Τ.Ε.Ι. Δυτικής Ελλάδας
Τμήμα Μηχανικών Πληροφορικής Τ.Ε

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάλυσή της επίδοσής των αλγορίθμων κρυπτογράφησης στα ενσύρματα πρωτοκολλά επικοινωνίας

ΑΠΟ ΤΟΥΣ ΣΠΟΥΔΑΣΤΕΣ:

ΦΕΡΡΟΣ ΙΩΑΝΝΗΣ
ΑΜ:884
ΠΡΟΚΟΠΗ ΙΡΙΣ ΣΟΦΙΑ
ΑΜ:832

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

ΤΣΑΚΑΝΙΚΑΣ ΒΑΣΙΛΕΙΟΣ

ΝΑΥΠΑΚΤΟΣ-ΑΝΤΙΡΡΙΟ 2014

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Ναύπακτος …../…../2014

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ **ΥΠΟΓΡΑΦΗ**

1. …… ……

2. …… ……..

3. ……. ……..

ΕΙΣΑΓΩΓΗ

Το Διαδίκτυο (Internet) , φέρνοντας την τελευταία δεκαετία επανάσταση στον κόσμο των επικοινωνιών , έγινε το κυριότερο εργαλείο για την εξέλιξη και την υποστήριξη διαδικτυακών υπηρεσιών (server – client services). Ένα μέλλον γεμάτο προοπτικές διανοίχτηκε για τον κόσμο του εμπορίου χάρις στην ταχύτατη εξάπλωση της χρήσης του Διαδικύου και στη διαθεσιμότητα μηχανημάτων μεγάλης υπολογιστικής ισχύος . Στις μέρες μας αυτός ο διαδικτυακός ιστός είναι απαραίτητος σαν μηχανισμός διακίνησης πληροφοριών , σαν μέσο επικοινωνίας και συνεργασίας μεταξύ ατόμων , κυβερνητικών υπηρεσιών , οικονομικών εταιριών , ακαδημαϊκών κύκλων και επιχειρήσεων ανεξαρτήτως γεωγραφικής τοποθεσίας τους.

Παράλληλα όμως με την ανάπτυξη των τεχνολογιών αυτών , εξελίσσονται και οι επιθέσεις στα συστήματα που τις υλοποιούν. Οι επιθέσεις αυτές μπορούν να πάρουν διάφορες μορφές αλλά και να συνδιάσουν την δράση τους. Δεδομένης της κατάστασης αυτής η ασφάλεια των δικτύων και των εφαρμογών τους από κακόβουλους χρήστες είναι επιτακτική.

Είναι συνεπώς αυτονόητο το γεγονός ότι αυτή η συνεχώς αυξανόμενη χρήση του Διαδικτύου έχει ως άμεση συνέπεια τα προβλήματα ασφάλειας που προκύπτουν να αντιμετωπίζονται απο τους προγραμματιστές με μέγιστη προσοχή και σαφήνεια. Το κυριότερο εργαλείο για την αντιμετώπιση επιθέσεων ασφάλειας είναι η κρυπτογραφία. Η κρυπτογραφία παρέχει στους προγραμματιστές τις απαραίτητες τεχνικές για την διατήρηση της μυστικότητας της πληροφορίας (Confidentiality) , βοηθά στην εξασφάλιση της ακεραιότητάς της (Integrity) , στην πιστοποίηση της προέλευσής της (Authentication) καθώς επίσης και στην μη δυνατότητα άρνησης αποστολής της από ένα χρήστη (Non-repudiation) .

Στην εργασία αυτή λοιπόν μελετάμε και αναλύουμε την επίδοση των βασικών αλγορίθμων κρυπτογράφησης στα ενσύρματα πρωτοκολλά επικοινωνίας δίνοντας παράλληλα και κάποια παραδείγματα σύγκρισης αυτών. Ως συνέχεια του θέματος θα ήταν ενδιαφέρον να αναλυθεί η σύγκριση λειτουργίας ενσύρματων πρωτοκόλλων σε σχέση με ασύρματα.

Π Ε Ρ Ι Ε Χ Ο Μ Ε Ν Α

Κεφάλαιο III

Κεφάλαιο IV

Κεφάλαιο V

Περίληψη

Στην παρακάτω πτυχιακή εργασία μελετάμε και αναλύουμε την επίδοση των βασικών αλγορίθμων κρυπτογράφησης στα ενσύρματα πρωτοκολλά επικοινωνίας δίνοντας παράλληλα και κάποια παραδείγματα σύγκρισης αυτών.
Αρχικά ανατρέχουμε σε ιστορικές αναφορές για την κρυπτογραφία στα αρχαία χρόνια περιγράφοντας τους τότε τρόπους κρυπτογράφησης και τις εφαρμογές της σε διάφορους τομείς. Αναγραφεται η εξέλιξη της κρυπτογραφίας και οι πρώτοι τύποι αλγορίθμων που χρησιμοποιήθηκαν σε εποχές υπολογιστών.
Στο δεύτερο κεφάλαιο έχουμε πλήρη ανάλυση και περιγραφή των κυριότερων αλγορίθμων ανα κατηγορία που χρησιμοποιούνται.Χωρίζονται σε συμμετρικοί μπλοκ αλγόριθμοι (block ciphers) , συμμετρικοί αλγόριθμοι ροής (Stream Ciphers) και ασύμμετροι αλγόριθμοι. Περιγράφονται πλήρη χαρακτηριστικά και χρήσεις για τον κάθε αλγόριθμο ξεχωριστά.
Στη συνέχεια υλοποιούνται και συγκρίνονται βασικοί και διαδεδομένοι αλγόριθμοι παλαιότερης εποχής και σύγχρονοι , κρυπτογραφόντας ενα συγκεκριμένο κέιμενο.Γίνεται αναλυτική κρυπτογράφιση με όλα τα βήματα που χρειάζονται για τη διαδικασία.Παρατηρούνται παρόμοιοι τρόποι λειτουργίας μεταξύ τους σε ορισμένους αλγόριθμους και σε άλλους πολυπλοκότητα και χρήση. Πραγματοποιείται επίσης σύγκριση size overhead για τους αλγορίθμους που χρησιμοποιούνται επισυνάπτοντας πίνακες και διαγράμματα σύγκρισης όπως επίσης και διαχωρισμός αλγορίθμων σε σχέση χρήσης μνήμης κατή τη διάρκεια κρυπτογράφησης.
Στο τέταρτο κεφάλαιο της πτυχιακής , καταμετράμε τα βήματα που χρειάζεται ο κάθε αλγόριθμος που χρησιμοποιήθηκε στο πείραμά μας ξεχωριστά , ώστε να συγκριθεί ο χρόνος που χρειάζεται για κάθε κρυπτογράφιση και να βρούμε ποιός είναι προτιμότερος στην περίπτωση μας..
Τέλος στο $5^{ο}$ και τελευταίο κεφάλαιο της εργασίας μας , αναλύουμε παρόμοια πειράματα με το δικό μας που έχουν καταγραφεί μαζί με τις βιβλιογραφικές αναφορές τους και τα συμπεράσματα.

Abstract

In the following graduation project we study and analyze the performance of basic encryption algorithms in wired communication protocols providing as well some comparing examples.
Firstly we look at historical references about the cryptography in ancient times and describing older encryption modes and applications in various fields. We have indicate the evolution of cryptography and the first types of algorithms used in computer seasons.
In the second chapter we have a full analysis and description of the main algorithms that are used in each category. They are divided into symmetric block algorithms (block ciphers), symmetric algorithms flow (Stream Ciphers) and asymmetric algorithms. We describe full features and uses for each algorithm separately.
Then we implemented and compared basic and widely known algorithms of an older and modern period and cryptographic in a specific text of 128 bits. In addition we run the analytical encryption with all the steps that we needed for the process. There are observed similarities modes between some algorithms and in others complexity and more use. We have also compared the size overhead for the algorithms that we used by attaching tables and comparison charts also recommend separation algorithms comparing memory usage during encryption.
In the fourth chapter we count the steps that we need by any algorithm which was used in our experiment separately to compare the time we needed for each encryption and find what is preferable in our case.
Finally in the fifth and final chapter of our project, we analyze experiments similar to ours recorded with the bibliographic reference and conclusions.

ΚΕΦΑΛΑΙΟ Ι

Κρυπτογραφία και Ιστορία

Εισαγωγή: Η κρυπτογραφία, η επιστήμη της κρυπτογράφησης και αποκρυπτογράφησης πληροφοριών, μπορεί να χαρακτηριστεί σαν ένα από τα αρχαιότερα επαγγέλματα της ανθρωπότητας, έχοντας τις ρίζες της βαθιά στο παρελθόν

1.1 Η Κρυπτογραφία στους αρχαίους χρόνους.

Η εξέλιξη της κρυπτογραφίας χωρίζεται ιστορικά σε τρεις περιόδους:

- Η πρώτη ξεκινά περίπου το 1900 π.Χ. και τερματίζεται στις αρχές του εικοστού αιώνα.
- Η δεύτερη περίοδος, ουσιαστικά καλύπτει το πρώτο μισό του εικοστού αιώνα.
- Η τρίτη και τελευταία περίοδος ξεκινά το 1950, με την έξαρση της ανάπτυξης στους επιστημονικούς κλάδους των μαθηματικών, της μικροηλεκτρονικής και των υπολογιστικών συστημάτων και συνεχίζεται μέχρι και τις ημέρες μας.

Κύριο χαρακτηριστικό των παλαιότερων μορφών κρυπτογράφησης ήταν ότι η επεξεργασία γινόταν πάνω στην γλωσσική δομή. Στις νεότερες μορφές η κρυπτογραφία κάνει χρήση του αριθμητικού ισοδύναμου, η έμφαση έχει μεταφερθεί σε διάφορα πεδία των μαθηματικών, όπως διακριτά μαθηματικά, θεωρία αριθμών, θεωρία πληροφορίας, υπολογιστική πολυπλοκότητα, στατιστική και συνδυαστική ανάλυση.

