογράφηση όσο και για την αποκρυπτογράφηση του μηνύματος.

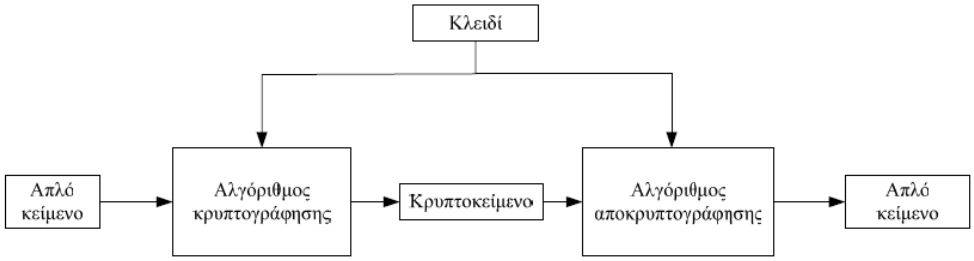
Ένα κλασικό πρότυπο μεταξύ των συμμετρικών αλγορίθμων είναι το Data Encryption Standard (DES). Το DES αναπτύχθηκε το 1970 και πήρε την επίσημη έγκριση του The United States National Institute of Standards and Technology (NIST) το 1977. Το DES χρησιμοποιεί αντικατάσταση και μετάθεση για να ανακατεύει τα bits ενός μηνύματος. Σήμερα, το DES θεωρείται ως μια αδύναμη μέθοδος κρυπτογράφησης, δεδομένου ότι εξαρτιόταν από μία μηχανή που κατασκευάστηκε από την οργάνωση Electronic Frontier Foundation (EFF) το 1998. Το μηχάνημα Deep Crack "έσπασε" τον αλγόριθμο σε 4,5 ημέρες. Χρησιμοποιούσε 19-δισεκατομμύρια κλειδιά ανά δευτερόλεπτο για να προσπαθήσει να μαντέψει το σωστό κλειδί. Το 1999, ένα έργο στο Διαδίκτυο ήταν σε θέση να δοκιμάσει 250 δισεκατομμύρια κλειδιά ανά δευτερόλεπτο, με αποτέλεσμα το DES να "σπάσει" σε λίγες ώρες. (Morkel & Eloff, 2004: 8)

Στην συνέχεια, αναπτύχθηκε το Triple Data Encryption Standard (3DES), ως βελτίωση του αλγόριθμου DES. Χρησιμοποιεί τρία κλειδιά διαδοχικά, μαζί με τρεις διαφορετικές λειτουργίες κρυπτογράφησης και σίγουρα είναι πολύ πιο ασφαλές από τον DES.

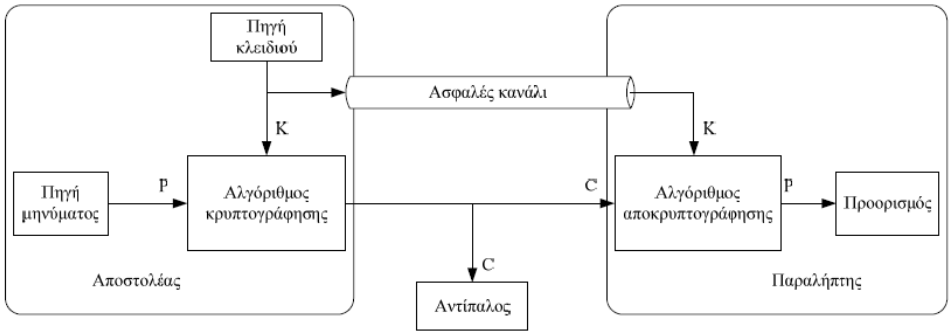


Μετά τον 3DES ανακαλύφθηκε το Advanced Encryption Standard (AES). Το AES βασίζεται στον αλγόριθμο Rijndael που επιλέχθηκε από μια λίστα υποψηφίων από την NIST. Το AES βασίζεται επίσης, στη μεταφορά των bits ενός μηνύματος σε συνδυασμό με το κρυπτογραφικό κλειδί. Το 3DES εξακολουθεί να χρησιμοποιείται σήμερα, αλλά θεωρείται αλγόριθμος κρυπτογράφησης παλαιού τύπου. Το DES είναι εγγενώς ανασφαλές, ενώ το 3DES έχει πολύ καλύτερα χαρακτηριστικά ασφαλείας, αλλά εξακολουθεί να θεωρείται προβληματικό. Το NIST είναι ο κυβερνητικός οργανισμός που τυποποιεί τους κρυπτογραφικούς αλγόριθμους. Ο πιο συνηθισμένος αλγόριθμος κρυπτογράφησης συμμετρικού κλειδιού του NIST είναι ο AES (το πιο

προηγμένο πρότυπο κρυπτογράφησης). Στην πραγματικότητα, υπήρχαν αρκετές καλές υποψηφιότητες για να πάρουν την θέση ως ο αλγόριθμος του AES, συμπεριλαμβανομένου του αλγορίθμου Rijndael που έγινε ο αλγόριθμος του AES, ο αλγόριθμος Blowfish του Bruce Schneier, ο αλγόριθμος RC6, ο αλγόριθμος MARS, ο αλγόριθμος Twofish και ο αλγόριθμος του Serpent. (Morkel & Eloff, 2004: 8 & https://stackoverflow.com/questions/1619212/is-des-or-3des-still-being-used-today)



Το κύριο πρόβλημα με τη συμμετρική κρυπτογράφηση είναι ότι, εάν το κλειδί χαθεί ή κλαπεί, ολόκληρη η μετάδοση μπορεί να μη θεωρείται ασφαλής, αφού οι αντίπαλοι μπορούν αμέσως να αποκρυπτογραφήσουν το μήνυμα με το ένα κλειδί. Αυτό οδηγεί σε ένα άλλο πρόβλημα που είναι η κατανομή των κλειδιών. Ένα κλειδί πρέπει, είτε να ανταλλάσσεται αυτοπροσώπως, είτε να μεταδίδεται μέσω ενός πολύ ασφαλούς καναλιού. (Κάτος & Στεφανίδης, 2003: 4-5)



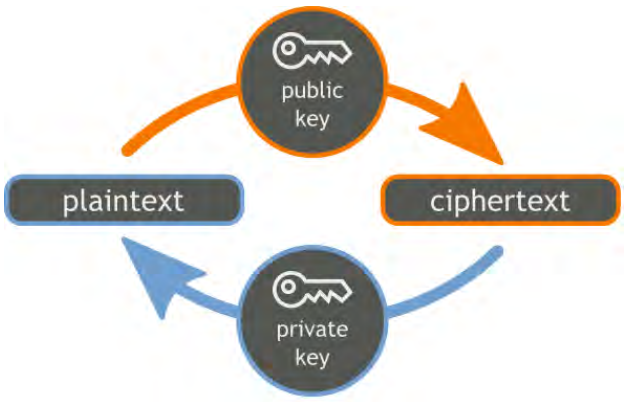
Ασύμμετρη Κρυπτογράφηση

Οι ασύμμετρες μέθοδοι κρυπτογράφησης, που επίσης αναφέρονται ως δημόσιου κλειδιού (public key) συστήματα κρυπτογράφησης, αναπτύχθηκαν το 1976 από τους Whitefield Diffie και Martin Hellman. Η αρχή της κρυπτογράφησης δημόσιου κλειδιού είναι ότι και τα δύο μέρη, ο

αποστολέας καθώς και ο παραλήπτης, έχουν ένα ζεύγος κλειδιών. Το μόνο κλειδί που δεν πρέπει να κρατηθεί μυστικό είναι το δημόσιο κλειδί. Τα δύο διαφορετικά κλειδιά που κατέχουν οι χρήστες έχουν διαφορετικές χρήσεις, το ένα χρησιμοποιείται για την κρυπτογράφηση και το άλλο για την αποκρυπτογράφηση. Το κλειδί κρυπτογράφησης είναι το δημόσιο κλειδί, ενώ το κλειδί αποκρυπτογράφησης είναι το ιδιωτικό κλειδί (private key). Το ιδιωτικό κλειδί πρέπει να φυλάσσεται μυστικό. (Morkel & Eloff, 2004: 9)

Το δημόσιο και το ιδιωτικό κλειδί σχετίζονται μαθηματικά, έτσι ώστε οτιδήποτε κρυπτογραφημένο με το ένα, να μπορεί να αποκρυπτογραφηθεί με το άλλο. Ο αποστολέας παίρνει το κλειδί του παραλήπτη (public key), το οποίο είναι δημοσίως διαθέσιμο σε έναν Web Server για παράδειγμα, και κρυπτογραφεί το μήνυμα. Στη συνέχεια, το στέλνει στον παραλήπτη ο οποίος μπορεί να αποκρυπτογραφήσει το μήνυμα με το ιδιωτικό του κλειδί (private key). Το κύριο πλεονέκτημα αυτής της μεθόδου είναι ότι ο αποστολέας και ο δέκτης δεν χρειάζεται να ανταλλάσσουν κλειδιά συνέχεια. (Morkel & Eloff, 2004: 9)

Η πρώτη εφαρμογή ενός κρυπτοσυστήματος δημόσιου κλειδιού αναπτύχθηκε το 1978 από τους Ronald Rivest, Adi Shamir και Leonard Adleman και ονομάστηκε αλγόριθμος RSA. Το RSA χρησιμοποιεί μία λειτουργία μονής κατεύθυνσης που βασίζεται στον πολλαπλασιασμό πρώτων αριθμών, για να προσδιορίσει το κλειδί και βασίζεται στο γεγονός ότι είναι πολύ δύσκολο να παραγοντοποιήσουμε έναν μεγάλο αριθμό σε δύο πρώτους αριθμούς. Η πολυπλοκότητα αυτού του μαθηματικού προβλήματος αυξάνεται εκθετικά, όσο μεγαλύτεροι είναι οι αριθμοί και για το λόγο αυτό, το μέγεθος του κλειδιού του RSA είναι συνήθως μεγάλο. Ανεξάρτητα από αυτό, το RSA θεωρείται ένα πολύ ασφαλές σύστημα και χρησιμοποιείται ευρέως σήμερα, ειδικά για τη διανομή κλειδιών (π.χ. διανομή κλειδιού του AES). (Morkel & Eloff, 2004: 9-10)



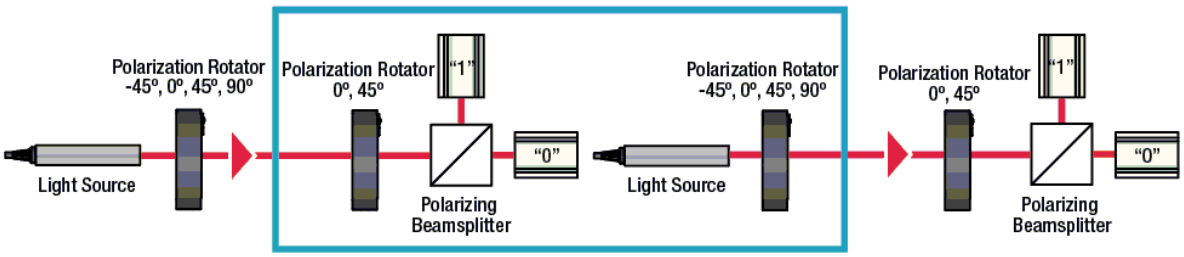
Η κρυπτογράφηση δημόσιου κλειδιού λύνει το πρόβλημα της διανομής κλειδιών της συμμετρικής κρυπτογράφησης, αλλά δυστυχώς, με πιθανά μελλοντικά προβλήματα. Η δυσκολία των μαθηματικών λειτουργιών που βασίζεται στην κρυπτογράφηση του δημόσιου κλειδιού, μπορεί να θεωρηθεί σχετική, γιατί προς το παρόν δεν υπάρχει ένας μαθηματικός αλγόριθμος, που να μπορεί να συντελέσει έναν αριθμό σε δύο πρώτους αριθμούς γρήγορα, αλλά, εάν ένας μαθηματικός αναπτύξει έναν τέτοιο αλγόριθμο, το σύστημα RSA θα "σπάσει" και όσοι οργανισμοί-ιδρύματα τον χρησιμοποιούν δεν θα είναι ασφαλείς. (Morkel & Eloff, 2004: 10)

Κβαντική Κρυπτογράφηση

Το 1984 ο Giles Brassard και ο Charles Bennett δημοσίευσαν το πρωτόκολλο BB84, βασισμένο σε μια ιδέα του Stephen Weisner, που προτάθηκε στη δεκαετία του 1970, το οποίο θα χρησιμοποιούσε την κβαντική μηχανική για να λύσει το πρόβλημα της διανομής κλειδιών. Η πρώτη εφαρμογή του πρωτοκόλλου BB84 αναπτύχθηκε το 1991, αλλά μόνο για απόσταση 32 εκατοστών.

Η κβαντική κρυπτογράφηση χρησιμοποιεί φωτεινά σωματίδια, που ονομάζονται φωτόνια (photons), για να επικοινωνούν, αντί για bits. Ένα φωτόνιο μπορεί να έχει έναν από τους τέσσερις προσανατολισμούς, είτε οριζόντιο, κάθετο, 45° διαγώνιο και -45° διαγώνιο. Καθένα από αυτά αντιπροσωπεύει ένα bit: - και / αντιπροσωπεύει ένα bit 0, ενώ | και \ αντιπροσωπεύει ένα bit 1. Κάθε bit σε ένα μήνυμα μεταφράζεται τυχαία σε έναν από τους δύο προσανατολισμούς που συνδέονται με αυτό το bit. Τα πραγματικά bits αποστέλλονται στον παραλήπτη μέσω οπτικών ινών (optic fibers). Ο παραλήπτης με τη σειρά του έχει δύο φίλτρα: ένα + (ευθύγραμμο) και ένα x (διαγώνιο) φίλτρο. Αν ένα κατακόρυφο ή οριζόντιο φωτόνιο μετακινείται μέσω ενός ευθύγραμμου φίλτρου, παραμένει το ίδιο, αλλά όταν ένα διαγώνιο φωτόνιο κινηθεί μέσα από αυτό, θα αλλάξει. Αυτά τα φίλτρα επιλέγονται τυχαία στην μεριά του παραλήπτη για κάθε φωτόνιο. Τα αποτελέσματα που δημιουργούνται κρατούνται μυστικά από τον παραλήπτη. Η ακολουθία των φίλτρων που χρησιμοποιήθηκαν αποστέλλεται πίσω στον αποστολέα όπου συγκρίνονται με τα απεσταλμένα φωτόνια. Οι θέσεις όπου χρησιμοποιήθηκαν τα σωστά φίλτρα αποστέλλονται και πάλι στον παραλήπτη και τα bits που προκύπτουν

χρησιμοποιούνται ως κλειδί. Τα κομμάτια που άλλαξαν απορρίπτονται. Η κβαντική κρυπτογράφηση είναι η πρώτη μέθοδος κρυπτογράφησης αυτού του είδους. Η τεχνολογία βρίσκεται ακόμη σε φάση ανάπτυξης, παρόλο που η εταιρεία με έδρα τη Νέα Υόρκη, MagiQ Technologies ισχυρίζεται ότι έχει αναπτύξει το πρώτο εμπορικά διαθέσιμο κβαντικό σύστημα κρυπτογράφησης. (Morkel & Eloff, 2004: 10-11)



3. ΣΥΓΧΡΟΝΗ ΚΡΥΠΤΟΓΡΑΦΙΑ

Όπως αναφέραμε και στο προηγούμενο κεφάλαιο η σύγχρονη κρυπτογραφία λειτουργεί με βάση τις δυαδικές ακολουθίες bit. Βασίζεται σε γνωστούς μαθηματικούς αλγορίθμους για την κρυπτογράφηση των πληροφοριών. Η μυστικότητα επιτυγχάνεται μέσω του μυστικού κλειδιού, το οποίο χρησιμοποιείται ως είσοδος για τους συγκεκριμένους αλγορίθμους. Η υπολογιστική δυσκολία των αλγορίθμων, η απουσία μυστικού κλειδιού, αλλά και άλλοι παράγοντες καθιστούν σχεδόν αδύνατο για έναν αντίπαλο να αποκτήσει το αρχικό μήνυμα ακόμα και αν ξέρει τον αλγόριθμο που χρησιμοποιείται για την κρυπτογράφηση. Η σύγχρονη κρυπτογραφία απαιτεί από τα μέρη που ενδιαφέρονται για ασφαλή επικοινωνία να κατέχουν μόνο το μυστικό κλειδί. Το κρυπτογραφικό κλειδί είναι ένα σημαντικό χαρακτηριστικό των σύγχρονων τεχνικών κρυπτογράφησης. Οι σύγχρονες τεχνικές κρυπτογράφησης μπορούν να χωριστούν σε δύο κατηγορίες: την συμμετρική κρυπτογράφηση και την ασύμμετρη κρυπτογράφηση. (Menezes, Oorschot & Vanstone , 1996: 1-4)

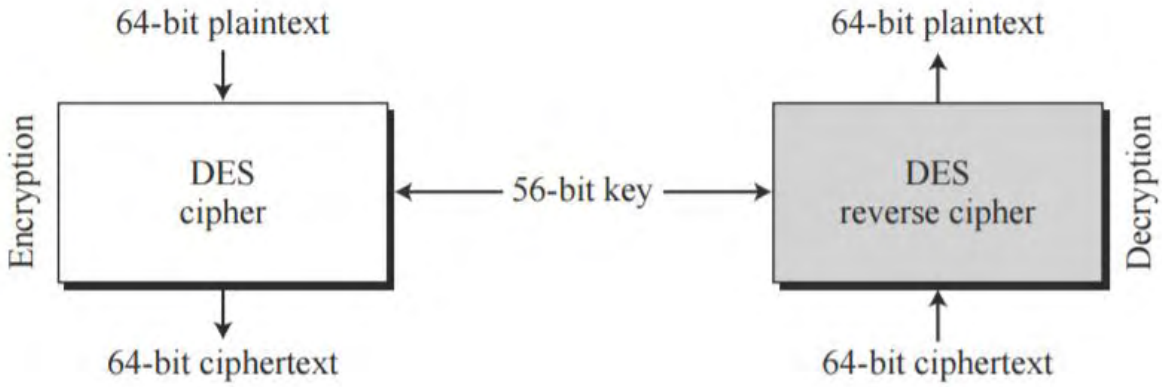
3.1 Συμμετρική Κρυπτογράφηση

Τα συμμετρικά κρυπτοσυστήματα απαιτούν από τον αποστολέα και τον παραλήπτη να έχουν το ίδιο μυστικό κλειδί. Αυτό το κλειδί απαιτείται τόσο για την κρυπτογράφηση όσο και για την αποκρυπτογράφηση του μηνύματος.