Α. Πρώτη Περίοδος Κρυπτογραφίας (1900 π.Χ. – 1900 μ.Χ.) **[1]**

Κατά τη διάρκεια αυτής της περιόδου αναπτύχθηκε μεγάλο πλήθος μεθόδων και αλγορίθμων κρυπτογράφησης, που βασίζονταν κυρίως σε απλές αντικαταστάσεις γραμμάτων. Όλες αυτές δεν απαιτούσαν εξειδικευμένες γνώσεις και πολύπλοκες συσκευές, αλλά στηρίζονταν στην ευφυΐα και την ευρηματικότητα των δημιουργών τους. Όλα αυτά τα συστήματα έχουν στις μέρες μας κρυπταναλυθεί και έχει αποδειχθεί ότι, εάν είναι γνωστό ένα μεγάλο κομμάτι του κρυπτογραφημένου μηνύματος, τότε το αρχικό κείμενο μπορεί σχετικά εύκολα να επανακτηθεί.
Όπως προκύπτει από μία μικρή σφηνοειδή επιγραφή, που ανακαλύφθηκε στις όχθες του ποταμού Τίγρη, οι πολιτισμοί που αναπτύχθηκαν στη Μεσοποταμία ασχολήθηκαν με την κρυπτογραφία ήδη από το 1500 π.Χ.. Επίσης, ως το αρχαιότερο βιβλίο κρυπτοκωδικών στον κόσμο, θεωρείται μία σφηνοειδής επιγραφή στα Σούσα της Περσίας, η οποία περιλαμβάνει τους αριθμούς 1 έως 8 και από το 32 έως το 35, τοποθετημένους τον ένα κάτω από τον άλλο, ενώ απέναντι τους βρίσκονται τα αντίστοιχα για τον καθένα σφηνοειδή σύμβολα.
Η πρώτη στρατιωτική χρήση της κρυπτογραφίας αποδίδεται στους Σπαρτιάτες. Γύρω στον 5ο π.Χ. αιώνα εφηύραν την «σκυτάλη», την πρώτη κρυπτογραφική συσκευή, στην οποία χρησιμοποίησαν για την κρυπτογράφηση τη μέθοδο της αντικατάστασης. Όπως αναφέρει ο Πλούταρχος, η «Σπαρτιατική Σκυτάλη» **ΕΙΚΟΝΑ 1**, ήταν μια

ξύλινη ράβδος, ορισμένης διαμέτρου, γύρω από την οποία ήταν τυλιγμένη ελικοειδώς μια λωρίδα περγαμηνής. Το κείμενο ήταν γραμμένο σε στήλες, ένα γράμμα σε κάθε έλικα, όταν δε ξετύλιγαν τη λωρίδα, το κείμενο ήταν ακατάληπτο εξαιτίας της αναδιάταξης των γραμμάτων. Το «κλειδί» ήταν η διάμετρος της σκυτάλης.

ΕΙΚΟΝΑ 1.Σπαρτιατική Σκυτάλη, μια πρώιμη συσκευή για την κρυπτογράφηση

Στην αρχαιότητα χρησιμοποιήθηκαν κυρίως συστήματα, τα οποία βασίζονταν στη στεγανογραφία και όχι τόσο στην κρυπτογραφία. Μολονότι ο διαχωρισμός της στεγανογραφίας από την κρυπτογραφία είναι λεπτός, εντούτοις σημαντικές είναι οι τεχνικές διαφοροποιήσεις οι οποίες παρατηρούνται μεταξύ των δύο αυτών εννοιών. Η στεγανογραφία (Steganography) είναι η τεχνική της απόκρυψης της ίδιας της ύπαρξης της πληροφορίας, ενώ αντίθετα η κρυπτογραφία μετασχηματίζει το μήνυμα έτσι ώστε να το καθιστά ακατανόητο σε οποιοδήποτε τρίτο. Στα συστήματα της κρυπτογραφίας (cryptography) οι πληροφορίες οι οποίες πρόκειται να αποσταλούν κωδικοποιούνται με ένα κλειδί και μόνο το πρόσωπο το οποίο διαθέτει το κλειδί είναι σε θέση να αποκρυπτογραφήσει να αναγνώσει το μήνυμα. Η στεγανογραφία σε σύγκριση με την κρυπτογραφία, προσθέτει ακόμα ένα επίπεδο ενισχύοντας την ασφάλεια των ευαίσθητων προς αποστολή αρχείων. Στην περίπτωση που κάποιος τρίτος αποκτήσει πρόσβαση σε κάποιο αρχείο το οποίο προστατεύεται από κωδικό, η χρήση των κατάλληλων εργαλείων μπορεί να τον οδηγήσουν στην αποκάλυψη του κωδικού προστασίας.

Οι Έλληνες συγγραφείς δεν αναφέρουν αν και πότε χρησιμοποιήθηκαν συστήματα γραπτής αντικατάστασης γραμμάτων, αλλά τα βρίσκουμε στους Ρωμαίους, κυρίως την εποχή του Ιουλίου Καίσαρα. Ο Ιούλιος Καίσαρας έγραφε στον Κικέρωνα και σε άλλους φίλους του, αντικαθιστώντας τα γράμματα του κειμένου, με γράμματα, που βρίσκονται 3 θέσεις μετά, στο Λατινικό Αλφάβητο. Έτσι, σήμερα, το σύστημα κρυπτογράφησης που στηρίζεται στην αντικατάσταση των γραμμάτων του αλφαβήτου με άλλα που βρίσκονται σε καθορισμένο αριθμό θέσης πριν ή μετά, λέγεται κρυπτοσύστημα αντικατάστασης του Καίσαρα.Ο Καίσαρας χρησιμοποίησε και άλλα, πιο πολύπλοκα συστήματα κρυπτογράφησης, για τα οποία έγραψε ένα βιβλίο ο Valerius Probus, το οποίο δυστυχώς δεν διασώθηκε, αλλά αν και χαμένο,

θεωρείται το πρώτο βιβλίο κρυπτολογίαςΤο σύστημα αντικατάστασης του Καίσαρα, χρησιμοποιήθηκε ευρύτατα και στους επόμενους αιώνες.
Στη διάρκεια του Μεσαίωνα, η κρυπτολογία ήταν κάτι το απαγορευμένο και αποτελούσε μια μορφή αποκρυφισμού και μαύρης μαγείας, κάτι που συντέλεσε στην καθυστέρηση της ανάπτυξης της. Η εξέλιξη, τόσο της κρυπτολογίας, όπως και των μαθηματικών, συνεχίζεται στον Αραβικό κόσμο. Στο γνωστό μυθιστόρημα «Χίλιες και μία νύχτες» κυριαρχούν οι λέξεις-αινίγματα, οι γρίφοι, τα λογοπαίγνια και οι αναγραμματισμοί. Έτσι, εμφανίστηκαν βιβλία που περιείχαν κρυπταλφάβητα, όπως το αλφάβητο «Dawoudi» που πήρε το όνομα του από τον βασιλιά Δαυίδ. Οι Άραβες είναι οι πρώτοι που επινόησαν αλλά και χρησιμοποίησαν μεθόδους κρυπτανάλυσης. Το κυριότερο εργαλείο στην κρυπτανάλυση, η χρησιμοποίηση των συχνοτήτων των γραμμάτων κειμένου, σε συνδυασμό με τις συχνότητες εμφάνισης στα κείμενα των γραμμάτων της γλώσσας, επινοήθηκε από αυτούς γύρω στον 14ο αιώνα. Η κρυπτογραφία, λόγω των στρατιωτικών εξελίξεων, σημείωσε σημαντική ανάπτυξη στους επόμενους αιώνες. Ο Ιταλός Giovanni Batista Porta, το 1563, δημοσίευσε το περίφημο για την κρυπτολογία βιβλίο «De furtivis literarum notis», με το οποίο έγιναν γνωστά τα πολυαλφαβητικά συστήματα κρυπτογράφησης και τα διγραφικά κρυπτογραφήματα, στα οποία, δύο γράμματα αντικαθίστανται από ένα. Σημαντικός εκπρόσωπος εκείνης της εποχής είναι και ο Γάλλος Vigenere, του οποίου ο πίνακας πολυαλφαβητικής αντικατάστασης, χρησιμοποιείται ακόμη και σήμερα.

Β. Δεύτερη Περίοδος Κρυπτογραφίας (1900 μ.Χ. – 1950 μ.Χ.) **[1]**

Η δεύτερη περίοδος της κρυπτογραφίας όπως προαναφέρθηκε τοποθετείται στις αρχές του εικοστού αιώνα και φτάνει περίπου μέχρι το 1950. Καλύπτει, επομένως, τους δύο παγκόσμιους πολέμους, εξαιτίας των οποίων (λόγω της εξαιρετικά μεγάλης ανάγκης που υπήρξε για ασφάλεια κατά την μετάδοση ζωτικών πληροφοριών μεταξύ των στρατευμάτων των εμπλεκομένων χωρών) αναπτύχθηκε η κρυπτογραφία τόσο όσο δεν είχε αναπτυχθεί τα προηγούμενα 3000 χρόνια. Τα κρυπτοσυστήματα αυτής της περιόδου αρχίζουν να γίνονται πολύπλοκα, και να αποτελούνται από μηχανικές και ηλεκτρομηχανικές κατασκευές, οι οποίες ονομάζονται «κρυπτομηχανές». Η κρυπτανάλυση τους, απαιτεί μεγάλο αριθμό προσωπικού, το οποίο εργαζόταν επί μεγάλο χρονικό διάστημα ενώ ταυτόχρονα γίνεται εξαιρετικά αισθητή η ανάγκη για μεγάλη υπολογιστική ισχύ. Παρά την πολυπλοκότητα που αποκτούν τα συστήματα κρυπτογράφησης κατά την διάρκεια αυτής της περιόδου η κρυπτανάλυση τους είναι συνήθως επιτυχημένη. Η πιο γνωστή κρυπτογραφική μηχανή εκείνης της περιόδου είναι το Enigma (**ΕΙΚΟΝΑ 2**). Η φήμη της πηγάζει κυρίως από τον αποφασιστικό ρόλο που διαδραμάτισε η αποκρυπτογράφησή της στην τελική έκβαση του δευτέρου παγκοσμίου πολέμου.

ΕΙΚΟΝΑ 2. Κρυπτόμηχανή Enigma

Το όνομα Enigma οι δημιουργοί της το δανείστηκαν από την ελληνική λέξη αίνιγμα και με αυτό ήθελαν να δώσουν έμφαση στην περίπλοκη δομή της, καθώς και στην απόλυτη ασφάλεια των μηνυμάτων που αυτή κρυπτογραφούσε. Το παραπάνω σύστημα χρησιμοποιήθηκε εκτεταμένα από τους Γερμανούς, σε διάφορες παραλλαγές του. Παρόλα αυτά δεν κατάφερε να εξασφαλίσει το απόρρητο των επικοινωνιών των γερμανικών δυνάμεων και η αποκρυπτογράφηση της έδωσε ένα σημαντικό πλεονέκτημα στις συμμαχικές δυνάμεις έναντι αυτών του άξονα. Οι συμμαχικές κρυπτομηχανές που χρησιμοποιήθηκαν στον δεύτερο παγκόσμιο πόλεμο περιλάμβαναν το βρετανικό TypeX και το αμερικανικό SIGABA. Και τα δύο ήταν ηλεκτρομηχανικά σχέδια παρόμοια στο πνεύμα με το Enigma, με σημαντικές εν τούτοις βελτιώσεις. Κανένα δεν έγινε γνωστό ότι παραβιάστηκε κατά τη διάρκεια του πολέμου. Τα στρατεύματα στο πεδίο μάχης χρησιμοποίησαν το M-209 και τη λιγότερη ασφαλή οικογένεια κρυπτομηχανών M-94. Οι Βρετανοί πράκτορες της Υπηρεσίας "SOE" χρησιμοποίησαν αρχικά ένα τύπο κρυπτογραφίας που βασιζόταν σε ποιήματα (τα απομνημονευμένα ποιήματα ήταν τα κλειδιά).

Γ. Τρίτη Περίοδος Κρυπτογραφίας (1950 μ.Χ. - Σήμερα) **[1]**

Αυτή η περίοδος χαρακτηρίζεται από την έξαρση της ανάπτυξης στους επιστημονικούς κλάδους των μαθηματικών, της μικροηλεκτρονικής και των υπολογιστικών συστημάτων. Η εποχή της σύγχρονης κρυπτογραφίας αρχίζει ουσιαστικά με τον Claude Shannon, αναμφισβήτητα ο πατέρας των μαθηματικών συστημάτων κρυπτογραφίας. Το 1949 δημοσίευσε το έγγραφο «Θεωρία επικοινωνίας των συστημάτων μυστικότητας» (Communication Theory of Secrecy Systems) στο τεχνικό περιοδικό Bell System και λίγο αργότερα στο βιβλίο του, «*Μαθηματική Θεωρία της Επικοινωνίας*» (Mathematical Theory of Communication), μαζί με τον Warren Weaver. Αυτά, εκτός από τις άλλες εργασίες του επάνω στη θεωρία δεδομένων και επικοινωνίας καθιέρωσε μια στερεά θεωρητική βάση για την κρυπτογραφία και την κρυπτανάλυση. Εκείνη την εποχή η κρυπτογραφία εξαφανίζεται και φυλάσσεται από τις μυστικές υπηρεσίες κυβερνητικών επικοινωνιών όπως η NSA. Πολύ λίγες εξελίξεις δημοσιοποιήθηκαν ξανά μέχρι τα μέσα της δεκαετίας του '70, όταν όλα άλλαξαν.

Στα μέσα της δεκαετίας του '70 έγιναν δύο σημαντικές δημόσιες (δηλ. μη-μυστικές) πρόοδοι. Πρώτα ήταν η δημοσίευση του σχεδίου προτύπου κρυπτογράφησης DES (Data Encryption Standard) στον ομοσπονδιακό κατάλογο της Αμερικής στις 17 Μαρτίου 1975. Το προτεινόμενο DES υποβλήθηκε από την IBM, στην πρόσκληση του Εθνικού Γραφείου των Προτύπων (τώρα γνωστό ως NIST), σε μια προσπάθεια να αναπτυχθούν ασφαλείς ηλεκτρονικές εγκαταστάσεις επικοινωνίας για επιχειρήσεις όπως τράπεζες και άλλες μεγάλες οικονομικές οργανώσεις. Μετά από τις συμβουλές και την τροποποίηση από την NSA, αυτό το πρότυπο υιοθετήθηκε και δημοσιεύθηκε ως ένα ομοσπονδιακή τυποποιημένο πρότυπο επεξεργασίας πληροφοριών το 1977 (αυτήν την περίοδο αναφέρεται σαν FIPS 46-3). Ο DES ήταν ο πρώτος δημόσια προσιτός αλγόριθμος κρυπτογράφησης που εγκρίνεται από μια εθνική αντιπροσωπεία όπως η NSA. Η απελευθέρωση της προδιαγραφής της από την NBS υποκίνησε μια έκρηξη δημόσιου και ακαδημαϊκού ενδιαφέροντος για τα συστήματα κρυπτογραφίας. Ο DES αντικαταστάθηκε επίσημα από τον AES το 2001 όταν ανήγγειλε ο NIST το FIPS 197. Μετά από έναν ανοικτό διαγωνισμό, ο NIST επέλεξε τον αλγόριθμο Rijndael, που υποβλήθηκε από δύο Φλαμανδούς κρυπτογράφους, για να είναι το AES. Ο DES και οι ασφαλέστερες παραλλαγές του όπως ο 3DES ή TDES χρησιμοποιούνται ακόμα σήμερα, ενσωματωμένος σε πολλά εθνικά και οργανωτικά πρότυπα. Εντούτοις, το βασικό μέγεθος των 56-bit έχει αποδειχθεί ότι είναι ανεπαρκές να αντισταθεί στις επιθέσεις ωμής βίας (μια τέτοια επίθεση πέτυχε να σπάσει τον DES σε 56 ώρες ενώ το άρθρο που αναφέρεται ως το σπάσιμο του DES δημοσιεύτηκε από τον O'Reilly and Associates). Κατά συνέπεια, η χρήση απλής κρυπτογράφησης με τον DES είναι τώρα χωρίς την αμφιβολία επισφαλής για χρήση στα νέα σχέδια των κρυπτογραφικών συστημάτων και μηνύματα που προστατεύονται από τα παλαιότερα κρυπτογραφικά συστήματα που χρησιμοποιούν DES, και όλα τα μηνύματα που έχουν αποσταλεί από το 1976 με τη χρήση DES, διατρέχουν επίσης σοβαρό κίνδυνο αποκρυπτογράφησης. Ανεξάρτητα από την έμφυτη ποιότητά του, το βασικό μέγεθος του DES (56-bit) ήταν πιθανά πάρα πολύ μικρό ακόμη και το 1976, πράγμα που είχε επισημάνει ο Whitfield Diffie. Υπήρξε επίσης η υποψία ότι κυβερνητικές οργανώσεις είχαν ακόμα και τότε ικανοποιητική υπολογιστική δύναμη ώστε να σπάσουν μηνύματα που είχαν κρυπτογραφηθεί με τον DES.

1.2 Η Κρυπτογραφία στην εποχή των υπολογιστών. **[2]**

Μια ξεχωριστή κατηγορία στο χώρο της κρυπτογραφίας αποτελούν οι συσκευές που χρησιμοποιούσαν κάποιο μηχανικό τρόπο για την κρυπτογράφηση και αποκρυπτογράφηση των μηνυμάτων. Παρακάτω αναφέρουμε μερικές από αυτές.

Jefferson cylinder: αναπτύχθηκε το 1790 και αποτελούνταν από 36 δίσκους. Ο κάθε δίσκος είχε ένα τυχαίο αλφάβητο και η σειρά των δίσκων ήταν το κλειδί αποκρυπτογράφησης.

Wheatstone disc: ανακαλύφτηκε από τον Wadsworth το 1817 και αναπτύχθηκε από τον Wheatstone το 1860. Αποτελούνταν από δύο τροχούς που χρησιμοποιούνταν για τη δημιουργία ενός πολυαλφαβητικού αλγορίθμου.

Hagelin machine: Τη δεκαετία του 1930, ο Σουηδός Boris Hagelin επινόησε μια μηχανή κρυπτογρά-φησης βασισμένη στην τεχνική της πολυαλφαβητικής αντικατάστασης [LUB1998].Η μηχανή αυτή έκανε χρήση ενός συνόλου αλφαβήτων, του

ονομαζόμενου τετραγώνου του Beaufort, συγκρίσιμου με τον Πίνακα του Vigenére, αλλά με τα γράμματα των αλφαβήτων σε αντίστροφη σειρά. Το κρυπτόγραμμα υπολογίζονταν ως εξής: $y = b + 1 - x \pmod{26}$, όπου x είναι η αλφαβητική θέση του γράμματος του καθαρού κειμένου, παραδείγματος χάρη του «a» το 1, του «b» το «2» κ.ο.κ., το b είναι η γραμμή του τετραγώνου του Beaufort που υπαγορεύεται από το κλειδί και το y η αλφαβητική θέση του κώδικα. Αυτό επαναλαμβανόταν για όλα τα γράμματα του μηνύματος

Enigma Rotor machine:μια από τις πολύ σημαντικές κατηγορίες μηχανών κρυπτογράφησης.Χρησιμοποιήθηκε πολύ κατά τον 2ο Παγκόσμιο Πόλεμο. Αποτελούνταν από μια σειρά περιστρεφόμενους τροχούς με εσωτερικές διασυνδέσεις που παρείχαν την αντικατάσταση χρησιμοποιώντας ένα αλφάβητο που συνεχώς άλλαζε. Ήταν βασισμένη σε ένα σχέδιο που αναπτύχθηκε από τον Arthur Scherbius το 1910. Αποτελείται από τρία μέρη που συνδέονταν με σύρματα. Ένα πληκτρολόγιο εισόδου, τη μονάδα κρυπτογράφησης και ενδεικτικές λυχνίες. Η μονάδα κρυπτογράφησης περιστρεφόταν κατά μια ορισμένη γωνία κάθε φορά που ένα γράμμα κρυπτογραφούνταν. Η μηχανή που χρησιμοποιήθηκε κατά τον 2ο Παγκόσμιο Πόλεμο είχε τρεις μονάδες κρυπτογράφησης.