3.1.1 Data Encryption Standard (DES)

Ο αλγόριθμος DES έχει σχεδιαστεί για να κρυπτογραφεί το μήνυμα σε blocks δεδομένων που αποτελούνται από 64bits με την χρήση ενός κλειδιού μεγέθους 64bits (56bits το κλειδί και 8 parity bits), όπου το ίδιο ισχύει και για την αποκρυπτογράφηση. Η αποκρυπτογράφηση επιτυγχάνεται χρησιμοποιώντας το ίδιο κλειδί όπως στην κρυπτογράφηση, με την διαδικασία

αποκρυπτογράφησης να είναι η αντίστροφη της διαδικασίας της κρυπτογράφησης. (PUB, 1999: 1-2)

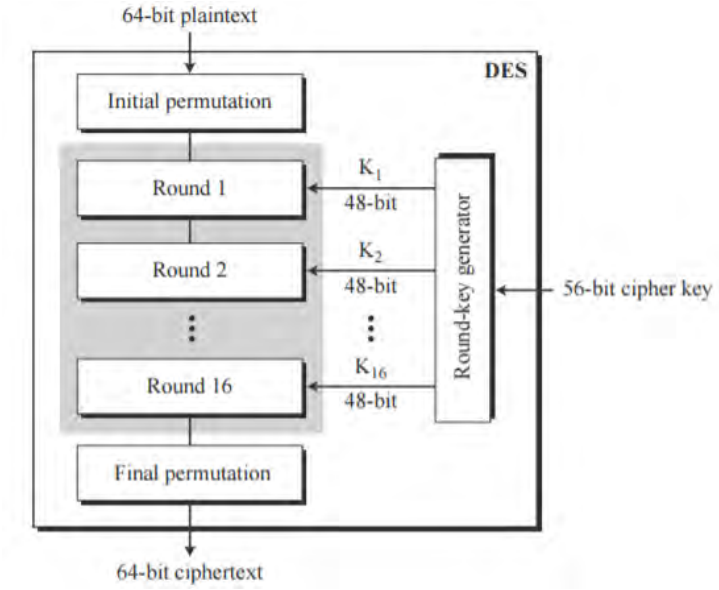


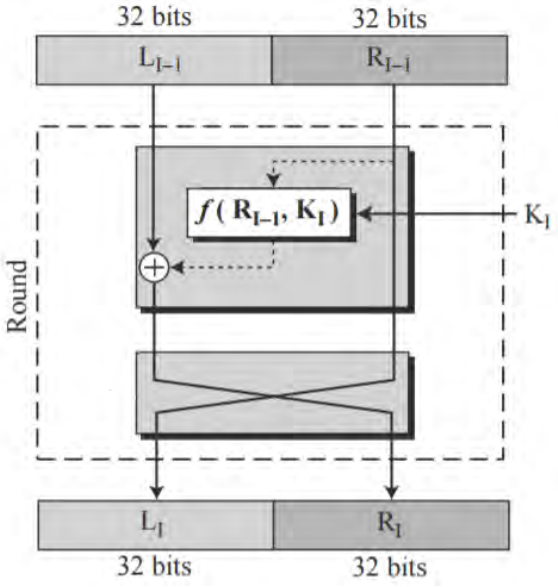
Στην συνέχεια για την κρυπτογράφηση ακολουθούνται τα παρακάτω βήματα:

1. Initial Permutation (IP)
2. Expansion (D-Box)
3. XOR
4. S-Box
5. Permutation
6. Αντιμετάθεση των τελικών 32bits
7. Final Permutation (IP^{-1})

(βλ. Ντούσκας, 2017: 15)

Για να γίνουν οι παραπάνω διαδικασίες ο DES χρησιμοποιεί δομή δικτύου Feistel 16 επιπέδων όπως φαίνεται με απλό τρόπο στις παρακάτω εικόνες.





Πιο αναλυτικά για τα παραπάνω 7 βήματα έχουμε:

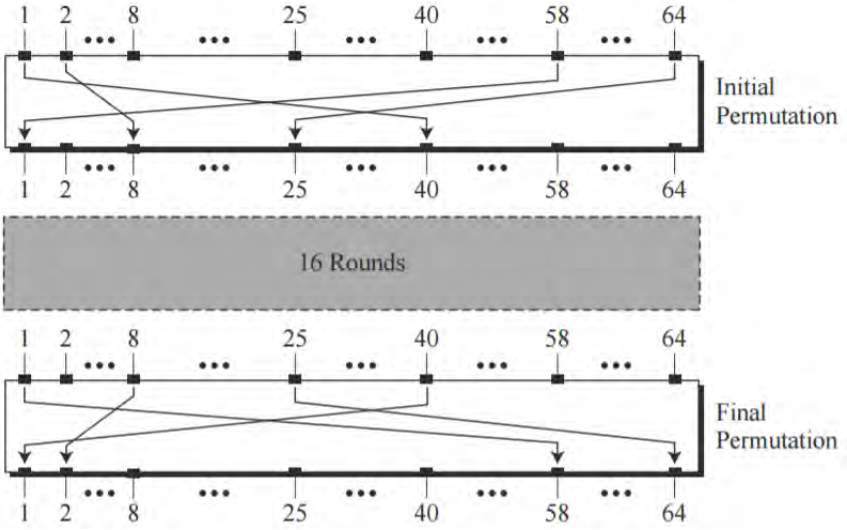
- **Initial και Final Permutation**: αυτές οι διαδικασίες αλλάζουν τις θέσεις των bits με βάση τους πίνακες IP και IP^{-1} όπως φαίνονται παρακάτω:

Initial Permutation	*Final Permutation*
58 50 42 34 26 18 10 02	40 08 48 16 56 24 64 32
60 52 44 36 28 20 12 04	39 07 47 15 55 23 63 31
62 54 46 38 30 22 14 06	38 06 46 14 54 22 62 30
64 56 48 40 32 24 16 08	37 05 45 13 53 21 61 29
57 49 41 33 25 17 09 01	36 04 44 12 52 20 60 28
59 51 43 35 27 19 11 03	35 03 43 11 51 19 59 27
61 53 45 37 29 21 13 05	34 02 42 10 50 18 58 26
63 55 47 39 31 23 15 07	33 01 41 09 49 17 57 25

Για οποιαδήποτε αλλαγή γίνεται από πίνακες μετάθεσης που έχουμε επισημάνει μέχρι στιγμής αλλά και για όλους τους υπόλοιπους που θα εξετάσουμε, η τιμή του κάθε στοιχείου

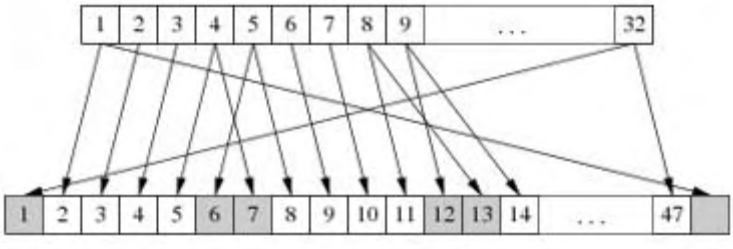
προσδιορίζει τον αριθμό εισόδου και η θέση του στοιχείου στον πίνακα ορίζει τον αριθμό θέσης της εξόδου.

- Στην συνέχεια θα δούμε την συνάρτηση f (f function), η οποία περιέχει την εκτέλεση των βημάτων 2 (Expansion), 3 (XOR), 4 (S-Box) και 5 (Permutation). Η συνάρτηση f ορίζεται ως $f(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$.



- **Expansion (D-Box)**: το R_{i-1} που είναι η είσοδος (In) στον γύρο i επεκτείνεται ($E(R_{\iota-1})$) από τα 32bits στα 48bits με βάση τον παρακάτω πίνακα:

E					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1



- **XOR**: γίνεται η πράξη $E(R_{i-1}) \oplus K_i$ όπου K_i είναι το κλειδί του γύρου i, με αποτέλεσμα 48bits.
- **S-Box**: σε αυτό το σημείο τα 48bits χωρίζονται σε οκτώ 6-bit strings για να περάσουν μέσα από 8 S-Boxes. Το κάθε S-Box παίρνει σαν είσοδο 6bits και με βάση τον παρακάτω πίνακα βγάζει σαν έξοδο 4bits, έτσι σύνολο έχουμε από 48bits → 32bits.

S_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

S_2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	01	08	14	06	11	03	04	09	07	02	13	12	00	05	10
1	03	13	04	07	15	02	08	14	12	00	01	10	06	09	11	05
2	00	14	07	11	10	04	13	01	05	08	12	06	09	03	02	15
3	13	08	10	01	03	15	04	02	11	06	07	12	00	05	14	09

S_3	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	10	00	09	14	06	03	15	05	01	13	12	07	11	04	02	08
1	13	07	00	09	03	04	06	10	02	08	05	14	12	11	15	01
2	13	06	04	09	08	15	03	00	11	01	02	12	05	10	14	07
3	01	10	13	00	06	09	08	07	04	15	14	03	11	05	02	12

S_4	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	07	13	14	03	00	06	09	10	01	02	08	05	11	12	04	15
1	13	08	11	05	06	15	00	03	04	07	02	12	01	10	14	09
2	10	06	09	00	12	11	07	13	15	01	03	14	05	02	08	04
3	03	15	00	06	10	01	13	08	09	04	05	11	12	07	02	14

S_5	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	02	12	04	01	07	10	11	06	08	05	03	15	13	00	14	09
1	14	11	02	12	04	07	13	01	05	00	15	10	03	09	08	06
2	04	02	01	11	10	13	07	08	15	09	12	05	06	03	00	14
3	11	08	12	07	01	14	02	13	06	15	00	09	10	04	05	03

S_6	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	12	01	10	15	09	02	06	08	00	13	03	04	14	07	05	11
1	10	15	04	02	07	12	09	05	06	01	13	14	00	11	03	08
2	09	14	15	05	02	08	12	03	07	00	04	10	01	13	11	06
3	04	03	02	12	09	05	15	10	11	14	01	07	06	00	08	13

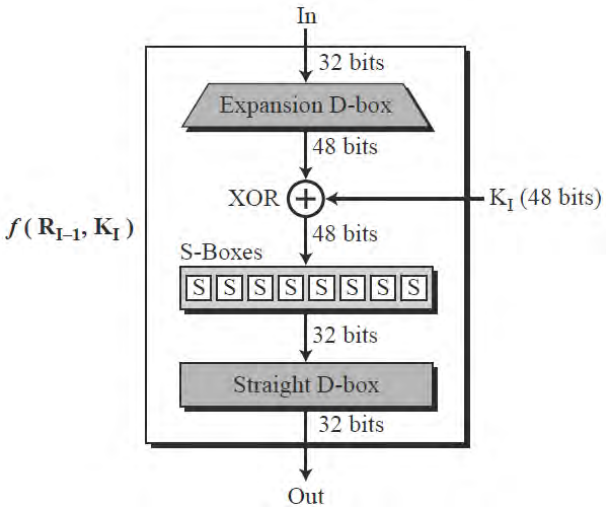
S_7	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	04	11	02	14	15	00	08	13	03	12	09	07	05	10	06	01
1	13	00	11	07	04	09	01	10	14	03	05	12	02	15	08	06
2	01	04	11	13	12	03	07	14	10	15	06	08	00	05	09	02
3	06	11	13	08	01	04	10	07	09	05	00	15	14	02	03	12

S_8	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	13	02	08	04	06	15	11	01	10	09	03	14	05	00	12	07
1	01	15	13	08	10	03	07	04	12	05	06	11	00	14	09	02
2	07	11	04	01	09	12	14	02	00	06	10	13	15	03	05	08
3	02	01	14	07	04	10	08	13	15	12	09	00	03	05	06	11

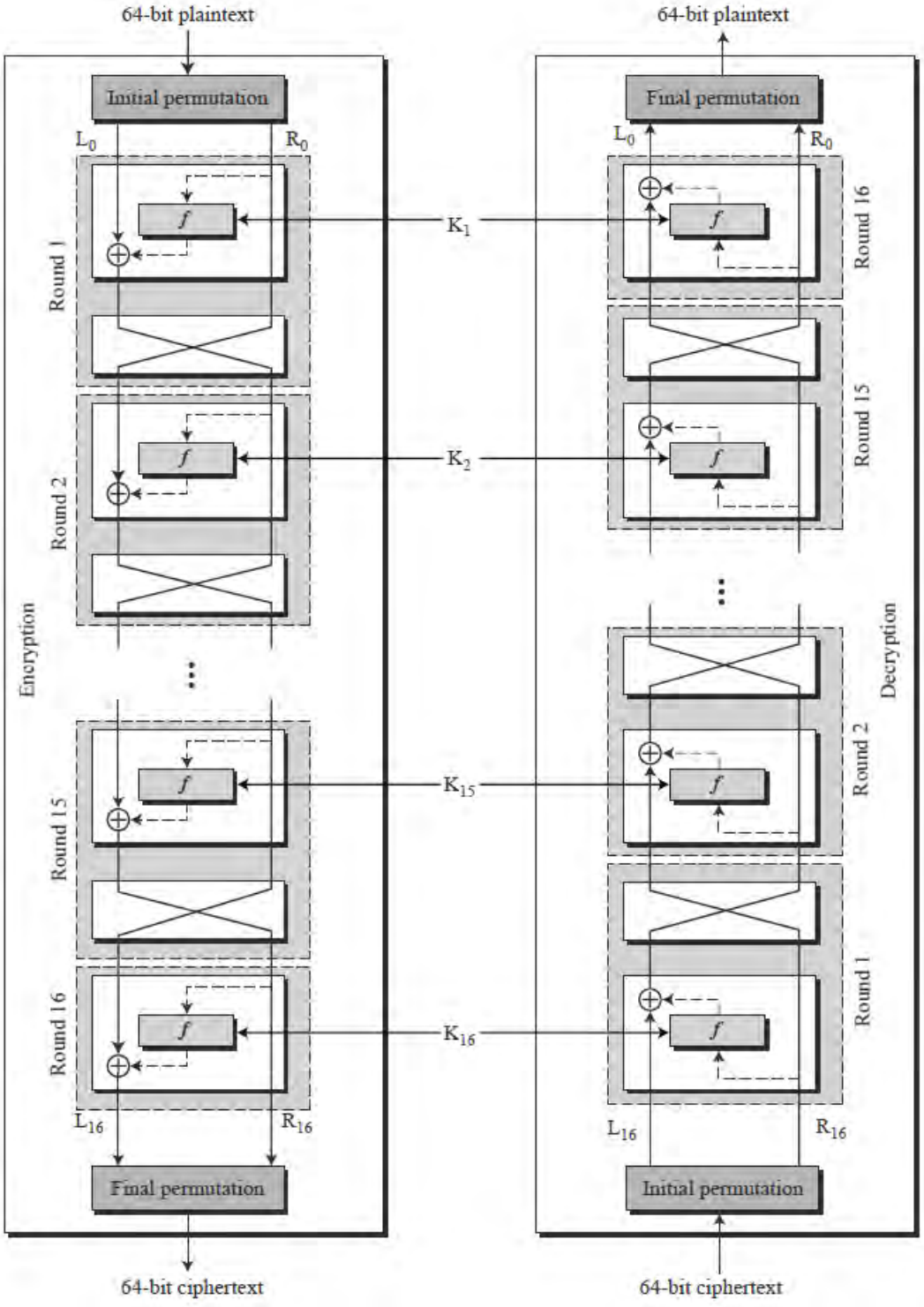
Η αντικατάσταση στα S-Boxes γίνεται παίρνοντας το πρώτο και το τελευταίο (έκτο) bit της εισόδου και ο συνδυασμός του μας προσδιορίζει την γραμμή στον πίνακα του συγκεκριμένου S-Box, αντίστοιχα τα υπόλοιπα 4bits μας προσδιορίζουν την στήλη.

- **Permutation**: είναι μια μετάθεση με είσοδο 32bits και έξοδο 32bits. Ο πίνακας που ορίζει τις μεταθέσεις είναι:

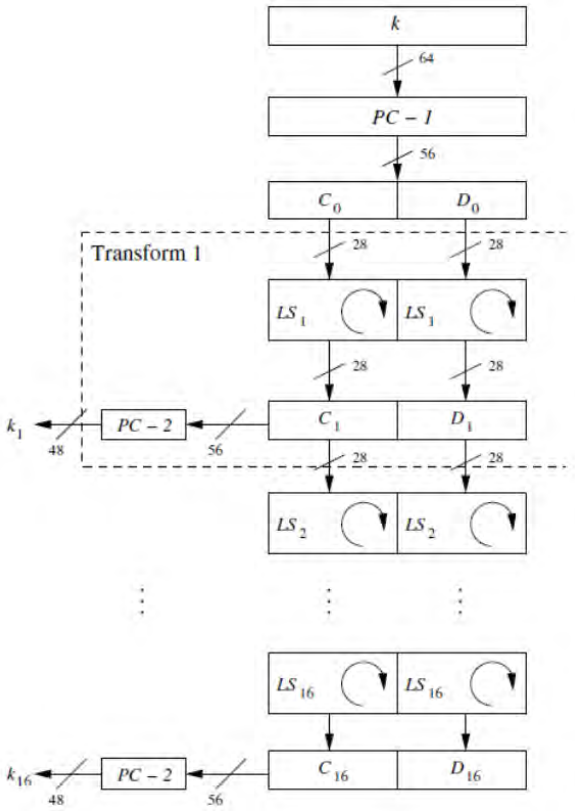
P							
16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25



- Τέλος έχουμε την αντιμετάθεση των τελικών 32bits (R_{16}, L_{16}) και στην συνέχεια το Final Permutation που εξηγήσαμε προηγουμένως.

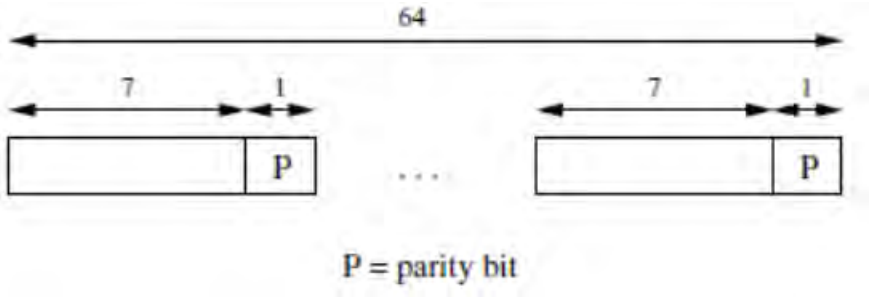


Στο σημείο αυτό θα αναφερθούμε στην διαδικασία παραγωγής κλειδιών (Key Schedule) που χρησιμοποιεί ο παραπάνω αλγόριθμος για κάθε γύρο.



Η όλη διαδικασία ακολουθεί 3 βήματα:

1. Το κλειδί κρυπτογράφησης δίνεται κανονικά ως κλειδί 64bits, στο οποίο 8 επιπλέον bits είναι τα parity bits (ή αλλιώς bits ισοτιμίας), τα οποία δεν χρησιμοποιούνται στην διαδικασία δημιουργίας κλειδιού.



2. Τώρα γίνεται μετάθεση των 64bits και απαλοιφή των parity bits με αποτέλεσμα να έχουμε έξοδο 56bits. Οι μεταθέσεις φαίνονται από τον παρακάτω πίνακα (Permutation Choice 1 - PC1):

57	49	41	33	25	17	09	01
58	50	42	34	26	18	10	02
59	51	43	35	27	19	11	03
60	52	44	36	63	55	47	39
31	23	15	07	62	54	46	38
30	22	14	06	61	53	45	37
29	21	13	05	28	20	12	04

Μετά την μετάθεση γίνεται χωρισμός της εξόδου σε δύο κομμάτια (C,D) μεγέθους 28bits το καθένα.

3. Στο τελευταίο βήμα αρχικά γίνεται αριστερή ολίσθηση κατά S_i bits στα C_{i-1} και D_{i-1}, όπου:

$$S_i = \begin{cases} 1 \ \gamma\iota\alpha \ i = 1, 2, 9, 16 \\ 2 \ \alpha\lambda\lambda\iota\acute{\omega}\varsigma \end{cases}$$

Στην συνέχεια γίνεται η τελευταία μετάθεση όπου από τα 56bits τα 8bit αγνοούνται (συμπιέζονται) και με τον παρακάτω πίνακα (Permutation Choice 2 – PC2) έχουμε το τελικό κλειδί του γύρου μεγέθους 48bits.

14	17	11	24	01	05	03	28
15	06	21	10	23	19	12	04
26	08	16	07	27	20	13	02
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

(Πηγή:http://highered.mheducation.com/sites/dl/free/007070208x/877405/Chapter_06_Data_Encription_Standard.pdf)

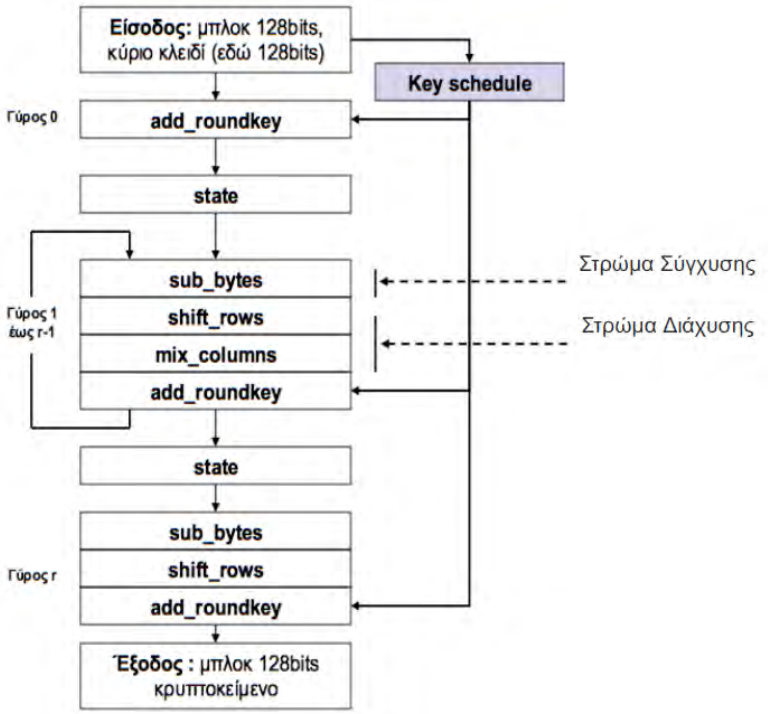
3.1.2 Advanced Encryption Standard (AES)

Ο AES δεν βασίζεται στα δίκτυα Feistel, αλλά στην χρήση blocks, όπου κάθε block για οποιοδήποτε μέγεθος κλειδιού έχει σταθερό μέγεθος 128bit (16 bytes). Τα κλειδιά που μπορεί να χρησιμοποιήσει, για να κρυπτογραφήσει-αποκρυπτογραφήσει είναι μεταβλητού μεγέθους. Το μέγεθος αυτό μπορεί να είναι 128 ή 192 ή 256 bits και από το μέγεθος του κλειδιού εξαρτάτε και πόσους γύρους (rounds) θα εκτελέσει ο αλγόριθμος. Συγκεκριμένα, για τους γύρους έχουμε: $r_{128} = 10, r_{192} = 12$ και $r_{256} = 14$. Από το μέγεθος των κλειδιών βλέπουμε ότι είναι αδύνατον να "σπάσει" ο αλγόριθμος, αφού για παράδειγμα, για κλειδί 128bits έχουμε $3.4 * 10^{38}$ πιθανούς συνδυασμούς που πρέπει να γίνουν, για να βρεθεί το σωστό κλειδί, πράγμα που για τους σημερινούς υπολογιστές θα χρειαστεί αρκετά δισεκατομμύρια χρόνια. Όσο για την αποκρυπτογράφηση του μηνύματος, προκύπτει κάνοντας αναστροφή τον αλγόριθμο (φθίνουσα αρίθμηση των γύρων, ώστε να ισχύει η αντιστοιχία με τα κλειδιά του κάθε γύρου) και αντικατάσταση όλων των συναρτήσεων με τις αντίστροφές τους. (Standard, 2001: 13-14)

Ας μιλήσουμε τώρα για τα χαρακτηριστικά και τον τρόπο λειτουργίας του. Υπάρχει ένας αποθηκευτικός χώρος state όπου αποθηκεύονται όλα τα ενδιάμεσα αποτελέσματα. Κάθε γύρος έχει τρία επίπεδα:

1. **Σύγχυσης**: επιτυγχάνεται με την διαδικασία μετάθεσης (substitute) sub_bytes μέσω των S-Boxes
2. **Διάχυσης**: επιτυγχάνεται με τις διαδικασίες ολίσθησης shift_rows και μίξης mix_columns
3. **Κρυπτογράφησης**: επιτυγχάνεται με την διαδικασία add_roundkey που γίνεται XOR με το κλειδί του γύρου

(βλ. Ντούσκας, 2017: 33)

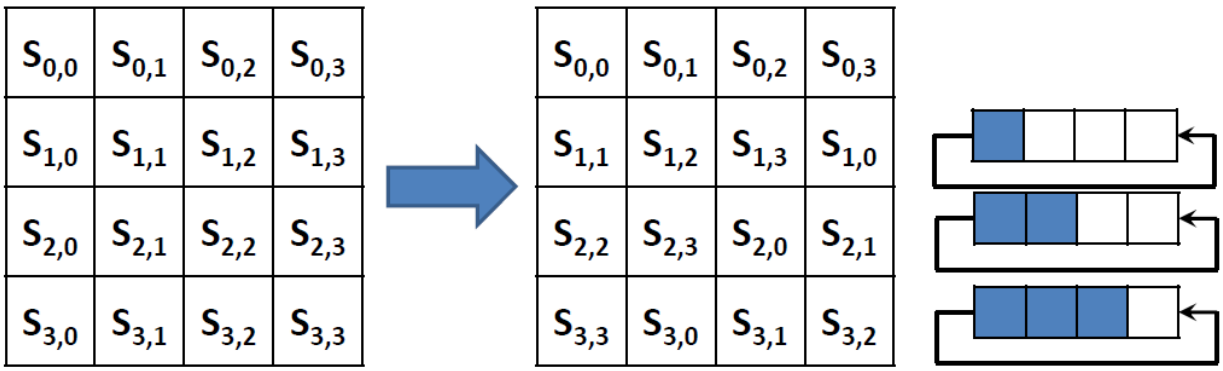


Συγκεκριμένα:

- **add_roundkey**: κάνει XOR (⊕) τον πίνακα state με το κλειδί που παράγεται από την διαδικασία key_schedule
- **sub_bytes**: κάθε byte του πίνακα state αντικαθίσταται από ένα άλλο byte μέσω ενός S-box (ο πρώτος αριθμός της εισόδου προσδιορίζει την γραμμή στον πίνακα του S-Box και ο δεύτερος την στήλη). Για κάθε byte έχουμε το ίδιο S-box. Ο πίνακας που δείχνει την αντικατάσταση των bytes μέσω του S-Box είναι:

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

- **shift_rows**: γίνεται αριστερή ολίσθηση στις γραμμές του πίνακα state κατά 0, 1, 2 και 3 θέσεις αντίστοιχα.



- **mix_columns**: σε αυτήν την διαδικασία πολλαπλασιάζεται ο πίνακας state με έναν σταθερό πίνακα που προκύπτει από πράξεις μεταξύ πολυωνύμων ο οποίος είναι:

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix}$$

- **key_schedule**: είναι η διαδικασία που παράγει το κλειδί του κάθε γύρου με βάση το αρχικό κλειδί. Αναλόγως με το μέγεθος του κλειδιού χωρίζουμε το κλειδί του κάθε γύρου σε λέξεις (words) (1 λέξη = 4 bytes).

  Για κλειδί:

  - 128bits χωρίζουμε σε 4 λέξεις
  - 192bits χωρίζουμε σε 6 λέξεις
  - και για 256bits χωρίζουμε σε 8 λέξεις

  Τώρα για τον υπολογισμό των λέξεων στον AES-128bits χρησιμοποιούμε τον αλγόριθμο:

```
(w0,...,w3) = key
for i = 4 to 43 {
    temp = wi-1
    if (i = 0 mod 4)
        then temp = sub_word(rot_word(temp))⊕ RCi/4
    wi = wi-4 ⊕ temp
}
```

Για τον AES-192bits και AES-256bits ο αλγόριθμος είναι πιο πολύπλοκος. Η συνάρτηση sub_word στον παραπάνω αλγόριθμο αντικαθιστά κάθε byte μέσω ενός S-Box και η συνάρτηση rot_word κάνει αριστερή ολίσθηση κατά 1 byte. Όσο για την μεταβλητή RC έχει προκαθορισμένες τιμές.

Παράδειγμα

Στο παράδειγμα θα χρησιμοποιηθεί ο AES-128bits άρα θα έχουμε 10 γύρους (rounds).

Έχουμε το κλειδί Thats my Kung Fu (16 ASCII χαρακτήρες από 1 byte (8 bit) ο καθένας), $16 * 8 = 128$ bit και το μήνυμα που θα κρυπτογραφηθεί είναι το Two One Nine Two.

Πριν αρχίσουμε τους γύρους κρυπτογράφησης, μετατρέπουμε το κλειδί και το μήνυμα σε δεκαεξαδική μορφή (hex).

Thats my Kung Fu = $(54\ 68\ 61\ 74\ 73\ 20\ 6D\ 79\ 20\ 4B\ 75\ 6E\ 67\ 20\ 46\ 75)_{16}$

Two One Nine Two = $(54\ 77\ 6F\ 20\ 4F\ 6E\ 65\ 20\ 4E\ 69\ 6E\ 65\ 20\ 54\ 77\ 6F)_{16}$

Τώρα θα βρούμε όλα τα round keys. Για 10 γύρους χρειαζόμαστε 11 υπό-κλειδιά των 128bits.

- Για κάθε κλειδί θα έχουμε 4 λέξεις
- Συνολικά πρέπει να παράγουμε 44 λέξεις ($44 * 32 = 1408$ bits) σε ένα πίνακα w_0,..., w_{43}
- Οι λέξεις w χρησιμοποιούνται ανά τέσσερις σε κάθε διαδικασία δημιουργίας round key.
- Αρχικά οι τέσσερις πρώτες λέξεις w_0 έως w_3 φορτώνονται με τα 128 bits του αρχικού κλειδιού στην δεκαεξαδική του μορφή.

$$w[0] = (54,68,61,74), w[1] = (73,20,6D, 79), w[2] = (20,4B, 75,6E), w[3] = (67,20,46,75)$$

Άρα το round key 0 είναι: 54 68 61 74 73 20 6D 79 20 4B 75 6E 67 20 46 75

Στη συνέχεια, ακολουθεί διαδικασία που επαναλαμβάνεται 40 φορές (4 λέξεις για τους υπόλοιπους 10 γύρους) για να βρούμε και τις λέξεις w_4 έως w_{43}. Για τον υπολογισμό τους χρησιμοποιούμε τον αλγόριθμο:

```
for i = 4 to 43 {
    temp = w_{i-1}
    if (i = 0 mod 4)
        then temp = sub_word(rot_word(temp)) ⊕ RC_{i/4}
    w_i = w_{i-4} ⊕ temp
}
```

Για την μεταβλητή RC έχουμε:

RC_1 = 01000000

RC_2 = 02000000

RC_3 = 04000000

RC_4 = 08000000

RC_5 = 10000000

RC_6 = 20000000

RC_7 = 40000000

RC_8 = 80000000

RC_9 = 1B000000

RC_{10} = 36000000

Έτσι έχουμε $w[4] = w[0] \oplus g(w[3])$ όπου g είναι οι διαδικασίες ολίσθησης, μετάθεσης και πρόσθεσης του round constant (RC).