Η αποκρυπτογράφηση της μηχανής Enigma έγινε αφού ένας Γερμανός (ο Hans-Thilo Schmidt) έδωσε κάποια βιβλία κωδικών σε ένα Γάλλο που με τη σειρά του τα έδωσε στον Poles. Βασικές τεχνικές αναπτύχθηκαν από τη Marian Rejewski και επεκτάθηκαν από την Αγγλική αντικατασκοπία με αποτέλεσμα το σπάσιμο του κώδικα. Μετά την πρώτη αποκρυπτογράφηση του κώδικα οι Γερμανοί άλλαξαν τον κώδικα, αλλά δεύτερη διαρροή και συντονισμένες προσπάθειες οδήγησαν ξανά στο σπάσιμό του.

1.2.1 Οι πρώτοι αλγόριθμοι κρυπτογράφησης

Αλγόριθμος του Καίσαρα [3]

Ο αλγόριθμος του Καίσαρα αποτελεί μια ειδική κατηγορία των κρυπτογραφικών αλγορίθμων απλής αντικατάστασης (simple substitution cipher). Σε αυτόν τον κρυπτογραφικό αλγόριθμο, το κλειδί αποτελεί μια μετάθεση των γραμμάτων της αλφαβήτου. Η κρυπτογράφηση περιλαμβάνει αντικατάσταση κάθε γράμματος με το αντίστοιχο γράμμα που προκύπτει από τη μετάθεση. Αντίστοιχα, η αποκρυπτογράφηση γίνεται με χρήση της ανάστροφης μετάθεσης.
Στον κρυπτογραφικό αλγόριθμο του Καίσαρα (Caesar cipher) το μήνυμα (αρχικό κείμενο) πρέπει να είναι μια ακολουθία από γράμματα. Κάθε γράμμα αντιστοιχίζεται με έναν αριθμό. Το κλειδί k είναι ένας αριθμός από το 1 ως το 25. Ο ορισμός του αλγορίθμου εκφράζεται ως εξής:

$$Let\ P = C = K = Z_{26}, \qquad x \in P, \qquad y \in C, \qquad k \in K$$
$$Encryption: \quad e_k(x) = x + k \bmod 26$$
$$Decryption: \quad d_k(y) = y - k \bmod 26$$

Κατά την κρυπτογράφηση το κλειδί k προστίθεται στον αριθμό κάθε γράμματος του μηνύματος και υπολογίζεται το υπόλοιπο της διαίρεσης του αθροίσματος με το πλήθος των γραμμάτων της αλφαβήτου (για το λατινικό αλφάβητο έχουμε modulo 26 αφού το πλήθος των γραμμάτων είναι 26). Έτσι, για παράδειγμα, εάν το κλειδί k είναι το 3, τότε το μήνυμα “SECURE“ κρυπτογραφείται σε “VHFXUH”. Πιο συγκεκριμένα για το γράμμα “S” προκύπτει το “V” γιατί το “S” έχει αντίστοιχο αριθμό το 18 και κρυπτογραφείται με τον υπολογισμό 18 + 3 = 21 οπότε 21mod26 = 21, που αντιστοιχεί στο γράμμα “V”

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Ο κρυπτογραφικός αλγόριθμος του Καίσαρα είναι ουσιαστικά ένας συνηθισμένος τύπος κρυπτογραφικού αλγορίθμου ροής. Είναι πολύ εύκολο να σπάσει (κρυπταναλυθεί). Η μέθοδος της εκτεταμένης αναζήτησης θα δώσει αποτέλεσμα γιατί υπάρχουν μόνο 25 διαφορετικά κλειδιά, καθώς το σύνολο των κλειδιών είναι ίσο με το σύνολο των όλων των δυνατών μεταθέσεων των γραμμάτων, ίσο δηλαδή με το πλήθος των γραμμάτων του λατινικού αλφαβήτου.
Γενικότερα, παρά τον μεγάλο αριθμό κλειδιών, πράγμα που αποκλείει μια απλά επίθεση εξαντλητικής αναζήτησης (exhaustive search attack) , ένας κρυπτογραφικός αλγόριθμος απλής αντικατάστασης είναι εύκολο να σπάσει. Ένας λόγος είναι ότι σε κάθε φυσική γλώσσα τα γράμματα της αλφαβήτου παρουσιάζουν πολύ διαφορετικές συχνότητες εμφάνισης στις διάφορες προτάσεις π.χ. στα Ελληνικά το γράμμα Α είναι πολύ πιο συχνά επαναλαμβανόμενο σε σχέση με γράμματα όπως Z και το Ψ. Αυτή η πληροφορία συνδυαζόμενη με συχνότητες εμφάνισης συνδυασμών δύο ή τριών γραμμάτων μπορεί να χρησιμοποιηθεί για να εξαχθούν αντιστοιχίες μεταξύ του αρχικού και του κρυπτογραφημένου κειμένου, από τις οποίες είναι δυνατόν στη συνέχεια να προκύψει η τιμή του κλειδιού.

Αλγόριθμος Vigenere [6]

Ένας άλλος παρόμοιος κρυπτογραφικός αλγόριθμος είναι ο αλγόριθμος Vigenere. Σε αυτόν τα γράμματα αντιστοιχίζονται πάλι με τους αριθμούς από το 0 ως το 25, όπως ακριβώς και με τον κρυπτογραφικό αλγόριθμο του Καίσαρα. Όμως το μυστικό κλειδί, τώρα, δεν είναι ένας αριθμός αλλά μια μικρή ακολουθία γραμμάτων, όπως για παράδειγμα μια λέξη.
Κατά την κρυπτογράφηση προστίθεται το αριθμητικό ισοδύναμο κάθε γράμματος του αρχικού κειμένου με το αριθμητικό ισοδύναμο ενός γράμματος του κλειδιού. Επειδή συνήθως το μήκος του αρχικού κειμένου είναι μεγαλύτερο από το μήκος του κλειδιού, τα γράμματα του κλειδιού ανακυκλώνονται και επαναλαμβάνεται η χρήση τους όσο χρειάζεται.
Αξίζει να σημειώσουμε ότι ο κρυπτογραφικός αλγόριθμος του Καίσαρα είναι μια ειδική περίπτωση του κρυπτογραφικού αλγορίθμου Vigenere για την περίπτωση που το μήκος της λέξης του κλειδιού είναι ίσο με 1.
Αυτός ο αλγόριθμος ανήκει στην κατηγορία των αποκαλούμενων Κρυπτογραφικών Αλγορίθμων Πολυαλφαβητκής Αντικατάστασης (polyalphabetic substitution ciphers).
Ο κρυπτογραφικός αλγόριθμος Vigenere είναι και αυτός μια ειδική μορφή

κρυπτογραφικού αλγορίθμου ροής. Ακριβώς όπως με τον κρυπτογραφικό αλγόριθμο του Καίσαρα, χρησιμοποιεί πρόσθεση με υπολογισμό του modulo 26 αντί για πρόσθεση με υπολογισμό του modulo 2 για να συνδυάσει το αρχικό κείμενο με το κλειδί. Είναι απλά η λέξη-κλειδί, η οποία επαναλαμβάνεται όσο χρειάζεται. Φυσιολογικά ο κρυπτογραφικός αλγόριθμος Vigenere σπάζει εύκολα.

Αλγόριθμος σημειωματάριου μιας χρήσης [6]

Ο κρυπτογραφικός αλγόριθμος του σημειωματάριου μιας χρήσης (The one-time pad cipher) ή αλγόριθμος του Vernam είναι μια ειδική παραλλαγή κρυπτογραφικού αλγορίθμου ροής. Το ψευδοτυχαίο κλειδί αντικαθίσταται από μια τυχαία (μη επαναλαμβανόμενη) ακολουθία δυαδικών ψηφίων (bits) η οποία χρησιμοποιείται μόνο μια φορά (από αυτό προκύπτει και ο χαρακτηρισμός «μιας χρήσης»). Αν χρησιμοποιηθεί σωστά, ο αλγόριθμος αυτός αποδεδειγμένα δεν είναι δυνατόν να σπάσει (unbreakable).
Το μοναδικό πρόβλημα αφόρα τη διαχείριση των κλειδιών. Πριν να καταστεί δυνατή η κρυπτογραφημένη επικοινωνία, τα δυο μέρη (αποστολέας και παραλήπτης) πρέπει να συμφωνήσουν σε τόσο υλικό τυχαίων κλειδιών όσα και τα δεδομένα που θα μεταδοθούν.

Αλγόριθμος ROT13[6]

Ο κρυπτογραφικός αγόριθμος ROT 13 περιλαμβάνεται σε συστήματα UNIX. Κατά την κρυπτογράφηση αντικαθίσταται κάθε γράμμα με την μετάθεσή του 13 θέσεις δεξιά στο αλφάβητο, ενώ κατά την αποκρυπτογράφηση γίνεται η αντίθετη ή η ίδια μετάθεση.

ΠΕΡΙΟΔΟΙ ΚΡΥΠΤΟΓΡΑΦΙΑΣ	ΧΡΟΝΟΛΟΓΙΕΣ	ΑΛΓΟΡΙΘΜΟΙ ΚΡΥΠΤΟΓΡΑΦΗΣΗΣ
1η ΠερίοδοςΚρυπτογραφίας	1900 π.Χ. – 1900 μ.Χ	“Σκυτάλη”,”Καίσαρα”,”Vigenere”
2η ΠερίοδοςΚρυπτογραφίας	1900 μ.Χ. – 1950 μ.Χ	“Enigma”,”SIGABA”,”one-time pad”
3η ΠερίοδοςΚρυπτογραφίας	1950 μ.Χ. – Σήμερα	“ROT13”“DES”, “AES” κ’ άλλοι..