Έτσι $g(w[3])$ κάνει:

- Αριστερή ολίσθηση κατά 1: $(20,46,75,67)$
- Μετάθεση (S-Box): $(B7,5A,9D,85)$
- Πρόσθεση του RC_1 (01000000): $(B6,5A,9D,85)$

Άρα τελικά $g(w[3]) = (B6,5A,9D,85)$ και $w[4] = w[0] \oplus g(w[3]) = (E2,32,FC,F1)$

και

$w[5] = w[4] \oplus w[1] = (91,12,91,88)$

$w[6] = w[5] \oplus w[2] = (B1,59,E4,E6)$

$w[7] = w[6] \oplus w[3] = (D6,79,A2,93)$

Άρα το round key 1 είναι: E2 32 FC F1 91 12 91 88 B1 59 E4 E6 D6 79 A2 93

Εφαρμόζοντας τον ίδιο αλγόριθμο και για τα υπόλοιπα προκύπτουν τα παρακάτω round keys:

Round key 0:	54 68 61 74 73 20 6D 79 20 4B 75 6E 67 20 46 75
Round key 1:	E2 32 FC F1 91 12 91 88 B1 59 E4 E6 D6 79 A2 93
Round key 2:	56 08 20 07 C7 1A B1 8F 76 43 55 69 A0 3A F7 FA
Round key 3:	D2 60 0D E7 15 7A BC 68 63 39 E9 01 C3 03 1E FB
Round key 4:	A1 12 02 C9 B4 68 BE A1 D7 51 57 A0 14 52 49 5B
Round key 5:	B1 29 3B 33 05 41 85 92 D2 10 D2 32 C6 42 9B 69
Round key 6:	BD 3D C2 B7 B8 7C 47 15 6A 6C 95 27 AC 2E 0E 4E
Round key 7:	CC 96 ED 16 74 EA AA 03 1E 86 3F 24 B2 A8 31 6A
Round key 8:	8E 51 EF 21 FA BB 45 22 E4 3D 7A 06 56 95 4B 6C
Round key 9:	BF E2 BF 90 45 59 FA B2 A1 64 80 B4 F7 F1 CB D8
Round key 10:	28 FD DE F8 6D A4 24 4A CC C0 A4 FE 3B 31 6F 26

Round 0

Κάνουμε XOR ($\oplus$) τον πίνακα state με το round key 0:

$$\begin{pmatrix} 54 & 4F & 4E & 20 \\ 77 & 6E & 69 & 54 \\ 6F & 65 & 6E & 77 \\ 20 & 20 & 65 & 6F \end{pmatrix} \oplus \begin{pmatrix} 54 & 73 & 20 & 67 \\ 68 & 20 & 4B & 20 \\ 61 & 6D & 75 & 46 \\ 74 & 79 & 6E & 75 \end{pmatrix}$$

Και έχουμε το αποτέλεσμα όπου γίνεται και ο νέος state πίνακας:

$$\begin{pmatrix} 00 & 3C & 6E & 47 \\ 1F & 4E & 22 & 74 \\ 0E & 08 & 1B & 31 \\ 54 & 59 & 0B & 1A \end{pmatrix}$$

Άρα το αποτέλεσμα του AES στον γύρο μηδέν είναι:

00 1F 0E 54 3C 4E 08 59 6E 22 1B 0B 47 74 31 1A

Round 1

Τώρα κάνουμε μετάθεση (sub_bytes) κάθε byte του πίνακα state με το byte που ορίζει το S-Box και έχουμε τον νέο πίνακα state:

$$\begin{pmatrix} 63 & EB & 9F & A0 \\ C0 & 2F & 93 & 92 \\ AB & 30 & AF & C7 \\ 20 & CB & 2B & A2 \end{pmatrix}$$

Στην συνέχεια, κάνουμε αλλαγή (shift_rows) και των τεσσάρων γραμμών του πίνακα state και προκύπτει ο νέος πίνακας state:

$$\begin{pmatrix} 63 & EB & 9F & A0 \\ 2F & 93 & 92 & C0 \\ AF & C7 & AB & 30 \\ A2 & 20 & CB & 2B \end{pmatrix}$$

Συνεχίζουμε κάνοντας XOR τον σταθερό πίνακα αυτής της διαδικασίας (mix_columns) με τον πίνακα state:

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \oplus \begin{pmatrix} 63 & EB & 9F & A0 \\ 2F & 93 & 92 & C0 \\ AF & C7 & AB & 30 \\ A2 & 20 & CB & 2B \end{pmatrix}$$

Και ως αποτέλεσμα έχουμε πάλι τον νέο πίνακα state:

$$\begin{pmatrix} BA & 84 & E8 & 1B \\ 75 & A4 & 8D & 40 \\ F4 & 8D & 06 & 7D \\ 7A & 32 & 0E & 5D \end{pmatrix}$$

Τελευταίο βήμα του γύρου 1 είναι η διαδικασία (add_roundkey) XOR του πίνακα state με το κλειδί του συγκεκριμένου γύρου:

$$\begin{pmatrix} BA & 84 & E8 & 1B \\ 75 & A4 & 8D & 40 \\ F4 & 8D & 06 & 7D \\ 7A & 32 & 0E & 5D \end{pmatrix} \oplus \begin{pmatrix} E2 & 91 & B1 & D6 \\ 32 & 12 & 59 & 79 \\ FC & 91 & E4 & A2 \\ F1 & 88 & E6 & 93 \end{pmatrix}$$

Με αποτέλεσμα τον νέο πίνακα state που θα χρησιμοποιηθεί με ίδιο τρόπο και στους επόμενους γύρους:

$$\begin{pmatrix} 58 & 15 & 59 & CD \\ 47 & B6 & D4 & 39 \\ 08 & 1C & E2 & DF \\ 8B & BA & E8 & CE \end{pmatrix}$$

Άρα το αποτέλεσμα του AES στον πρώτο γύρο είναι:

58 47 08 8B 15 B6 1C BA 59 D4 E2 E8 CD 39 DF CE

Round 2

Μετά από sub_bytes και shift_rows:

$$\begin{pmatrix} 6A & 59 & CB & BD \\ A0 & 4E & 48 & 12 \\ 30 & 9C & 98 & 9E \\ 3D & F4 & 9B & 8B \end{pmatrix} \qquad \begin{pmatrix} 6A & 59 & CB & BD \\ E4 & 48 & 12 & A0 \\ 98 & 9E & 30 & 9B \\ 8B & 3D & F4 & 9B \end{pmatrix}$$

Μετά από mix_columns και add_roundkey:

$$\begin{pmatrix} 15 & C9 & 7F & 9D \\ CE & 4D & 4B & C2 \\ 89 & 71 & BE & 88 \\ 65 & 47 & 97 & CD \end{pmatrix} \qquad \begin{pmatrix} 43 & 0E & 09 & 3D \\ C6 & 57 & 08 & F8 \\ A9 & C0 & EB & 7F \\ 62 & C8 & FE & 37 \end{pmatrix}$$

Άρα το αποτέλεσμα του AES στον δεύτερο γύρο είναι:

43 C9 A9 62 0E 57 C0 C8 09 08 EB FE 3D F8 7F 37

Round 3

Μετά από sub_bytes και shift_rows:

$$\begin{pmatrix} 1A & AB & 01 & 27 \\ B4 & 5B & 30 & 41 \\ D3 & BA & E9 & D2 \\ AA & E8 & BB & 9A \end{pmatrix} \qquad \begin{pmatrix} 1A & AB & 01 & 27 \\ 5B & 30 & 41 & B4 \\ E9 & D2 & D3 & BA \\ A9 & AA & E8 & BB \end{pmatrix}$$

Μετά από mix_columns και add_roundkey:

$$\begin{pmatrix} AA & 65 & FA & 88 \\ 16 & 0C & 05 & 3A \\ 3D & C1 & DE & 2A \\ B3 & 4B & 5A & 0A \end{pmatrix} \qquad \begin{pmatrix} 78 & 70 & 99 & 4B \\ 76 & 76 & 3C & 39 \\ 30 & 7D & 37 & 43 \\ 54 & 23 & 5B & F1 \end{pmatrix}$$

Άρα το αποτέλεσμα του AES στον τρίτο γύρο είναι:

78 76 30 54 70 76 7D 23 99 3C 37 5B 4B 39 43 F1

Round 4

Μετά από sub_bytes και shift_rows:

$$\begin{pmatrix} BC & 51 & EE & B3 \\ 38 & 38 & EB & 12 \\ 04 & FF & 9A & 18 \\ 20 & 26 & 39 & A1 \end{pmatrix} \qquad \begin{pmatrix} BC & 51 & EE & B3 \\ 38 & EB & 12 & 38 \\ 9A & 18 & 04 & FF \\ A1 & 20 & 26 & 39 \end{pmatrix}$$

Μετά από mix_columns και add_roundkey:

$$\begin{pmatrix} 10 & BC & D3 & F3 \\ D8 & 94 & E0 & E0 \\ 53 & EA & 9E & 25 \\ 24 & 40 & 73 & 7B \end{pmatrix} \qquad \begin{pmatrix} B1 & 08 & 04 & E7 \\ CA & DC & B1 & B2 \\ 51 & 54 & C9 & 6C \\ ED & E1 & D3 & 20 \end{pmatrix}$$

Άρα το αποτέλεσμα του AES στον τέταρτο γύρο είναι:

B1 CA 51 ED 08 DC 54 E1 04 B1 C9 D3 E7 B2 6C 20

Round 5

Μετά από sub_bytes και shift_rows:

$$\begin{pmatrix} C8 & 30 & F2 & 94 \\ 74 & B0 & C8 & 37 \\ D1 & 20 & DD & 50 \\ 55 & F8 & 66 & B7 \end{pmatrix} \qquad \begin{pmatrix} C8 & 30 & F2 & 94 \\ B0 & C8 & 37 & 74 \\ DD & 50 & D1 & 20 \\ B7 & 55 & F8 & 66 \end{pmatrix}$$

Μετά από mix_columns και add_roundkey:

$$\begin{pmatrix} 2A & 26 & 8F & E9 \\ 78 & 1E & 0C & 7A \\ 1B & A7 & 6F & 0A \\ 5B & 62 & 00 & 3F \end{pmatrix} \qquad \begin{pmatrix} 9B & 23 & 5D & 2F \\ 51 & 5F & 1C & 38 \\ 20 & 22 & BD & 91 \\ 68 & F0 & 32 & 56 \end{pmatrix}$$

Άρα το αποτέλεσμα του AES στον πέμπτο γύρο είναι:

9B 51 20 68 23 5F 22 F0 5D 1C BD 32 2F 38 91 56

Round 6

Μετά από sub_bytes και shift_rows:

$$\begin{pmatrix} 14 & 26 & 4C & 15 \\ D1 & CF & 9C & 07 \\ B7 & 93 & 7A & 81 \\ 45 & 8C & 23 & B1 \end{pmatrix} \qquad \begin{pmatrix} 14 & 26 & 4C & 15 \\ CF & 9C & 07 & D1 \\ 7A & 81 & B7 & 93 \\ B1 & 45 & 8C & 23 \end{pmatrix}$$

Μετά από mix_columns και add_roundkey:

$$\begin{pmatrix} A9 & 37 & AA & F2 \\ AE & D8 & 0C & 21 \\ E7 & 6C & B1 & 9C \\ F0 & FD & 67 & 3B \end{pmatrix} \qquad \begin{pmatrix} 14 & 8F & C0 & 5E \\ 93 & A4 & 60 & 0F \\ 25 & 2B & 24 & 92 \\ 77 & E8 & 40 & 75 \end{pmatrix}$$

Άρα το αποτέλεσμα του AES στον έκτο γύρο είναι:

14 93 25 77 8F A4 2B E8 C0 60 24 40 5E 0F 92 75

Round 7

Μετά από sub_bytes και shift_rows:

$$\begin{pmatrix} FA & 73 & BA & 58 \\ DC & 49 & D0 & 76 \\ 3F & F1 & 36 & 4F \\ F5 & 9B & 09 & 9D \end{pmatrix} \qquad \begin{pmatrix} FA & 73 & BA & 58 \\ 49 & D0 & 76 & DC \\ 36 & 4F & 3F & F1 \\ 9D & F5 & 9B & 09 \end{pmatrix}$$

Μετά από mix_columns και add_roundkey:

$$\begin{pmatrix} 9F & 37 & 51 & 37 \\ AF & EC & 8C & FA \\ 63 & 39 & 04 & 66 \\ 4B & DB & B1 & D7 \end{pmatrix} \qquad \begin{pmatrix} 53 & 43 & 4F & 85 \\ 39 & 06 & 0A & 52 \\ 8E & 93 & 3B & 57 \\ 5D & F8 & 95 & BD \end{pmatrix}$$

Άρα το αποτέλεσμα του AES στον έβδομο γύρο είναι:

53 39 8E 5D 43 06 93 F8 4F 0A 3B 95 85 52 57 BD

Round 8

Μετά από sub_bytes και shift_rows:

$$\begin{pmatrix} ED & 1A & 84 & 97 \\ 12 & 6F & 67 & 00 \\ 19 & DC & E2 & 5B \\ 4C & 41 & 2A & 7A \end{pmatrix} \qquad \begin{pmatrix} ED & 1A & 84 & 97 \\ 6F & 67 & 00 & 12 \\ E2 & 5B & 19 & DC \\ 7A & 4C & 41 & 2A \end{pmatrix}$$

Μετά από mix_columns και add_roundkey:

$$\begin{pmatrix} E8 & 8A & 4B & F5 \\ 74 & 75 & EE & E6 \\ D3 & 1F & 75 & 58 \\ 55 & 8A & 0C & 38 \end{pmatrix} \qquad \begin{pmatrix} 66 & 70 & AF & A3 \\ 25 & CE & D3 & 73 \\ 3C & 5A & 0F & 13 \\ 74 & A8 & 0A & 54 \end{pmatrix}$$

Άρα το αποτέλεσμα του AES στον όγδοο γύρο είναι:

66 25 3C 74 70 CE 5A A8 AF D3 0F 0A A3 73 13 54

Round 9

Μετά από sub_bytes και shift_rows:

$$\begin{pmatrix} 33 & 51 & 79 & 0A \\ 3F & 8B & 66 & 8F \\ EB & BE & 76 & 7D \\ 92 & C2 & 67 & 20 \end{pmatrix} \qquad \begin{pmatrix} 33 & 51 & 79 & 0A \\ 8B & 66 & 8F & 3F \\ 76 & 7D & EB & BE \\ 20 & 92 & C2 & 67 \end{pmatrix}$$

Μετά από mix_columns και add_roundkey:

$$\begin{pmatrix} B6 & E7 & 51 & 8C \\ 84 & 88 & 98 & CA \\ 34 & 60 & 66 & FB \\ E8 & D7 & 70 & 51 \end{pmatrix} \qquad \begin{pmatrix} 09 & A2 & F0 & 7B \\ 66 & D1 & FC & 3B \\ 8B & 9A & E6 & 30 \\ 78 & 65 & C4 & 89 \end{pmatrix}$$

Άρα το αποτέλεσμα του AES στον ένατο γύρο είναι:

09 66 8B 78 A2 D1 9A 65 F0 FC E6 C4 7B 3B 30 89

Round 10

Μετά από sub_bytes και shift_rows:

$$\begin{pmatrix} 01 & 3A & 8C & 21 \\ 33 & 3E & B0 & E2 \\ 3D & B8 & 8E & 04 \\ BC & 4D & 1C & A7 \end{pmatrix} \qquad \begin{pmatrix} 01 & 3A & 8C & 21 \\ 3E & B0 & E2 & 33 \\ 8E & 04 & 3D & B8 \\ A7 & BC & 4D & 1C \end{pmatrix}$$

Μετά από add_roundkey (στον τελευταίο γύρο δεν κάνουμε mix_columns):

$$\begin{pmatrix} 29 & 57 & 40 & 1A \\ C3 & 14 & 22 & 02 \\ 50 & 20 & 99 & D7 \\ 5F & F6 & B3 & 3A \end{pmatrix}$$

Άρα το αποτέλεσμα του AES στον δέκατο γύρο είναι:

29 C3 50 5F 57 14 20 F6 40 22 99 B3 1A 02 D7 3A

Οπότε το μήνυμα Two One Nine Two με κλειδί Thats my Kung Fu και αλγόριθμο AES-128bits κρυπτογραφείτε σε 29 C3 50 5F 57 14 20 F6 40 22 99 B3 1A 02 D7 3A.