1.2.2. Οι πρώτοι επιστήμονες κρυπτογραφίας [4]

Ο σημαντικότερος εκπρόσωπος των Αράβων κρυπτολόγων είναι ο πανεπιστήμων του αιώνα Αλ Κιντί. έγραψε πάνω από 290 βιβλία Μαθηματικών – Γλωσσολογίας – Αστρολογίας– Ιατρικής και Μουσικής. Η Ευρωπαϊκή κρυπτογραφία έχει τις ρίζες της το μεσαίωνα, που αναπτύχθηκε από τον Πάπα και τις Ιταλικές πόλεις κράτη, αλλά τα περισσότερα συστήματα βασίζονταν στην απλή αντικατάσταση γραμμάτων της αλφαβήτου (όπως στον αλγόριθμο του Καίσαρα). Οι πρώτοι αλγόριθμοι βασίζονταν στην αντικατάσταση των φωνηέντων. Το πρώτο Ευρωπαϊκό εγχειρίδιο κρυπτογραφίας (1379) ήταν μια συλλογή αλγορίθμων από τον Gabriele de Lavinde of Parma, για τον Πάπα. Το 1470 ο Leon Battista Alberti εξέδωσε το "Trattati in cifra", όπου περιγράφεται ο πρώτος δίσκος κρυπτογράφησης (τον οποίο είχε κατασκευάσει το 1460), χρησιμοποιώντας και την έννοια της χρήσης πολλαπλών αλφαβήτων. Επίσης στο βιβλίο αυτό περιέγραφε και τις αρχές της ανάλυσης συχνότητας των γραμμάτων. Ο Φλωρεντινός Λέων Μπατίστα Αλμπέρτι (1404 μΧ.) στην κρυπτανάλυση ήταν ο πρώτος που σκέφθηκε ένα πολυαλφαβητικό σύστημα, δηλ. ένα

σύστημα με περισσότερα από ένα κρυπτογραφικά αλφάβητα, γεγονός που κάνει την κρυπτανάλυση δυσκολότερη αφού δεν διατηρεί τις συχνότητες των γραμμάτων. Επίσης επινόησε και την πρώτη μετά τη σκυτάλη κρυπτογραφική μηχανή, τους λεγόμενους δίσκους του Alberti. Πήρε δύο χάλκινους δίσκους διαφορετικής διαμέτρου, τους έκανε ομόκεντρους και χάραξε ένα αλφάβητο στην περιφέρεια του κάθε δίσκου. Οι δύο δίσκοι μπορούν να περιστρέφονται ανεξάρτητα Για περισσότερη δυσκολία ένα μέρος του μηνύματος κρυπτογραφείται σε μια θέση του εσωτερικού δίσκου και ένα άλλο μέρος σε άλλη θέση του δίσκου (πολυαλφαβητικό).Το 1499 μΧ. ο Γερμανός abbas Johanes Trithemius έγραψε τη Στενογραφία, ένα βιβλίο επικοινωνίας με τα πνεύματα.

Α. Προ του 20ού αιώνα [5]

Al-Khalil ibn Ahmad Al-Farahidi: έγραψε σε ένα (τώρα χαμένο) βιβλίο στο σύστημα κρυπτογραφία με τον τίτλο το «βιβλίο των κρυπτογραφικών μηνυμάτων».
Al-Kindi, πολυμαθής της Αραβικής του 9ου αιώνα και δημιουργός της ανάλυσης συχνότητας.
Ibn Wahshiyya: δημοσίευσε διάφορα cipher αλφάβητα που χρησιμοποιήθηκαν για να κρυπτογραφήσουν τους μαγικούς τύπους.
Leone Battista Alberti : καθολική μεγαλοφυία, εφευρέτης της πολυαλφαβητικής αντικατάστασης (πιο συγκεκριμένα, cipher Alberti), και τι μπορεί να ήταν η πρώτη μηχανική βοήθεια κρυπτογράφησης.
Étienne Bazeries : Γάλλος, στρατιωτικός, που θεωρούνταν ένα από τα μέγιστα φυσικά cryptanalysts. Ο πιό γνωστός για την ανάπτυξη του «κυλίνδρου Bazeries»
Julius Caesar : Ρωμαίος στρατηγός / πολιτικός, ο Καίσαρας έχει cipher με το όνομα του, χρησιμοποίησε πολύπλοκα συστήματα κρυπτογράφησης, για τα οποία έγραψε ένα βιβλίο ο Valerius_Probus, που θεωρείται ότι έχει καλύψει τη χρήση στρατιωτικού συστήματος κρυπτογραφίας του με κάποιες λεπτομέρειες.

Β. Πρώτος Παγκόσμιος Πόλεμος και Δεύτερος Παγκόσμιος Πόλεμος κρυπτογράφοι του πολέμου [5]

Ludomir Danielewicz, Biuro Szyfrow : βοήθησαν για την κατασκευή των αντιγράφων μηχανής Enigma για να σπάσουν τα ciphers
William F. Friedman : εισήγαγε στατιστικές μεθόδους στην κρυπτογραφία
Nigel de Grey : έπαιξε σημαντικό ρόλο στην αποκρυπτογράφηση του τηλεγραφήματος Zimmermann κατά τη διάρκεια του Α Παγκοσμίου Πολέμου
Frank Rowlett : ηγέτης της ομάδας που έσπασε την πορφύρα
Henryk Zygalski : βοήθησε να σπάσει γερμανικά Enigma ciphers

1.3 Οι πρώτες εφαρμογές που εφαρμόστηκε η κρυπτογραφία. [1]

Η εξέλιξη της χρησιμοποίησης της κρυπτογραφίας ολοένα αυξάνεται καθιστώντας πλέον αξιόπιστη την μεταφορά της πληροφορίας για διάφορους λειτουργικούς σκοπούς

1. Ασφάλεια συναλλαγών σε τράπεζες δίκτυα - ATM
2. Κινητή τηλεφωνία (ΤΕΤΡΑ-ΤΕΤΡΑΠΟΛ-GSM)

3. Σταθερή τηλεφωνία (cryptophones)
4. Διασφάλιση Εταιρικών πληροφοριών
5. Στρατιωτικά δίκτυα (Τακτικά συστήματα επικοινωνιών μάχης)
6. Διπλωματικά δίκτυα (Τηλεγραφήματα)
7. Ηλεκτρονικές επιχειρήσεις (πιστωτικές κάρτες, πληρωμές)
8. Ηλεκτρονική ψηφοφορία
9. Ηλεκτρονική δημοπρασία
10. Ηλεκτρονικό γραμματοκιβώτιο
11. Συστήματα συναγερμών
12. Συστήματα βιομετρικής αναγνώρισης
13. Έξυπνες κάρτες
14. Ιδιωτικά δίκτυα (VPN)
15. World Wide Web
16. Δορυφορικές εφαρμογές (δορυφορική τηλεόραση)
17. Ασύρματα δίκτυα (Hipperlan, bluetooth, 802.11x)
18. Συστήματα ιατρικών δεδομένων και άλλων βάσεων δεδομένων
19. Τηλεσυνδιάσκεψη - Τηλεφωνία μέσω διαδικτύου (VOIP)

1.4 Συμπέρασμα

Η προπτική να χρησιμοποιηθεί το Internet ως μέσο για τις ηλεκτρονικές συναλλαγές και την γενικότερη ανθρώπινη επικοινωνία οδήγησε στην τεράστια ανάπτυξη του διαδικτύου που παρακολουθούμε τα τελευταία χρόνια. Όμως, δεν προβλέφθηκε από τους σχεδιαστές του ο ορισμός μηχανισμών ασφάλειας για την προστασία των πληροφοριών που διακινούνται, αφού το διαδίκτυο σχεδιάστηκε για ενδοπανεπιστημιακή επικοινωνία και όχι γι' αυτό που παρακολουθούμε και χρησιμοποιούμε σε παγκόσμια κλίμακα σήμερα. Για την προστασία των πληροφοριών που μεταδίδονται μέσω δικτύων ψηφιακών επικοινωνιών χρησιμοποιούνται σήμερα ευρέως συστήματα κρυπτογραφίας δημοσίου και μυστικού κλειδιού που θα δούμε στο επόμενο κεφάλαιο.

Αναφορές Κεφάλαιο Ι

[1] http://el.wikipedia.org/
[2] Σμυρνάκης Αθανάσιος Πτυχιακή εργασία με τίτλο "Κρυπτογραφία", 2008
[3] Φλωκατούλα Δώρα "Συμμετρικοί Αλγόριθμοι Κρυπτογράφησης Δεδομένων – Οι περιπτώσεις των αλγορίθμων DES και TDEA",2012
[4] Ελένη Γολέμη Μεταπτυχιακή Εργασία "Κρυπτογραφία & Εξόρυξη Δεδομένων" ,2010
[**5**] Φιολιτάκη Αντωνίου Πτυχιακή Εργασία "Μελέτη Αλγορίθμων Κρυπτογράφησης και Υλοποίηση του DES και Triple-DES σε FPGA με τη χρήση της Γλώσσας Περιγραφής Υλικού VHDL",2005
[6] Δημόπουλος Δημήτριος Διπλωματική εργασία "Υπολογιστική Υλοποίηση Πρωτοκόλλου DES με εφαρμογές στην Κρυπτογράφηση Κειμένων ",2008

ΚΕΦΑΛΑΙΟ ΙΙ

Αλγόριθμοι Κρυπτογράφησης [2]