(Πηγή παραδείγματος: https://kavaliro.com/wp-content/uploads/2014/03/AES.pdf)

3.2 Ασύμμετρη Κρυπτογράφηση

Κύρια διαφορά με την συμμετρική κρυπτογράφηση είναι ότι εδώ έχουμε δύο κλειδιά, όπου τα δύο διαφορετικά κλειδιά που κατέχουν οι χρήστες έχουν διαφορετικές χρήσεις, το ένα χρησιμοποιείται για την κρυπτογράφηση και το άλλο για την αποκρυπτογράφηση. Το κλειδί κρυπτογράφησης είναι το δημόσιο κλειδί (public key), το οποίο είναι διαθέσιμο σε όλους τους χρήστες, ενώ το κλειδί αποκρυπτογράφησης είναι το ιδιωτικό κλειδί (private key), το οποίο πρέπει να φυλάσσεται μυστικό. (Rivest, Shamir & Adleman, 1978)

3.2.1 Rivest Shamir Adelman (RSA)

Ο RSA χρησιμοποιεί μαθηματικούς τρόπους για την παραγωγή κλειδιών, αλλά και για την κρυπτογράφηση-αποκρυπτογράφηση του μηνύματος. Έτσι, από την μεριά του αντιπάλου χρειάζεται τεράστιο χρονικό διάστημα για να μαντέψει το κλειδί αποκρυπτογράφησης (ιδιωτικό κλειδί), λόγω της μαθηματικής πολυπλοκότητας (παραγοντοποίηση ενός μεγάλου αριθμού σε δύο πρώτους). (http://doctrina.org/How-RSA-Works-With-Examples.html)

Ο RSA για την δημιουργία του ζεύγους κλειδιών (δημόσιο και ιδιωτικό κλειδί) και την κρυπτογράφηση-αποκρυπτογράφηση εκτελεί τα παρακάτω βήματα:

1. Επιλέγονται δύο μεγάλοι πρώτοι αριθμοί (p, q)
2. Υπολογίζεται το $n = p * q$
3. Στη συνέχεια υπολογίζεται το $\varphi = (p - 1) * (q - 1)$
4. Επιλέγεται ακέραιος e που να ανήκει στο διάστημα $1 < e < \varphi$ και $gcd(e, \varphi) = 1$
5. Υπολογίζεται d έτσι ώστε $e * d = 1 \; mod \; \varphi$
6. Τέλος δημιουργείται το δημόσιο και το ιδιωτικό κλειδί όπου public key $= \{n, e\}$ και private key $= \{n, d\}$

7. Τώρα για την κρυπτογράφηση των μηνυμάτων χρησιμοποιείται η συνάρτηση $E(M)$ όπου M το μήνυμα και η συνάρτηση E ορίζεται ως εξής: $E(M) = M^e \ mod \ n$. Όσο για την αποκρυπτογράφηση των μηνυμάτων χρησιμοποιείται η συνάρτηση $D\big(E(M)\big)$ όπου $E(M)$ το κρυπτογραφημένο μήνυμα και η συνάρτηση D ορίζεται ως εξής: $D\big(E(M)\big) = E(M)^d \ mod \ n$

(βλ. Ντούσκας, 2017: 5)

Παράδειγμα

1. Έστω $p = 7$ και $q = 13$ είναι οι δύο πρώτοι αριθμοί (δεν είναι μεγάλοι για ευκολία στις πράξεις και την ευκολότερη κατανόηση του αλγορίθμου)
2. $n = p * q = 7 * 13 = 91$
3. $\varphi = (p-1) * (q-1) = 6 * 13 = 72$
4. Τώρα πρέπει να διαλέξουμε e τέτοιο ώστε $gcd(e, \varphi) = 1$ και $1 < e < \varphi$

   Έστω $e = 5$ βλέπουμε ότι $gcd(5{,}72) = 1$ και $1 < 5 < 72$, άρα διαλέγουμε $e = 5$
5. $e * d \equiv 1 \ mod \ \varphi$

   Μπορούμε να χρησιμοποιήσουμε τον αλγόριθμο του Ευκλείδη και τον αντίστροφό του για να βρούμε το d

   $$5 * d \equiv 1 \ mod \ 72$$

   Αλγόριθμος του Ευκλείδη:

   $72 = 5 * 14 + 2$

   $5 = 2 * 2 + 1$

   Αντίστροφος αλγόριθμος του Ευκλείδη:

   $1 = 5 - 2 * 2$

   $1 = 5 - 2 * (72 - 14 * 5)$

   $1 = 5 - 2 * 72 + 28 * 5$

   $1 = \mathbf{29} * 5 - 2 * 72$

   Άρα έχουμε ότι $d = 29 \ mod \ 72 = 29$

6. Οπότε το public key θα είναι $\{91,5\}$ και το private key θα είναι $\{91, 29\}$
7. Αν θέλουμε να κρυπτογραφήσουμε το μήνυμα $M = 18$ θα έχουμε:

   $E(M) = E(18) = 18^5 \ mod \ 91 = 44$

   $D\big(E(M)\big) = D(44) = 44^{29} \ mod \ 91 = 18$

(Πηγή παραδείγματος: https://asecuritysite.com/Encryption/rsa)

4. ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΑΣΦΑΛΟΥΣ ΕΠΙΚΟΙΝΩΝΙΑΣ

4.1 Εισαγωγή

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε η γλώσσα προγραμματισμού Java. Η γλώσσα προγραμματισμού Java αναπτύχθηκε αρχικά από την Sun Microsystems, από τον James Gosling και κυκλοφόρησε το 1995 ως το βασικό συστατικό της Java's Sun Microsystems (Java 1.0 [J2SE]). Η τελευταία έκδοση της Java Standard Edition είναι η Java SE 8 και είναι διαθέσιμη για όλους. Με την πρόοδο της Java και την ευρεία δημοτικότητά της, έχουν δημιουργηθεί πολλές διαμορφώσεις, για να ταιριάζουν σε διαφορετικούς τύπους πλατφορμών. Η γλώσσα προγραμματισμού Java είναι:

- Αντικειμενοστραφής
- Ανεξάρτητη πλατφόρμας
- Εύκολη στην κατανόηση
- Ασφαλής
- Πολυνηματική
- Υψηλής απόδοσης
- Δυναμική

Οι νέες εκδόσεις J2 μετονομάστηκαν ως Java SE, Java EE και Java ME αντίστοιχα. Έτσι η Java είναι μία γλώσσα προγραμματισμού, η οποία γράφεται μία φορά και μπορεί να τρέχει σε οποιαδήποτε πλατφόρμα. (https://www.java.com)

Το περιβάλλον το οποίο χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής είναι το NetBeans Integrated Development Environment (IDE) 8.2. Το NetBeans IDE είναι το επίσημο IDE για την Java 8 και παρέχει στους προγραμματιστές Java όλα τα εργαλεία που απαιτούνται για τη δημιουργία εφαρμογών. (https://netbeans.org)

Επιπλέον, χρησιμοποιήθηκαν δύο JAR (Java ARchive) αρχεία. Το commons-codec-1.10.jar για την μετατροπή των πινάκων από byte σε τύπο String και το javax.mail-1.5.6.jar για την αποστολή email. Όλα τα υπόλοιπα παρέχονταν από έτοιμες βιβλιοθήκες και από το ίδιο το NetBeans.

4.2 Περιγραφή Εφαρμογής

Πριν ξεκινήσει η περιγραφή (εγχειρίδιο χρήσης) της εφαρμογής, θα γίνει μία περιληπτική αναφορά στον τρόπο με τον οποίο επιτυγχάνεται η κρυπτογράφηση με την χρήση και των δύο αλγορίθμων στην εφαρμογή.

Αρχικά για την κρυπτογράφηση με τον αλγόριθμο AES έχουμε:

```
import javax.crypto.Cipher;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;

import org.apache.commons.codec.binary.Base64;
```

Όπου το `import javax.crypto.Cipher` είναι για να μπορέσουμε να χρησιμοποιήσουμε έναν από τους πολλούς αλγόριθμους κρυπτογράφησης που παρέχει η συγκεκριμένη βιβλιοθήκη. Στη περίπτωσή μας, τον αλγόριθμο AES.

Το `import javax.crypto.spec.IvParameterSpec` και το `import javax.crypto.spec.SecretKeySpec` μας δίνουν την δυνατότητα να δημιουργήσουμε το αρχικό διάνυσμα (initVector) και το μυστικό κλειδί (key), τα οποία θα χρειαστούμε για να πραγματοποιήσουμε την κρυπτογράφηση και αποκρυπτογράφηση με τον αλγόριθμο AES.

Όσο για το `import org.apache.commons.codec.binary.Base64` είναι για να χρησιμοποιήσουμε μία από τις συναρτήσεις που μας παρέχει το αρχείο JAR (commons-codec-1.10.jar), για να μετατρέψουμε τους πίνακες από byte σε τύπο String.

Αφού τελειώσαμε με όλα τα απαραίτητα imports συνεχίζουμε με την συνάρτηση της κρυπτογράφησης:

```
public String encrypt(String key, String initVector, String text) {
    try {
        IvParameterSpec iv = new IvParameterSpec(initVector.getBytes("UTF-8"));
        SecretKeySpec skeySpec = new SecretKeySpec(key.getBytes("UTF-8"), "AES");

        Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");
        cipher.init(Cipher.ENCRYPT_MODE, skeySpec, iv);
```

```
            byte[] encrypted = cipher.doFinal(text.getBytes());

            return Base64.encodeBase64String(encrypted);
        } catch (Exception ex) {
            ex.printStackTrace();
        }

        return null;
    }
```

Η συνάρτηση `public String encrypt` παίρνει τρία ορίσματα τύπου String: το μυστικό κλειδί (key), το αρχικό διάνυσμα (initVector) και το μήνυμα (text) που θέλουμε να κρυπτογραφήσουμε. Σε αυτό το σημείο, αξίζει να αναφέρουμε πως το initVector χρησιμοποιείται σαν επιπλέον ασφάλεια, αφού μας προσθέτει τυχαιότητα στην έναρξη της διαδικασίας κρυπτογράφησης.

Με την εντολή `IvParameterSpec iv = new IvParameterSpec (initVector.getBytes("UTF-8"))` δημιουργούμε ένα αρχικό διάνυσμα (iv) με βάση τα bytes του δεύτερου ορίσματος της συνάρτησης (`String initVector`) και με κωδικοποίηση UTF-8. Παρόμοια, για την εντολή `SecretKeySpec skeySpec = new SecretKeySpec (key.getBytes("UTF-8"), "AES")` δημιουργούμε ένα μυστικό κλειδί (skeySpec) με βάση τα bytes του πρώτου ορίσματος της συνάρτησης (`String key`) με την ίδια κωδικοποίηση και για αυτό. Να σημειώσουμε πως η κωδικοποίηση UTF-8 χρησιμοποιεί 8bits για κάθε χαρακτήρα.

Στην συνέχεια με την εντολή `Cipher cipher = Cipher.getInstance ("AES/CBC/PKCS5PADDING")` δημιουργούμε το αντικείμενο με το οποίο θα γίνει η κρυπτογράφηση και του ορίζουμε με ποιον αλγόριθμο θα γίνει. Με την εντολή `cipher.init(Cipher.ENCRYPT_MODE, skeySpec, iv)` που ακολουθεί, γίνεται η αρχικοποίησή του και του περνάμε τρία ορίσματα: το πρώτο καθορίζει αν είναι κρυπτογράφηση ή αποκρυπτογράφηση, το δεύτερο το μυστικό κλειδί και το τρίτο το αρχικό διάνυσμα.

Αφού δημιουργήθηκαν όλα τα απαραίτητα αντικείμενα, δημιουργούμε έναν πίνακα από byte και του αναθέτουμε τις κρυπτογραφημένες τιμές, καθενός byte του μηνύματος. Αυτό γίνεται με την εντολή `byte[] encrypted = cipher.doFinal(text.getBytes())`. Τέλος

επιστρέφουμε στο κρυπτογραφημένο μήνυμα, όμως, επειδή το χρειαζόμαστε σε μορφή String για την εγγραφή του σε αρχείο ή την αποστολή του με email, το μετατρέπουμε από πίνακα byte σε String με την εντολή `return Base64.encodeBase64String(encrypted)`.

Για την αποκρυπτογράφηση με τον αλγόριθμο AES έχουμε:

```
public String decrypt(String key, String initVector, String text) {
    try {
        IvParameterSpec iv = new IvParameterSpec(initVector.getBytes("UTF-8"));
        SecretKeySpec skeySpec = new SecretKeySpec(key.getBytes("UTF-8"), "AES");

        Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");
        cipher.init(Cipher.DECRYPT_MODE, skeySpec, iv);

        byte[] original = cipher.doFinal(Base64.decodeBase64(text));

        return new String(original);
    } catch (Exception ex) {
        ex.printStackTrace();
    }

    return null;
}
```

Η συνάρτηση `public String decrypt,` όπως βλέπουμε, είναι πολύ όμοια με την encrypt. Οι μόνες διαφορές είναι ότι το τρίτο όρισμα της συνάρτησης είναι το κρυπτογραφημένο μήνυμα (text) και στην εντολή `cipher.init(Cipher.DECRYPT_MODE, skeySpec, iv)`, στην οποία αλλάζει το πρώτο όρισμα και αυτή τη φορά δηλώνει μέθοδο αποκρυπτογράφησης. Αφού δημιουργήθηκαν όλα τα απαραίτητα αντικείμενα, δημιουργούμε έναν πίνακα από byte και του αναθέτουμε τις αποκρυπτογραφημένες τιμές του κρυπτογραφημένου μηνύματος. Αυτό γίνεται με την εντολή `byte[] original = cipher.doFinal (Base64.decodeBase64(text))`. Η εντολή `Base64.decodeBase64(text)` κάνει την αντίστροφη διαδικασία από την εντολή `Base64.encode Base64String(encrypted)` που χρησιμοποιήσαμε στην συνάρτηση encrypt. Τέλος, επιστρέφουμε το αποκρυπτογραφημένο μήνυμα, σε μορφή String με την εντολή `return new String(original)`.

Για την κρυπτογράφηση με τον αλγόριθμο RSA έχουμε:

```
import javax.crypto.Cipher;
import java.security.PrivateKey;
import java.security.PublicKey;

import org.apache.commons.codec.binary.Base64;
```

Το `import javax.crypto.Cipher` εξηγήθηκε με την περιγραφή του αλγόριθμου AES, όπως και το `import org.apache.commons.codec.binary.Base64`.

Το `import java.security.PrivateKey` και το `import java.security.PublicKey` μας δίνουν την δυνατότητα να δημιουργήσουμε το ζεύγος κλειδιών (δημόσιο και ιδιωτικό κλειδί) τα οποία θα χρειαστούμε για να πραγματοποιήσουμε την κρυπτογράφηση και αποκρυπτογράφηση με τον αλγόριθμο RSA.

Αφού τελειώσαμε με όλα τα νέα imports συνεχίζουμε με την συνάρτηση της κρυπτογράφησης:

```
public String encrypt(String text, PublicKey key) {
    try {
        final Cipher cipher = Cipher.getInstance(ALGORITHM);
        cipher.init(Cipher.ENCRYPT_MODE, key);

        byte[] cipherText = cipher.doFinal(text.getBytes());

        return Base64.encodeBase64String(cipherText);
    } catch (Exception ex) {
        ex.printStackTrace();
    }

    return null;
}
```

Η συνάρτηση `public String encrypt` παίρνει δύο ορίσματα: το μήνυμα (text) τύπου String που θέλουμε να κρυπτογραφήσουμε και το δημόσιο κλειδί (key) τύπου Public Key.

Η εντολή `final Cipher cipher = Cipher.getInstance(ALGORITHM)` κάνει την ίδια δουλειά, ακριβώς όπως είδαμε και πριν, στην κρυπτογράφηση με τον αλγόριθμο AES, μόνο που στην θέση της μεταβλητής ALGORITHM έχουμε την τιμή RSA (`public static final String ALGORITHM = "RSA"`) αφού θα χρησιμοποιήσουμε τον αλγόριθμο RSA. Με την

εντολή `cipher.init(Cipher.ENCRYPT_MODE, key)` γίνεται η αρχικοποίηση του αλγόριθμου και του περνάμε δύο ορίσματα: το πρώτο καθορίζει αν είναι κρυπτογράφηση ή αποκρυπτογράφηση και το δεύτερο το δημόσιο κλειδί. Όσο για τις εντολές `byte[] cipherText = cipher.doFinal(text.getBytes())` και `return Base64.encode Base64String(cipherText)` κάνουν ακριβώς την ίδια δουλειά όπως και στην κρυπτογράφηση με τον αλγόριθμο AES.

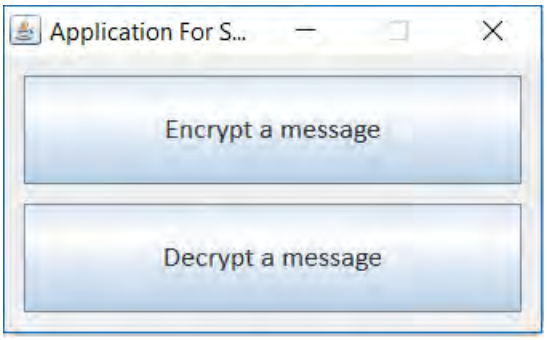
Για την αποκρυπτογράφηση με τον αλγόριθμο RSA έχουμε:

```
public String decrypt(String text, PrivateKey key) {
    try {
        final Cipher cipher = Cipher.getInstance(ALGORITHM);
        cipher.init(Cipher.DECRYPT_MODE, key);

        byte[] dectyptedText = cipher.doFinal(Base64.decodeBase64(text));

        return new String(dectyptedText);
    } catch (Exception ex) {
        ex.printStackTrace();
    }

    return null;
}
```

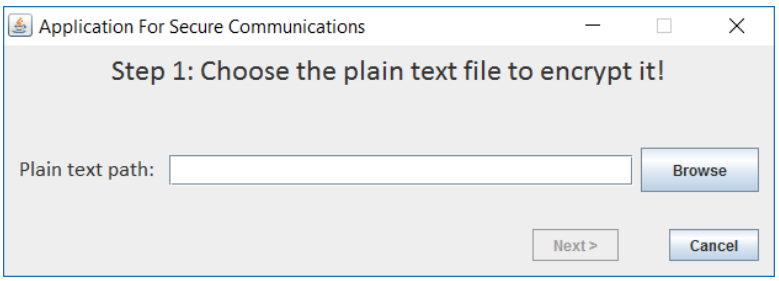
Η συνάρτηση `public String decrypt,` όπως βλέπουμε, είναι πολύ όμοια με την encrypt. Οι μόνες διαφορές είναι ότι το πρώτο όρισμα της συνάρτησης είναι το κρυπτογραφημένο μήνυμα (text) και ως δεύτερο όρισμα έχουμε το ιδιωτικό κλειδί (key) τύπου PrivateKey. Η υπόλοιπη διαδικασία είναι ίδια με αυτήν της αποκρυπτογράφησης με τον αλγόριθμο AES.

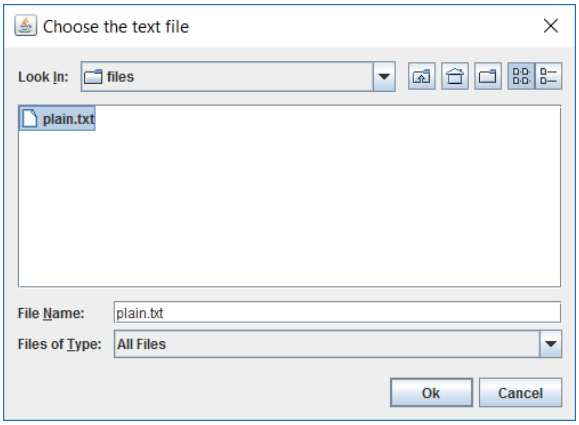
Ακολουθεί περιγραφή (εγχειρίδιο χρήσης) της εφαρμογής.

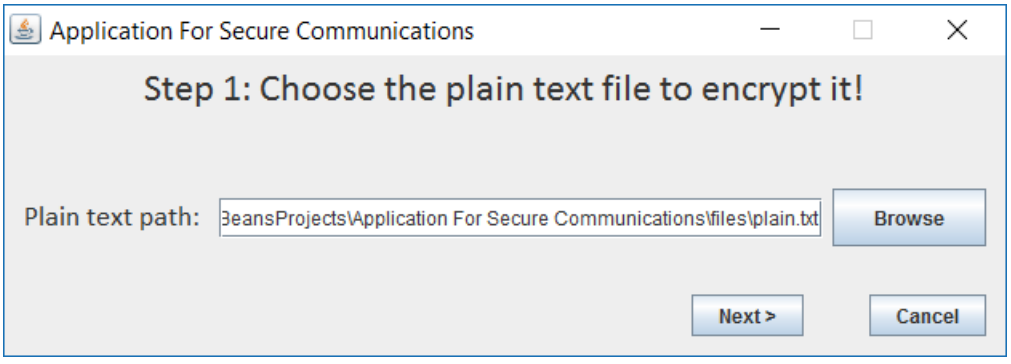
Με την εκκίνηση της εφαρμογής δίνεται στο χρήστη η δυνατότητα επιλογής κρυπτογράφησης ή αποκρυπτογράφησης αντίστοιχα.



Αρχικά θα περιγράψουμε την διαδικασία κρυπτογράφησης. Αφού ο χρήστης πατήσει το κουμπί Encrypt a message, εμφανίζεται ένα παράθυρο, στο οποίο πρέπει να συμπληρώσει την διαδρομή (path) του plaintext ή να αναζητήσει το αρχείο πατώντας το κουμπί Browse.

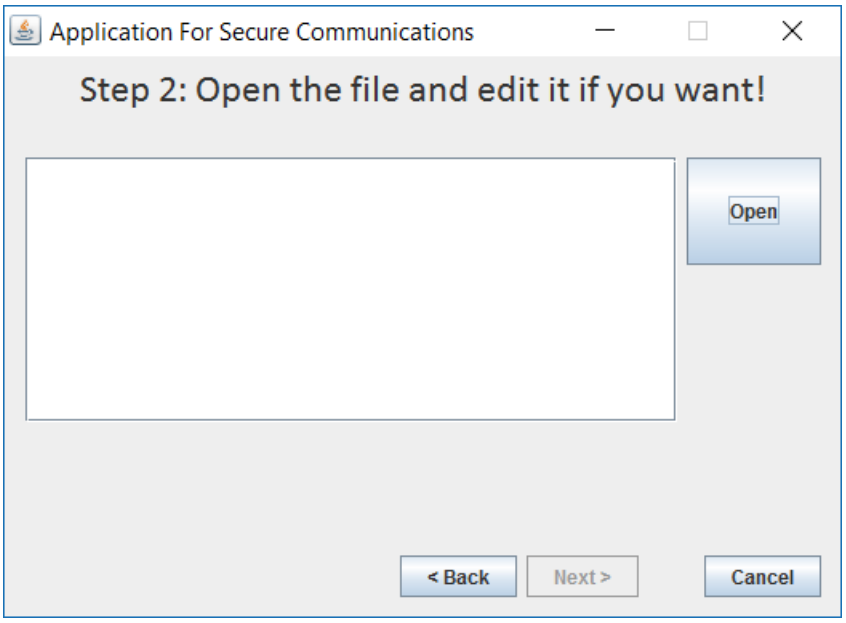




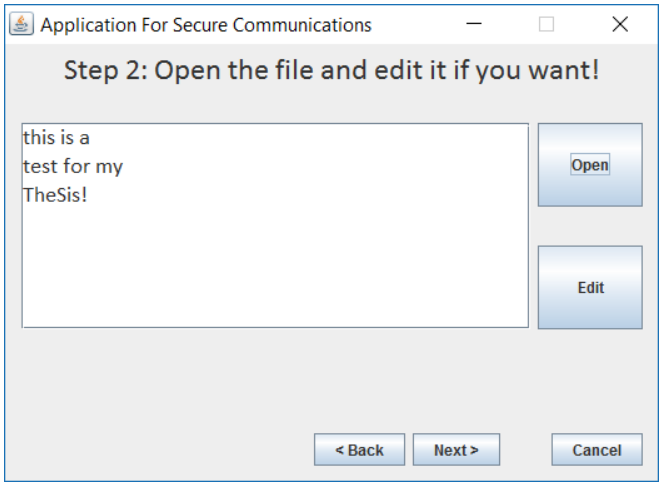


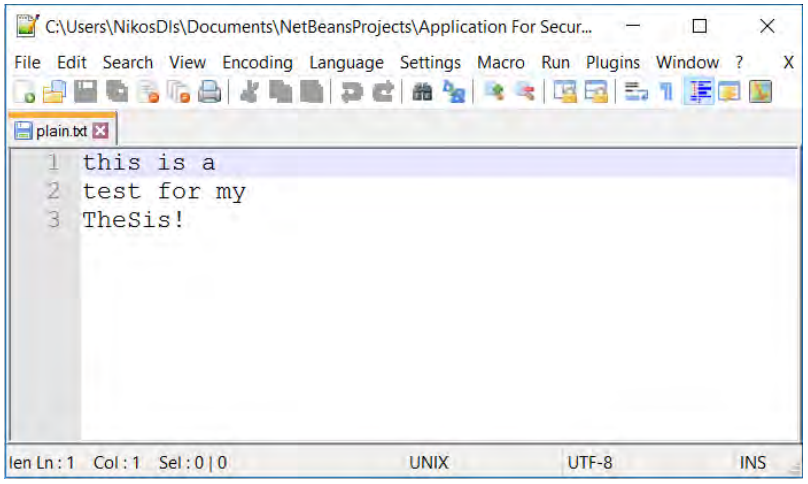
Αφού ολοκληρώσει την διαδικασία πατάει το κουμπί Next για να πάει στο επόμενο βήμα.

Το επόμενο παράθυρο που εμφανίζεται περιέχει ένα πλαίσιο στο οποίο θα εμφανιστεί το περιεχόμενο του txt αρχείου.

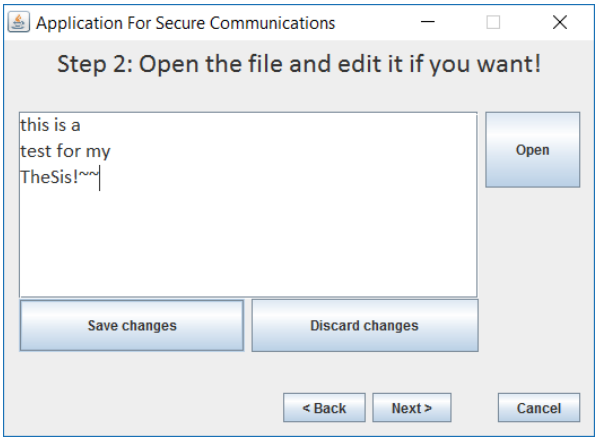


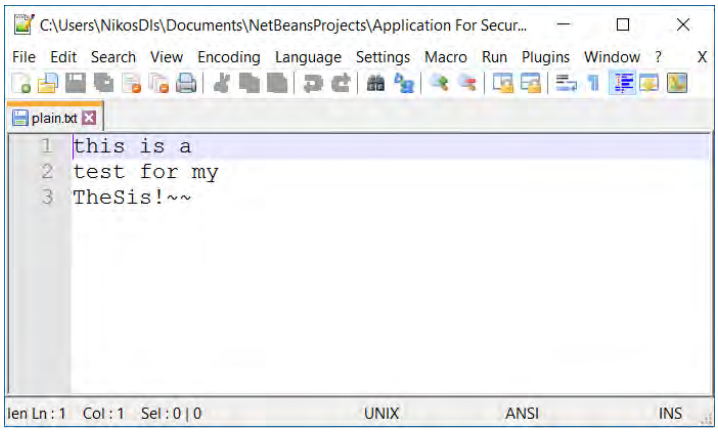
Όταν ο χρήσης πατήσει το κουμπί Open, εμφανίζεται στο πλαίσιο το περιεχόμενο του αρχείου και εμφανίζεται ένα νέο κουμπί Edit.





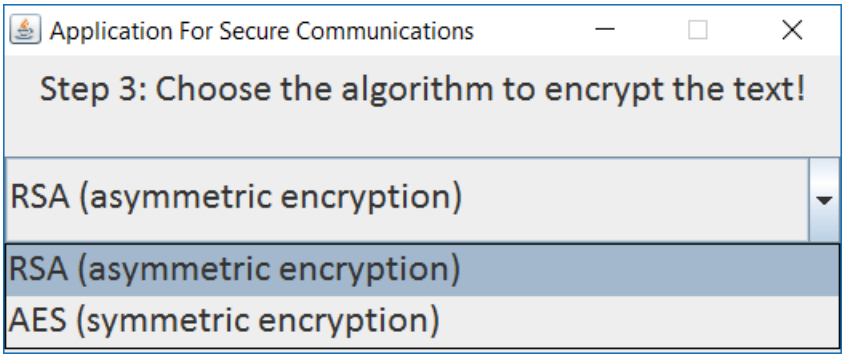
Αν πατήσει το κουμπί Edit, εμφανίζονται δύο νέα κουμπιά Save changes και Discard changes, με τα οποία δίνεται στον χρήστη η δυνατότητα να αλλάξει το κείμενο, αλλά και το περιεχόμενο του αρχείου, αν πατήσει το Save changes. Αν πατήσει το Discard changes, δεν γίνεται καμία αλλαγή στο αρχικό κείμενο.



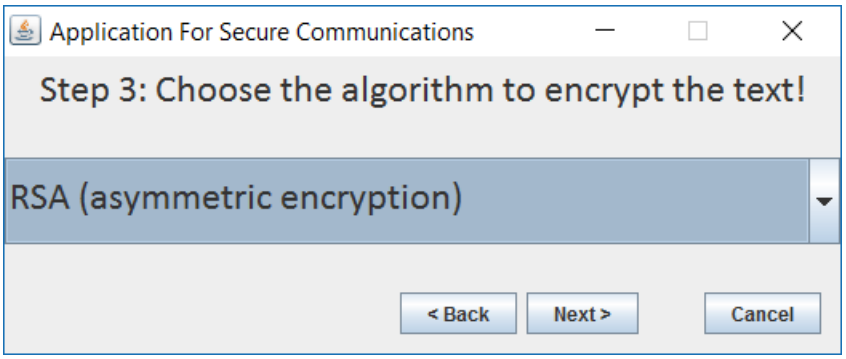


Αφού ολοκληρώσει την διαδικασία πατάει το κουμπί Next για να πάει στο επόμενο βήμα.

Στο επόμενο παράθυρο ο χρήστης μπορεί να διαλέξει μεταξύ δύο αλγορίθμων κρυπτογράφησης, τον RSA (ασύμμετρος) ή τον AES (συμμετρικός).

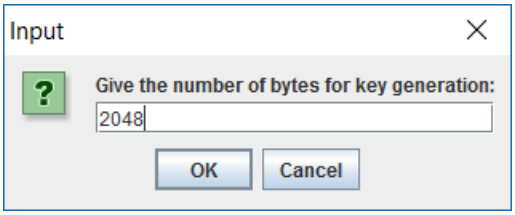


Πρώτα θα περιγράψουμε την περίπτωση που θα επιλέξει τον αλγόριθμο RSA.

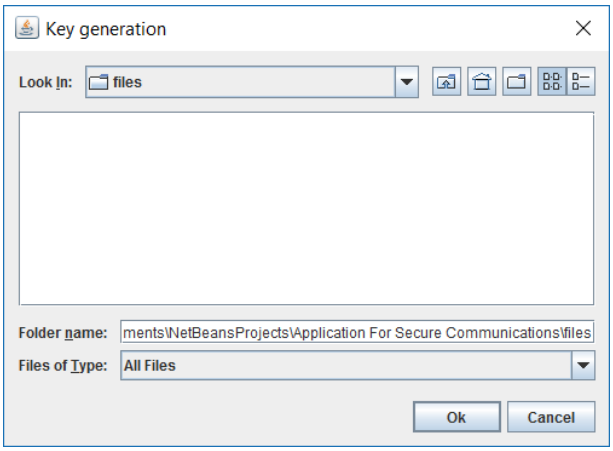


Όταν πατήσει το κουμπί Next, εμφανίζεται στον χρήστη μήνυμα το οποίο τον ρωτάει, αν θέλει να δημιουργήσει δικό του ζεύγος κλειδιών (public και private key). Να σημειώσουμε, πως αυτή η λειτουργία προστέθηκε σε περίπτωση που χρειαστεί τα κλειδιά για μελλοντική χρήση.

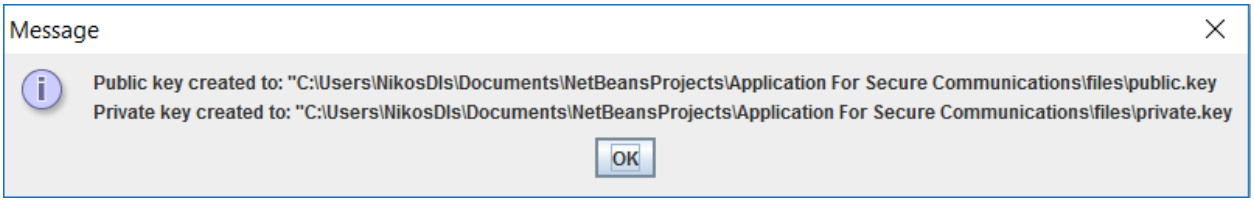
Αν ο χρήστης πατήσει το κουμπί Yes, απλά συνεχίζει στο επόμενο βήμα. Αν πατήσει το κουμπί No, τότε εμφανίζεται στον χρήση ένα παράθυρο, στο οποίο πρέπει να συμπληρώσει το μέγεθος του κλειδιού.