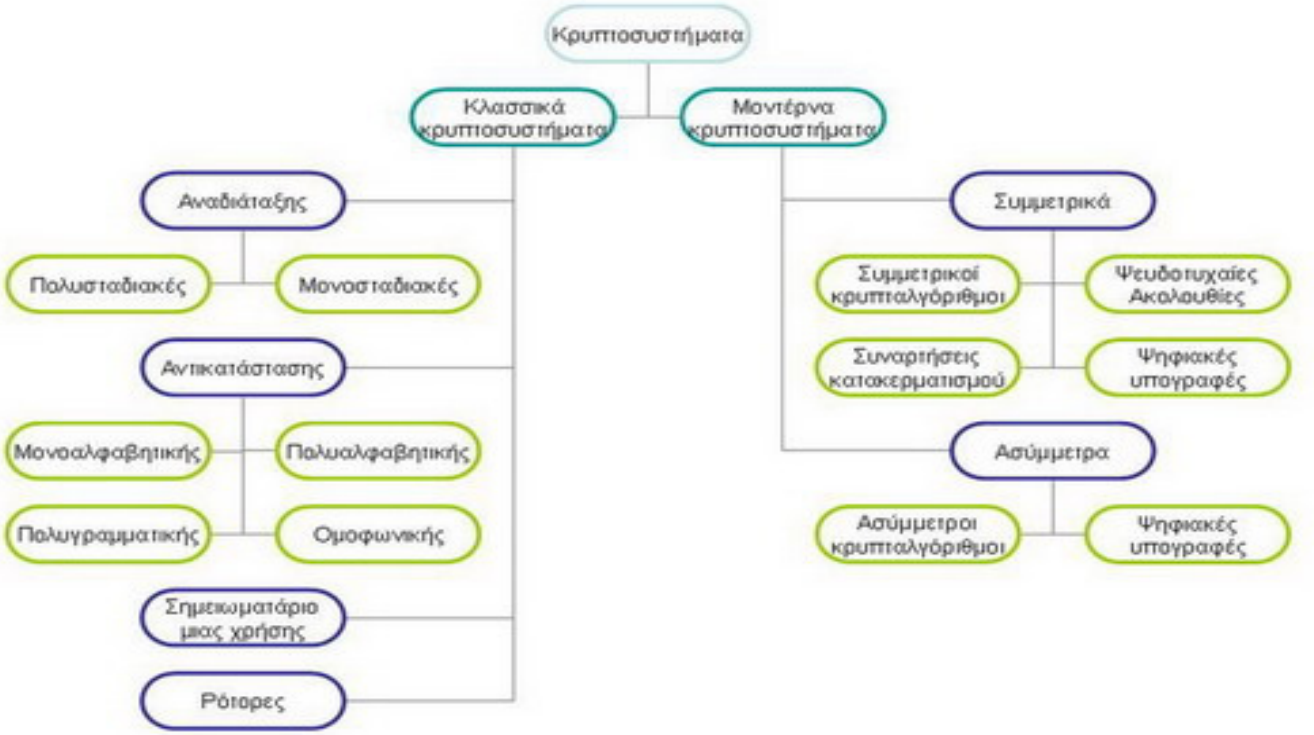
Η τεράστια ανάπτυξη των δικτύων υπολογιστών και η επικοινωνία πληροφοριών κάθε μορφής έφερε ένα τεράστιο πρόβλημα στην επιφάνεια, την ανάγκη για προστασία αυτής της πληροφορίας. Σύμφωνα με το ISO 74982, οι πέντε βασικές υπηρεσίες ασφαλείας που μπορούν να υποστηριχθούν σε ανοικτά συστήματα είναι:

1. η γνησιότητα χρηστών και πληροφοριών (authentication)
2. ο έλεγχος πρόσβασης (access control)
3. η εμπιστευτικότητα (confidentiality)
4. η ακεραιότητα των δεδομένων (data integrity)
5. η μη δυνατότητα αποκήρυξης γεγονότων που έχουν συμβεί (non-repudiation)

Αυτές οι υπηρεσίες παρέχονται από διάφορους μηχανισμούς. Οι υπηρεσίες (1), (3), (4) και (5) μπορούν να υποστηριχθούν από κρυπτογραφικές μεθόδους ενώ η υπηρεσία (2) προϋποθέτει προσθετικά και την χρήση φυσικών μεθόδων.

Κρυπτογράφηση (encryption) είναι ο μετασχηματισμός των δεδομένων σε μορφή που δεν μπορεί να διαβαστεί από κανένα παρά μόνο από αυτόν που διαθέτει ένα κατάλληλο κλειδί. Υπάρχουν δύο μεγάλες οικογένειες αλγόριθμων κρυπτογράφησης, οι συμμετρικοί αλγόριθμοι (ή αλγόριθμοι μυστικού κλειδιού) και οι ασύμμετροι (ή αλγόριθμοι δημόσιου κλειδιού).

Τα κρυπτοσυστήματα χωρίζονται σε 2 μεγάλες κατηγορίες τα **Κλασσικά Κρυπτοσυστήματα και τα Μοντέρνα Κρυπτοσυστήματα** (Συμμετρικά κρυπτοσυστήματα και Ασύμμετρα κρυπτοσυστήματα) **ΣΧΗΜΑ 1**[1]



ΣΧΗΜΑ 1

2.1 Κυριότεροι αλγόριθμοι ανα κατηγορία.[1]

Οι **συμμετρικοί κρυπτογραφικοί** αλγόριθμοι μπορούν να χωριστούν σε δύο διαφορετικές κατηγορίες με βάση τον τρόπο κρυπτογράφησης των μηνυμάτων:

- *Δέσμης (Block Ciphers)*, οι οποίοι χωρίζουν το μήνυμα σε κομμάτια και κρυπτογραφούν κάθε ένα από τα κομμάτια αυτά χωριστά.
- *Ροής (Stream Ciphers)*, οι οποίοι κρυπτογραφούν μία ροή μηνύματος (stream) χωρίς να τη διαχωρίζουν σε τμήματα

Συμμετρικοί Κρυπταλγόριθμοι Τμήματος (Block Ciphers) :

Data Encryption Standard,3-Way ,Blowfish, CAST ,CMEA ,Triple-DES,DEAL FEAL , GOST,IDEA ,LOKI ,Lucifer,MacGuffin,Twofish

MARS , MISTY ,MMB ,NewDES ,RC2, RC5 , RC6 REDOC ,Rijndael ,Safer ,Serpent,SQUARE, Skipjack ,Tiny Encryption Algorithm

Συμμετρικοί Κρυπταλγόριθμοι ροής (Stream Ciphers) :

ORYX ,RC4 , SEAL

Οι κυριότεροι **συμμετρικοί μπλοκ αλγόριθμοι** είναι ο DES (Data Encryption Standard), ο Triple-DES και ο AES.

2.1.1 Συμμετρικοί αλγόριθμοι [1] [2]

Στους συμμετρικούς αλγόριθμους το κλειδί κρυπτογράφησης μπορεί να υπολογιστεί από το κλειδί που χρησιμοποιείται για την αποκρυπτογράφηση και το ανάποδο. Μάλιστα στις περισσότερες περιπτώσεις τα κλειδιά κρυπτογράφησης και αποκρυπτογράφησης είναι τα ίδια. Αυτοί οι αλγόριθμοι χρειάζονται την συμφωνία μεταξύ του αποστολέα και του παραλήπτη για το κλειδί που θα χρησιμοποιηθεί, για να μπορέσουν να επικοινωνήσουν με ασφάλεια. Η ασφάλεια των αλγόριθμων βασίζεται στην μυστικότητα αυτού του κλειδιού. Για όσο καιρό επιθυμούμε η επικοινωνία να παραμείνει μυστική, για τον ίδιο καιρό πρέπει και το κλειδί να παραμείνει μυστικό.

Τα στάδια της επικοινωνίας του σχήματος **2** είναι τα ακόλουθα:

1. Ο Κώστας ή η Βασιλική αποφασίζει για ένα κλειδί το οποίο το επιλέγει τυχαία μέσα από τον κλειδοχώρο.
2. Η Βασιλική αποστέλλει το κλειδί στον Κώστα μέσα από ένα ασφαλές κανάλι.
3. Ο Κώστας δημιουργεί ένα μήνυμα όπου τα σύμβολα m ανήκουν στον χώρο των μηνυμάτων.
4. Κρυπτογραφεί το μήνυμα με το κλειδί που έλαβε από τη Βασιλική και η παραγόμενη κρυπτοσυμβολοσειρά αποστέλεται.
5. Η Βασιλική λαμβάνει την κρυπτοσυμβολοσειρά και στη συνέχεια με το ίδιο κλειδί την αποκρυπτογραφεί και η έξοδος που παράγεται είναι το μήνυμα.

ΣΧΗΜΑ 2:Μοντέλο Συμμετρικού Κρυπτοσυστήματος

Οι αλγόριθμοι συμμετρικού κλειδιού χωρίζονται σε δύο κατηγορίες ανάλογα με την μεταχείριση της πληροφορίας προς κρυπτογράφηση :

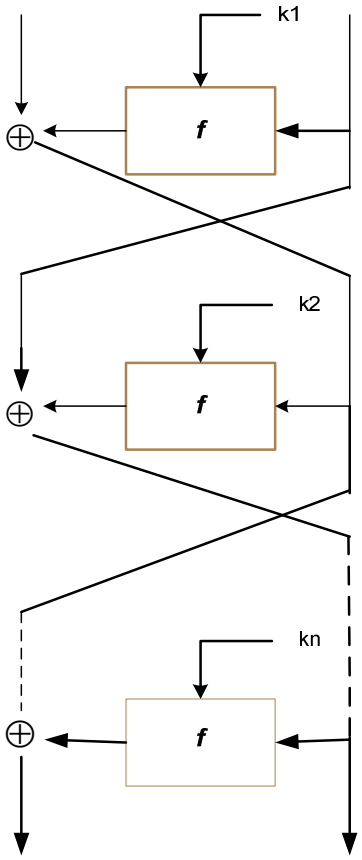
- τους αλγόριθμους που χειρίζονται μπλοκ δυαδικών ψηφίων (block ciphers)
- τους αλγόριθμους που χειρίζονται ξεχωριστά κάθε δυαδικό ψηφίο της πληροφορίας (stream ciphers)

2.1.1.1 Συμμετρικοί μπλοκ αλγόριθμοι (block ciphers) [3]