Και αφού πατήσει το κουμπί OK, του ζητείται να επιλέξει που θα αποθηκευτούν αυτά τα δύο κλειδιά.

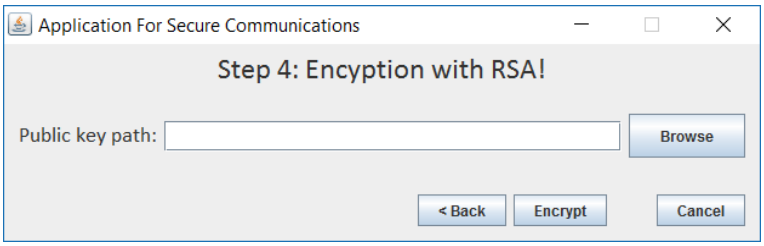


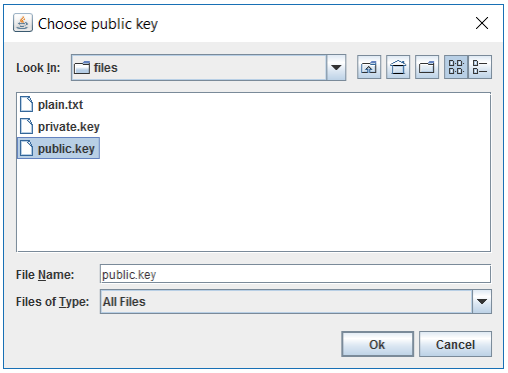
Αν όλα γίνουν σωστά, εμφανίζεται το παρακάτω μήνυμα.



Αφού πατήσει το κουμπί OK, ολοκληρώνεται η διαδικασία και περνάμε στο επόμενο βήμα.

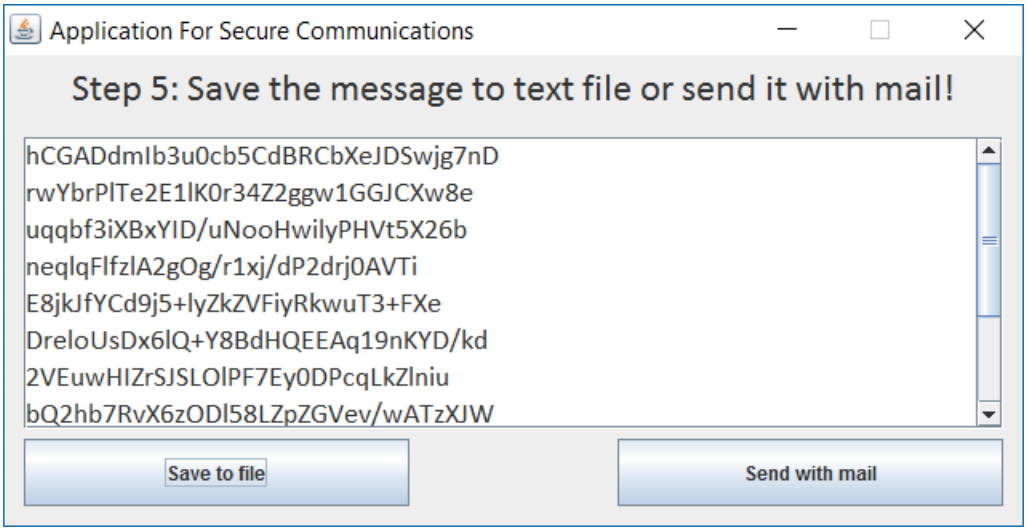
Σε αυτό το παράθυρο ο χρήστης πρέπει να συμπληρώσει την διαδρομή του public key από τον χρήστη, στον οποίο θέλει να στείλει το μήνυμα ή να αναζητήσει το κλειδί πατώντας το κουμπί Browse.



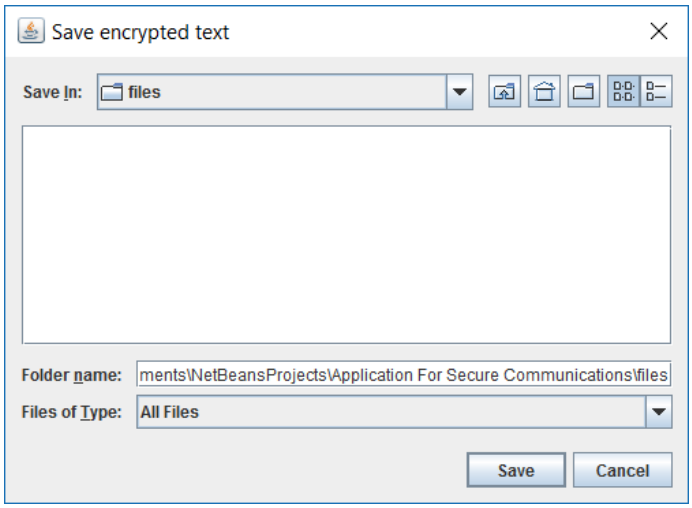


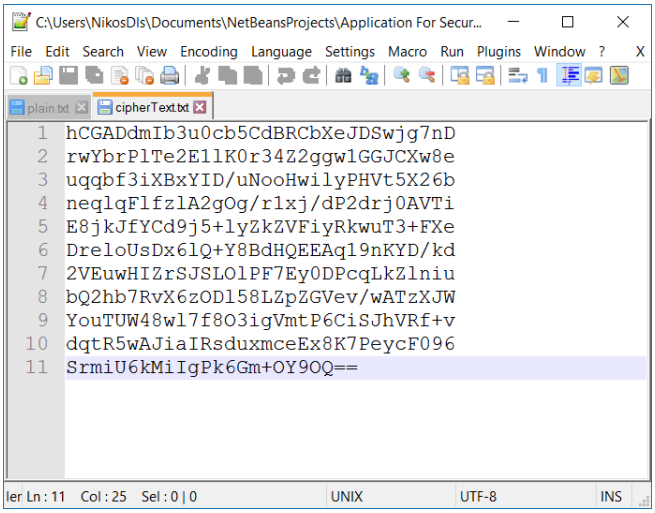
Αφού ολοκληρώσει την διαδικασία, πατάει το κουμπί Encrypt, για να πάει στο επόμενο βήμα.

Σε αυτό το παράθυρο εμφανίζεται το κρυπτογραφημένο μήνυμα και δίνεται στον χρήστη η δυνατότητα να το αποθηκεύσει σε κάποιο αρχείο και να επιλέξει ο ίδιος τον τρόπο που επιθυμεί να το στείλει στο χρήστη ή να το στείλει κατευθείαν μέσω email.

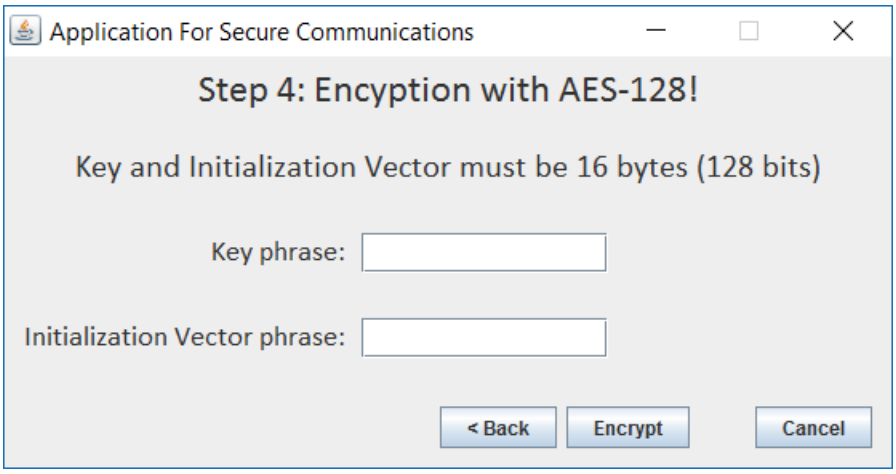


Αρχικά, θα περιγράψουμε την περίπτωση που θα πατήσει το κουμπί Save to file. Όταν το πατήσει, του ζητείται να επιλέξει που θα αποθηκευτεί το ciphertext και αφού ολοκληρωθεί και αυτή η ενέργεια, τερματίζεται η εφαρμογή. Στην συνέχεια, θα περιγράψουμε την περίπτωση που θα πατήσει το κουμπί Send with mail σε συνδυασμό της κρυπτογράφησης με την χρήση του αλγορίθμου AES.

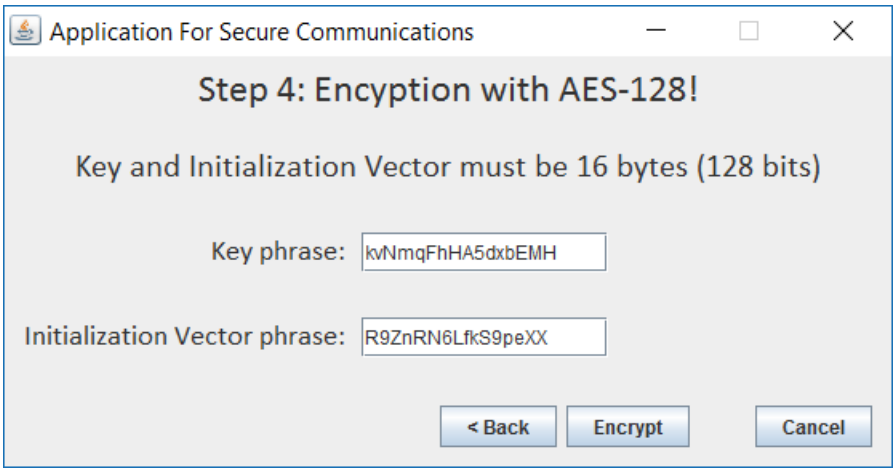




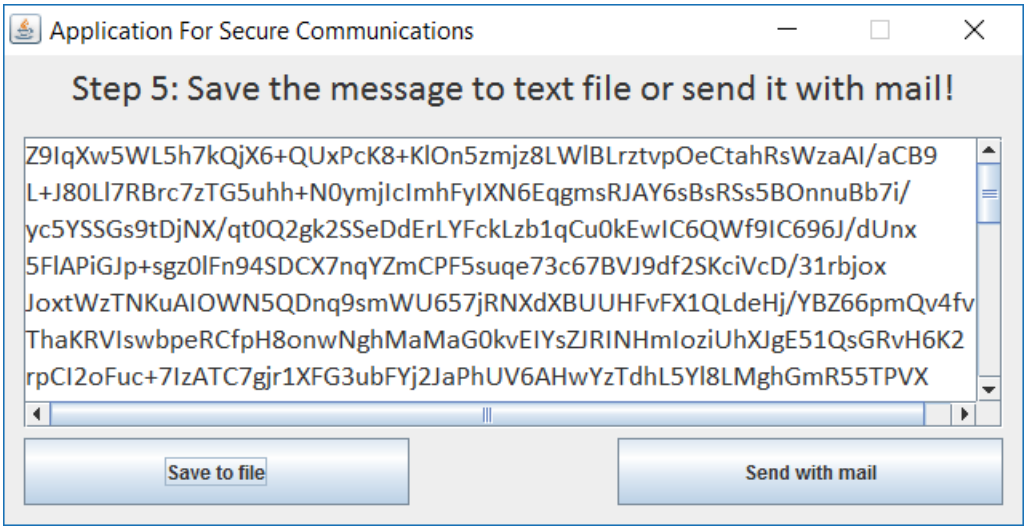
Σε αυτό το σημείο θα περιγράψουμε την διαδικασία κρυπτογράφησης με την χρήση του αλγορίθμου AES.



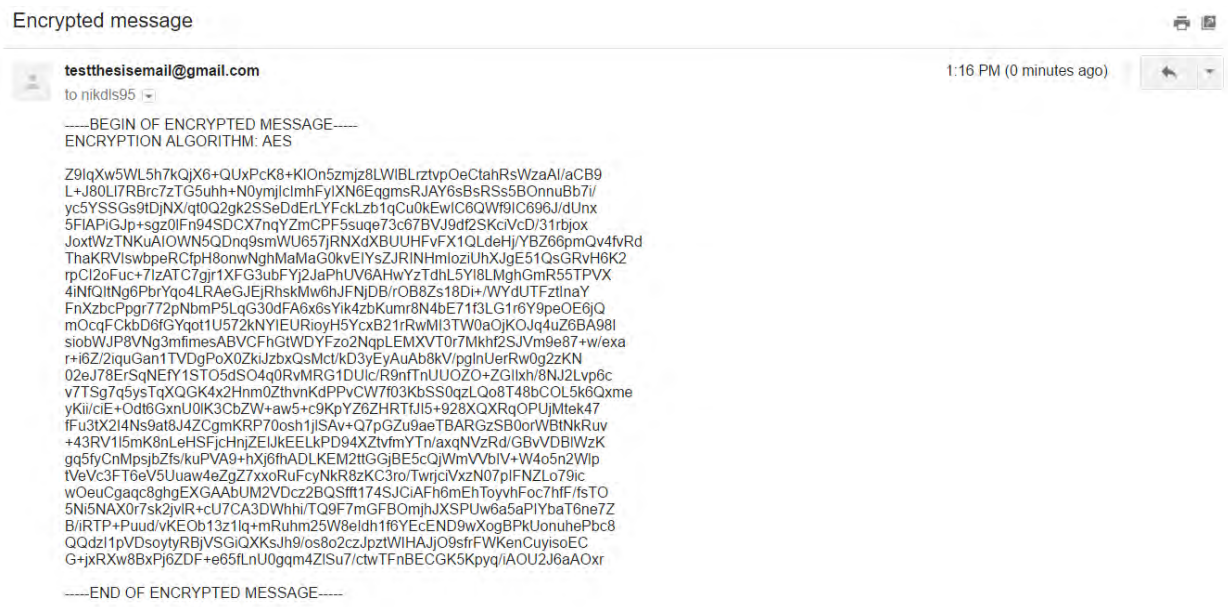
Ο χρήστης πρέπει να συμπληρώσει τα δύο παραπάνω πεδία, για να κρυπτογραφήσει το μήνυμα και να συνεχίσει στο επόμενο βήμα.



Αφού ολοκληρώσει την διαδικασία, πατάει το κουμπί Encrypt, για να πάει στο επόμενο βήμα.

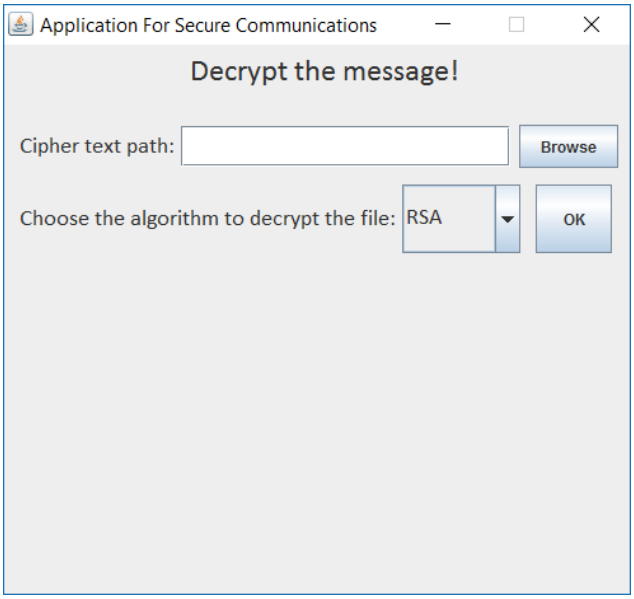


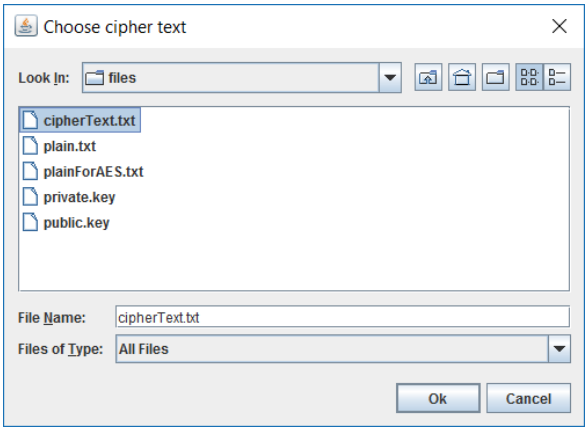
Έχουμε εξηγήσει το παραπάνω παράθυρο και αν ο χρήστης πατήσει το κουμπί Send with mail, το ciphertext θα σταλεί με email.



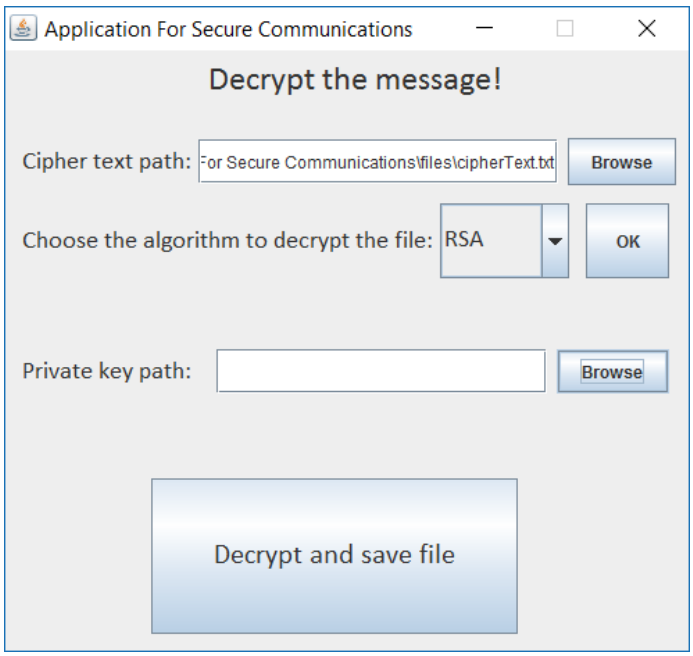
Όλες οι λειτουργείες στο κομμάτι της κρυπτογράφησης έχουν περιγραφή πλήρως και τώρα θα περάσουμε στο κομμάτι της αποκρυπτογράφησης.

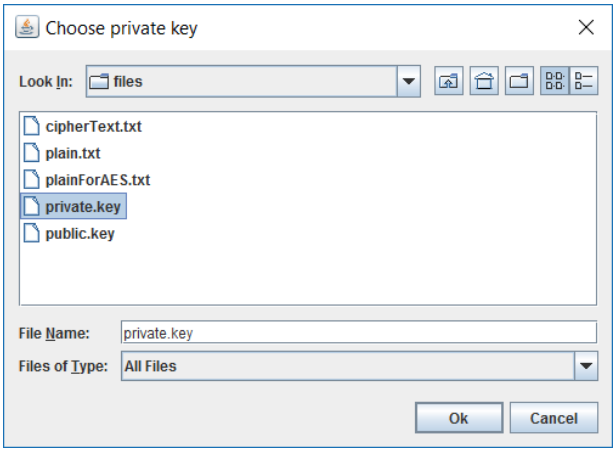
Αν ο χρήστης πατήσει στην αρχή το κουμπί Decrypt a message, του εμφανίζεται το παρακάτω παράθυρο, στο οποίο ο χρήστης πρέπει να συμπληρώσει την διαδρομή του ciphertext ή να αναζητήσει το αρχείο πατώντας το κουμπί Browse



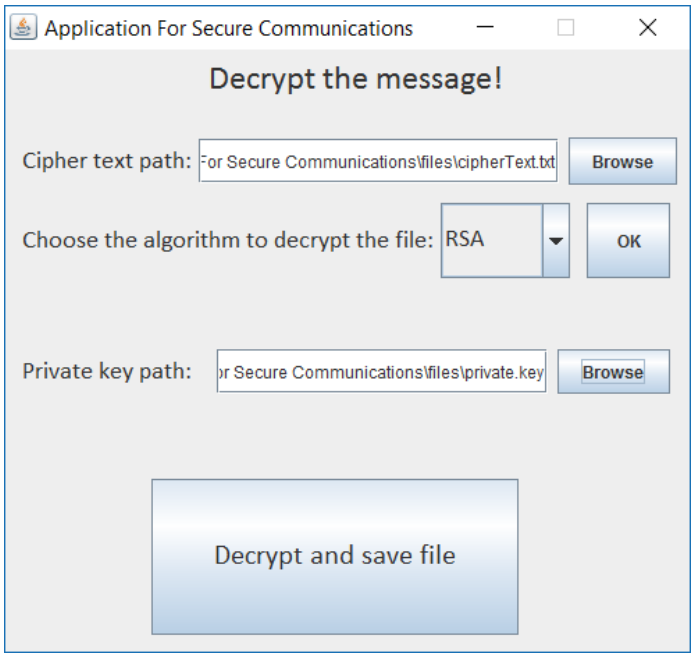


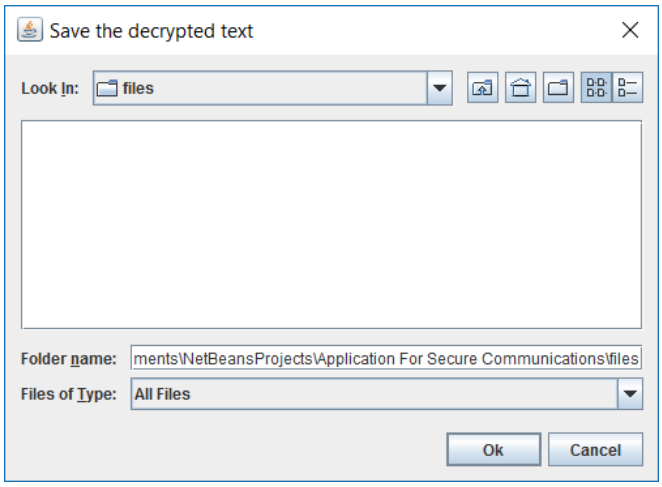
Στην συνέχεια, αν επιλέξει τον αλγόριθμο RSA, του ζητείται να δώσει και την διαδρομή του private key, για την αποκρυπτογράφηση ή να το αναζητήσει πατώντας το κουμπί Browse.



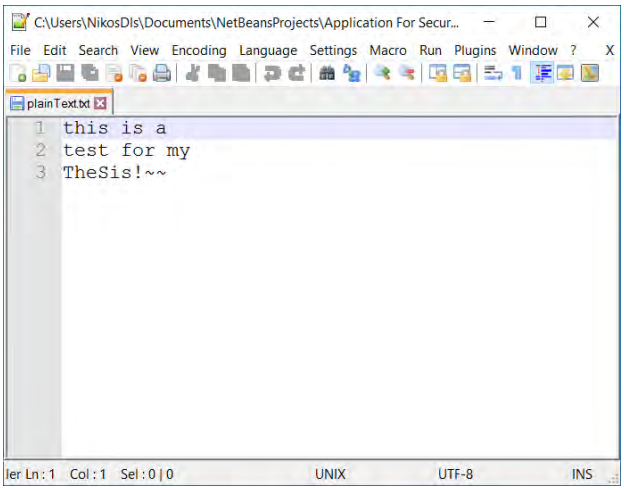


Όταν ολοκληρώσει όλες τις παραπάνω ενέργειες, πατάει το κουμπί Decrypt and save file και εμφανίζεται παράθυρο στο οποίο μπορεί να επιλέξει που θα αποθηκεύσει το αποκρυπτογραφημένο μήνυμα.

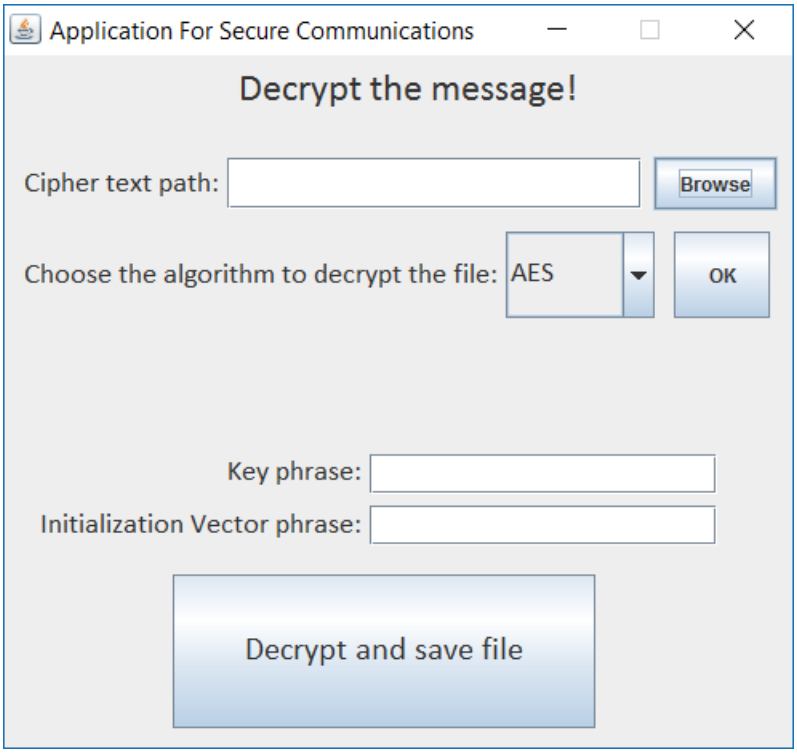




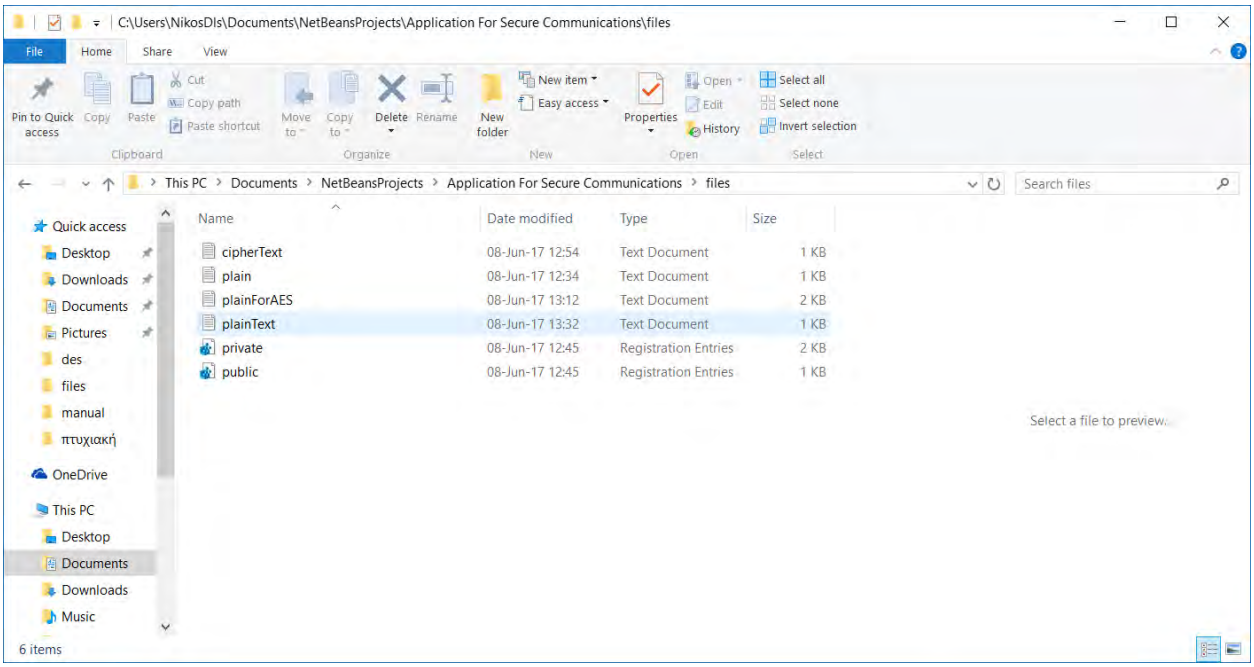
Μπορούμε να δούμε και από το αποτέλεσμα, ότι το αρχικό κείμενο είναι ίδιο με το αποκρυπτογραφημένο.



Για την αποκρυπογράφηση με τον αλγόριθμο AES η διαδικασία είναι ίδια.



Συνολικά όλα τα αρχεία που χρειαστήκαμε και δημιουργήθηκαν κατά την εκτέλεση της εφαρμογής είναι:



5. ΣΥΜΠΕΡΑΣΜΑΤΑ

Όπως φαίνεται και από την ιστορική αναδρομή, έγιναν πολλές προσπάθειες ασφαλούς κρυπτογράφησης μέχρι σήμερα και οι περισσότεροι αλγόριθμοι από αυτούς δεν είναι πλέον έγκυροι. Από αυτές τις τεχνικές κρυπτογράφησης οι οποίες εξακολουθούν να χρησιμοποιούνται μέχρι και σήμερα, ο μόνος τρόπος να προσδιοριστεί ποιες είναι “καλύτερες”, είναι με την αξιολόγηση και τη σύγκριση των διαφόρων μεθόδων. Κάθε τεχνική κρυπτογράφησης έχει τα ισχυρά της σημεία, αλλά και τις αδυναμίες της. Επίσης, μπορούμε να καταλάβουμε πως η συμμετρική και η ασύμμετρη κρυπτογράφηση, είναι δύο διαφορετικά είδη κρυπτογράφησης και για αυτό χρησιμοποιούνται για διαφορετικούς σκοπούς. Για παράδειγμα, αν χρειάζεται να κρυπτογραφήσουμε δεδομένα, ένας αλγόριθμος συμμετρικής κρυπτογράφησης θα ήταν καλύτερος, λόγω ταχύτητας, αλλά και πιο ανθεκτικός σε θέματα ασφάλειας (επιθέσεις επιλεγμένου κρυπτογραφήματος (chosen ciphertext attacks)). Από την άλλη όμως ένας αλγόριθμος ασύμμετρης κρυπτογράφησης χρησιμοποιείται σε περιπτώσεις που οι συμμετρικοί αλγόριθμοι δεν μπορούν να χρησιμοποιηθούν, όπως για παράδειγμα, την διαχείριση κλειδιών (key management). Έτσι, για να αποφασίσει κάποιος ποια τεχνική κρυπτογράφησης θα χρησιμοποιήσει, θα πρέπει να αποφασίσει τι θέλει από την κρυπτογράφηση. Τέλος, με την δημιουργία της συγκεκριμένης εφαρμογής, φαίνεται πόσο εύκολα εφαρμόζεται ένας αλγόριθμος κρυπτογράφησης στα δεδομένα, αλλά και πόσο δύσκολο είναι να εφαρμοστεί σωστά μία τεχνική κρυπτογράφησης. Η συγκεκριμένη εφαρμογή δεν ασχολείται με κομμάτι του διαμοιρασμού των κλειδιών (Έμπιστη Τρίτη Οντότητα, Αρχή Πιστοποίησης, κα), το οποίο θα μπορούσε να προστεθεί μελλοντικά και να δημιουργηθεί μία πιο ολοκληρωμένη εφαρμογή.

6. ΒΙΒΛΙΟΓΡΑΦΙΑ

ΞΕΝΟΓΛΩΣΣΗ:

Daemen, J., & Rijmen, V. (1999). AES proposal: Rijndael.

Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (1996). *Handbook of applied cryptography*. CRC press.

Morkel, T., & Eloff, J. H. P. (2004). Encryption techniques: a timeline approach. *ICSA, grant*, (2054024).

PUB, F. (1999). Data Encryption Standard (DES). *FIPS PUB*, 46-3.

Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, *21*(2), 120-126.

Standard, N. F. (2001). Announcing the advanced encryption standard (AES). *Federal Information Processing Standards Publication*, *197*, 1-51.

ΕΛΛΗΝΟΓΛΩΣΗ:

Κάτος, Β., & Στεφανίδης, Γ. (2003). *Τεχνικές κρυπτογραφίας και κρυπτανάλυσης.* Θεσσαλονίκη: Ζυγός.

Ντούσκας, Θ. (2017). Πανεπιστημιακές σημειώσεις σε ηλεκτρονική μορφή. *Crypto I - Εισαγωγικές Έννοιες.* Λαμία.

Ντούσκας, Θ. (2017). Πανεπιστημιακές σημειώσεις σε ηλεκτρονική μορφή. *Lesson IX - Data Encryption Standard (DES).* Λαμία.

Ντούσκας, Θ. (2017). Πανεπιστημιακές σημειώσεις σε ηλεκτρονική μορφή. *Lesson VII - AES.* Λαμία.

Ντούσκας, Θ. (2017). Πανεπιστημιακές σημειώσεις σε ηλεκτρονική μορφή. *Lesson VIII - RSA.* Λαμία.

ΔΙΚΤΥΑΚΟΙ ΤΟΠΟΙ

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

http://world.std.com/~cme/html/timeline.html

https://en.wikipedia.org/wiki/Vigen%C3%A8re_cipher

http://www.cryptomuseum.com/crypto/vernam.htm

https://kavaliro.com/wp-content/uploads/2014/03/AES.pdf

https://www.lri.fr/~fmartignon/documenti/systemesecurite/5-AES.pdf

http://highered.mheducation.com/sites/dl/free/007070208x/877405/Chapter_06_Data_Encription_Standard.pdf

https://asecuritysite.com/Encryption/rsa

http://doctrina.org/How-RSA-Works-With-Examples.html

https://netbeans.org

https://www.java.com