Ένας συμμετρικός μπλοκ αλγόριθμος χωρίζει τα μη κρυπτογραφημένα δεδομένα (plaintext) σε μπλοκ δυαδικών ψηφίων συγκεκριμένου μήκους t (block size) και κρυπτογραφεί ξεχωριστά το κάθε ένα από αυτά με τη χρήση ενός μυστικού κλειδιού. Το αποτέλεσμα είναι ένα μπλοκ κρυπτογραφημένων δεδομένων (ciphertext) το οποίο έχει το ίδιο μήκος με το αρχικό. Το σύνολο των κρυπτογραφημένων μπλοκ αποτελεί το κρυπτογραφημένο μήνυμα.
Το μεγαλύτερο ποσοστό των αλγόριθμων αυτών είναι επαναλαμβανόμενοι καθώς χειρίζονται αλυσιδωτά τα μπλοκ δεδομένων. Ο αριθμός των επαναλήψεων ενός τέτοιου αλγόριθμου εξαρτάται από το επίπεδο της ασφάλειας που θέλει κάποιος να επιτύχει. Όπως είναι φυσικό ένας αυξημένος αριθμός επαναλήψεων αυξάνει σημαντικά την ασφάλεια αλλά έχει αρνητικές συνέπειες στην απόδοση του καθώς αυξάνεται ο υπολογιστικός χρόνος. Επίσης πολύ σημαντικό στοιχείο είναι και το μέγεθος των μπλοκ δεδομένων (block size). Οι περισσότεροι από τους αλγόριθμους χρησιμοποιούν μεγέθη πάνω από 8 bytes (64 bits). Ένας αλγόριθμος του οποίου το μέγεθος μπλοκ (block size) είναι πολύ μικρό είναι σίγουρο ότι είναι ευάλωτος σε επιθέσεις βασισμένες σε στατιστική ανάλυση (ανάλυση της συχνότητας εμφάνισης συγκεκριμένων συστοιχιών bits (bit patterns) μέσα στα δεδομένα). Εντούτοις το να διαλέξει κάποιος ένα μεγάλο μέγεθος μπλοκ αυξάνει σημαντικά τον υπολογιστικό φόρτο του αλγόριθμου. Οι συμμετρικοί αλγόριθμοι είναι πολύ πιο γρήγοροι, εφαρμοσμένοι είτε σε υλικό είτε σε λογισμικό. από τους ασύμμετρους αλγόριθμους. Ως εκ τούτου οι συμμετρικοί αλγόριθμοι χρησιμοποιούνται για την κρυπτογράφηση του κυρίου μέρους των δεδομένων, ενώ οι αλγόριθμοι δημόσιου κλειδιού βρίσκουν κατάλληλη εφαρμογή σε πρωτόκολλα ανταλλαγής κλειδιών και ψηφιακών υπογραφών.

2.1.1.1.1 Αλγόριθμος Data Encryption Standard (DES) [3]

Ο αλγόριθμος DES είναι ο πιο χαρακτηριστικός συμμετρικός μπλοκ αλγόριθμος και έχει εξελιχθεί τα τελευταία 20 χρόνια. Χρησιμοποιεί μπλοκ των 64 bits και κλειδί των 56 bits. Η είσοδος (input) στον αλγόριθμο είναι μπλοκ των 64 bits μη κρυπτογραφημένου κειμένου (plaintext), και μετά από 16 επαναλήψεις (συνδιασμός μεταθέσεων και αντικαταστάσεων) δίνει έξοδο (output) ένα μπλοκ ίδιου μεγέθους με κρυπτογραφημένα δεδομένα (ciphertext). Αν το μέγεθος του μη κρυπτογραφημένου κειμένου (plaintext) δεν είναι ακέραιο πολλαπλάσιο του μεγέθους του μπλοκ (64 bits) τότε χρησιμοποιείται μια διαδικασία συμπλήρωσης (padding) με bits ώστε να προκύψει plaintext πολλαπλάσιου μεγέθους των 64 bits. Οι πιο γνωστοί μέθοδοι συμπλήρωσης είναι ο PKCS #5 , ο PKCS#7 , ο ISO10126-2Padding και ο X9.23Padding.
Η λειτουργία του DES στηρίζεται στους λεγόμενους αλγόριθμους του Feistel , οι οποίοι ονομάζονται και τύπου DES(**ΣΧΗΜΑ 3**)



ΣΧΗΜΑ 3 Feistel Cipher

Μια παραλλαγή του DES είναι ο Triple-DES αλγόριθμος, ο οποίος χρησιμοποιεί το ίδιο μέγεθος μπλοκ με τον DES (64 bits) εφαρμόζοντας τον DES τρεις φορές διαδοχικά , με τρία διαφορετικά κλειδιά των 56 bits (K1 , K2 , K3). Δοθέντος ενός αρχικού μηνύματος προς κρυπτογράφηση, το πρώτο κλειδί χρησιμοποιείται από τον DES για την κρυπτογράφηση του μηνύματος. Το δεύτερο κλειδί χρησιμοποιείται για να αποκρυπτογραφήσει το κρυπτογραφημένο με το πρώτο κλειδί μήνυμα. Επειδή όμως το δεύτερο κλειδί δεν είναι το σωστό κλειδί για την αποκρυπτογράφηση του μηνύματος, το μόνο που επιτυγχάνεται με αυτή τη διαδικασία είναι να μπερδεύεται ακόμα περισσότερο το ήδη κρυπτογραφημένο μήνυμα. Τελικά το μήνυμα ξανακρυπτογραφείται με το τρίτο κλειδί και έτσι προκύπτει το τελικό κρυπτογραφημένο μήνυμα. Αυτή λοιπόν η διαδικασία τριών βημάτων αποκαλείται triple-DES (TDES ή TDEA). Η αποκρυπτογράφηση γίνεται με την αντίστροφη διαδικασία, εφαρμόζοντας δηλαδή πρώτα το κλειδί K3, μετά το K2 και τέλος το K1. Συνεπώς η κρυπτογραφική δύναμη του Triple-DES είναι τριπλάσια από αυτή του DES(56*3=168 bits).
Ο αλγόριθμος Triple-DES ανήκει στους αποδεχόμενους αλγόριθμους της κυβέρνησης των Η.Π.Α. (US Approved Algorithms List) , ενώ ο DES αφαιρέθηκε από την ανωτέρω λίστα το 2005.

2.1.1.1.2 Αλγόριθμος Advanced Encryption Standard (AES) [3] [12]

Σύμφωνα με όσα προαναφέρθηκαν, αν η ασφάλεια αποτελούσε το μοναδικό κριτήριο επιλογής του αλγορίθμου, τότε ο TDES θα ήταν μία εξαιρετικά κατάλληλη επιλογή για έναν τυποποιημένο αλγόριθμο κρυπτογράφησης για τα επόμενα χρόνια. Όμως, κύριο μειονέκτημα του TDES αποτελεί το γεγονός ότι ο αλγόριθμος είναι σχετικά αργός σε υλοποιήσεις με χρήση λογισμικού. Το σύστημα DES σχεδιάστηκε για υλοποίηση με χρήση υλικού τη δεκαετία του '70 και δε φαίνεται να παράγει αποδοτικό κώδικα λογισμικού. Ο TDES, που περιλαμβάνει τρεις φορές περισσότερους γύρους από τον DES, είναι προφανώς πολύ βραδύτερος. Επιπλέον μειονέκτημα αποτελεί η απαίτηση των DES και TDES για χρησιμοποίηση τμημάτων μεγέθους 64-bit. Για γενικότερους λόγους αποδοτικότητας και ασφάλειας, είναι επιθυμητό μεγαλύτερο μέγεθος τμήματος. Κατά συνέπεια ο TDES δε μπορεί να θεωρηθεί αποτελεσματικός προϊόντος του χρόνου. Για την αντιμετώπιση των προβλημάτων αυτών, ήδη από το 1997 το NIST εξέδωσε μία πρόσκληση υποβολής προτάσεων για νέο Προηγμένο Πρότυπο Κρυπτογράφησης (Advanced Encryption Standard – AES), διάδοχο του DES και προσδιόρισε ότι το AES θα πρέπει να αποτελεί κωδικοποιητή τμημάτων με συμμετρικό σύστημα κρυπτογράφησης, μήκους τμήματος 128 bit και να υποστηρίζει κλειδιά μήκους 128-bit, 192-bit και 256-bit. Τα κριτήρια συγκριτικής αξιολόγησης των υποψηφίων αλγορίθμων εντάχθηκαν σε τρεις κατηγορίες:

• **Στην ασφάλεια των αλγορίθμων**: τα κριτήρια που εντάσσονται σε αυτήν την κατηγορία περιλάμβαναν τη ρωμαλεότητα των αλγορίθμων σε κρυπταναλυτικές επιθέσεις, την ορθότητα του μαθηματικού τους φορμαλισμού, τη σχετική συγκριτική ασφάλεια του αλγορίθμου σε σχέση με τους υπόλοιπους υποψήφιους αλγορίθμους και την τυχαιότητα της συμπεριφοράς της εξόδου. Σε γενικές γραμμές οι αλγόριθμοι έπρεπε να έχουν χαρακτηριστικά ασφάλειας τουλάχιστον ισοδύναμα με του αλγορίθμου TDES, αλλά να χαρακτηρίζονται ταυτόχρονα από σημαντικά βελτιωμένη αποδοτικότητα.

• **Στο κόστος**: τα κριτήρια που εντάσσονταν σε αυτή την κατηγορία αναφέρονταν στις απαιτήσεις μνήμης και υπολογιστικής ισχύος του αλγορίθμου, καθώς και στις απαιτήσεις περί προστασίας δικαιωμάτων πνευματικής ιδιοκτησίας και πατέντες ώστε το υπό ανάπτυξη πρότυπο να μπορεί να είναι αξιοποιήσιμο σε διεθνή κλίμακα.

• **Στην απλότητα**: τα κριτήρια που εντάσσονταν σε αυτήν την κατηγορία περιλάμβαναν την απλότητα, την ευελιξία - δηλαδή τη δυνατότητα του αλγορίθμου να χειρίζεται μεγέθη μυστικών κλειδιών και τμημάτων μη κρυπτογραφημένου κειμένου μεγαλύτερα από τα ελάχιστα τεθέντα - τη δυνατότητα υλοποίησης σε διάφορα περιβάλλοντα όπως λογισμικό, υλικό, υλικολογισμικό (firmware), καθώς και την παροχή συμπληρωματικών κρυπτογραφικών λειτουργιών

Σε έναν πρώτο κύκλο αξιολόγησης έγιναν αποδεκτοί δέκα πέντε προτεινόμενοι αλγόριθμοι και σε δεύτερο κύκλο ο αριθμός των αποδεκτών αλγορίθμων μειώθηκε σε πέντε. Οι αλγόριθμοι αυτοί ήταν οι MARS, RC6, Rijndael, Serpent, Twofish. Τελικά επιλέχθηκε επισήμως ως AES ο αλγόριθμος Rijndael, ο οποίος είχε υποβληθεί από τους Βέλγους κρυπτογράφους J. Daemen και V. Rijmen και έλαβε την οριστική του σχεδιαστική μορφή στο τέλος του καλοκαιριού του 2001.

Ο αλγόριθμος Rijndael, που έχει υιοθετηθεί πλέον ως ο αλγόριθμος AES, χαρακτηρίζεται από απλότητα, ευελιξία, ρωμαλεότητα σε όλες τις γνωστές κρυπταναλυτικές επιθέσεις και υψηλή ταχύτητα λειτουργίας. Σχεδιαστικά, ο αλγόριθμος Rijndael δεν ακολουθεί την κλασική δομή Feistel, αλλά κάθε κύκλος λειτουργίας περιλαμβάνει τρεις όμοιους μετασχηματισμούς, με όρους ισότιμης αντιμετώπισης κάθε ξεχωριστού bit, γνωστούς ως επίπεδα (layers):

- Το επίπεδο γραμμικής ανάμιξης (linear mixing layer) επιτυγχάνει υψηλή διάχυση σε πολλαπλούς κύκλους
- Το μη γραμμικό επίπεδο (non-linear layer) αφορά στην παράλληλη εφαρμογή s-boxes τα οποία εμφανίζουν εξαιρετικές μη γραμμικές ιδιότητες για το ενδεχόμενο χειρότερης περίπτωσης
- Το επίπεδο πρόσθεσης κλειδιού (key addition layer) αφορά στη συσχέτιση του ενδιάμεσα προκύπτοντος αποτελέσματος με το υποκλειδί του κύκλου, με την πράξη XOR

2.1.1.1.3 Αλγόριθμος Blowfish[12]

Ο αλγόριθμος Blowfish αναπτύχθηκε το 1993 από τον επιφανή κρυπτογράφο B. Schneier και καθιερώθηκε ως μία από τις δημοφιλέστερες εναλλακτικές λύσεις του DES. Ο Blowfish σχεδιάστηκε ώστε να είναι εύκολος στην υλοποίηση και να παρουσιάζει μεγάλη ταχύτητα εκτέλεσης. Πρόκειται για ένα συνεπτυγμένο αλγόριθμο που μπορεί να εκτελεστεί σε μνήμη μικρότερη από 5K. Ενδιαφέρον χαρακτηριστικό γνώρισμα του Blowfish αποτελεί το μήκος κλειδιού, το οποίο είναι μεταβλητό, μπορεί να λάβει τιμές έως 448-bit, αν και πρακτικά χρησιμοποιούνται κλειδιά των 128-bit. Ο Blowfish χρησιμοποιεί 16 γύρους.

Όπως ο αλγόριθμος DES, ο αλγόριθμος Blowfish χρησιμοποιεί S-boxes, XOR, καθώς και δυαδική πρόσθεση. Αντίθετα από τον DES που χρησιμοποιεί σταθερά S-boxes, ο Blowfish χρησιμοποιεί δυναμικά S-boxes που παράγονται ως συνάρτηση του κλειδιού. Στον Blowfish, τα υποκλειδιά και τα S-boxes παράγονται από την επανειλημμένη εφαρμογή του ίδιου του αλγορίθμου Blowfish στο κλειδί. Συνολικά απαιτούνται 521 εκτελέσεις του αλγορίθμου κρυπτογράφησης Blowfish για την παραγωγή των υποκλειδιών και των S-boxes. Απόρροια των χαρακτηριστικών αυτώνείναι το συμπέρασμα ότι ο Blowfish δεν είναι κατάλληλος για εφαρμογές στις οποίες το μυστικό κλειδί αλλάζει συχνά. Ο Blowfish περιλαμβάνεται στους καλύτερους συμβατικούς αλγορίθμους κρυπτογράφησης που έχουν εφαρμοστεί, αφού τα υποκλειδιά και τα S-boxes παράγονται από διαδικασία επανειλημμένων εφαρμογών του Blowfish στον εαυτό του. Οι επαναλήψεις αυτές τροποποιούν πλήρως τα δυαδικά ψηφία και καθιστούν την κρυπτανάλυση εξαιρετικά δύσκολη. Οι μέχρι σήμερα δημοσιεύσεις των προσπαθειών για κρυπτανάλυση του Blowfish δεν αναφέρουν πρακτικές αδυναμίες. Ο Blowfish χρησιμοποιείται, επίσης, σε διάφορες εμπορικές εφαρμογές.

2.1.1.1.4 International Data Encryption Algorithm (IDEA) [12]

Ο αλγόριθμος International Data Encryption Algorithm – IDEA αποτελεί συμμετρικό κωδικοποιητή τμημάτων, που αναπτύχθηκε από τους X. Lai και J. Massey, στο Swiss Federal Institute of Technology, το 1991. Ο IDEA χρησιμοποιεί κλειδί μήκους 128-bit και διαφέρει από τον DES τόσο στη συνάρτηση F, όσο και στη συνάρτηση παραγωγής των υποκλειδιών. Για τη συνάρτηση F, ο IDEA δε χρησιμοποιεί S-boxes, αλλά στηρίζεται σε τρεις διαφορετικές μαθηματικές λειτουργίες: τη δυαδική πράξη XOR, τη δυαδική πρόσθεση ακεραίων των 16-bit και το δυαδικό πολλαπλασιασμό ακεραίων των 16-bit.

Οι συναρτήσεις συνδυάζονται με τρόπο ώστε να αναπτυχθεί ένας πολύπλοκος μετασχηματισμός που αναλύεται δύσκολα, ώστε να καθίσταται πολύ δύσκολη η διαδικασία κρυπτανάλυσης. Ο αλγόριθμος παραγωγής δευτερευόντων κλειδιών βασίζεται στη χρήση κυκλικών μετατοπίσεων, οι οποίες χρησιμοποιούνται με πολύπλοκο τρόπο για να παραχθούν συνολικά έξι δευτερεύοντα κλειδιά, για καθέναν από τους οκτώ γύρους του IDEA.

Ο IDEA ήταν ένας από τους προτεινόμενους 128-bit αντικαταστάτες του DES, έχει υποβληθεί σε αξιοσημείωτη διερεύνηση και εμφανίζεται ανθεκτικός σε κρυπταναλυτικές επιθέσεις. Ο IDEA χρησιμοποιείται στο προϊόν λογισμικού PGP, ως μία από τις εναλλακτικές επιλογές, καθώς και σε διάφορα εμπορικά προϊόντα.

2.1.1.1.5 RC5 [12]

Ο RC5 αναπτύχθηκε το 1994 από τον R. Rivest, έναν από τους σχεδιαστές του αλγορίθμου δημοσίου κλειδιού RSA. Ο RC5 προσδιορίζεται στο RFC 2040 και σχεδιάστηκε για να υποστηρίζει τα ακόλουθα χαρακτηριστικά:

• **Κατάλληλο για υλοποίηση σε υλικό ή λογισμικό**: ο RC5 χρησιμοποιεί μόνο βασικές υπολογιστικές λειτουργίες, που συνήθως περιλαμβάνονται στους μικροεπεξεργαστές.

• **Ταχύς**: προκειμένου να επιτευχθεί υψηλή ταχύτητα, ο RC5 είναι ένας απλός αλγόριθμος που βασίζεται στη λέξη (word). Οι βασικές λειτουργίες του στηρίζονται σε πλήρεις λέξεις δεδομένων ανά στιγμή.

• **Προσαρμόσιμος σε επεξεργαστές διαφορετικών μηκών λέξης**: Ο αριθμός των δυαδικών ψηφίων σε μία λέξη αποτελεί παράμετρο του RC5, έτσι ώστε διαφορετικά μήκη λέξης παράγουν διαφορετικούς αλγορίθμους.

• **Μεταβλητό μήκος γύρων**: Ο αριθμός των γύρων αποτελεί δεύτερη παράμετρο του RC5. Αυτή η παράμετρος επιτρέπει την εναλλαγή μεταξύ υψηλότερης ταχύτητας και υψηλότερης ασφάλειας.

• **Μεταβλητό μήκος κλειδιού**: Το μήκος κλειδιού αποτελεί την τρίτη παράμετρο του RC5. Επίσης επιτρέπει την εναλλαγή μεταξύ υψηλότερης ταχύτητας και υψηλότερης ασφάλειας.

• **Απλός**: Η απλή δομή του RC5 υλοποιείται εύκολα και διευκολύνει τον υπολογισμό της ισχύος του αλγορίθμου.

• **Χαμηλή απαίτηση μνήμης**: Η χαμηλή απαίτηση μνήμης καθιστά τον αλγόριθμο RC5 κατάλληλο για αξιοποίηση σε έξυπνες κάρτες και άλλες συσκευές περιορισμένης μνήμης.

• **Υψηλή ασφάλεια**: Ο RC5 προορίζεται για να παρέχει υψηλή ασφάλεια με προσδιορισμό των κατάλληλων παραμέτρων.

• **Περιστροφές εξαρτώμενες από τα δεδομένα**: Ο RC5 ενσωματώνει τις περιστροφές, δηλαδή κυκλικές μετατοπίσεις δυαδικών ψηφίων, των οποίων ο αριθμός είναι στοιχείο εξαρτώμενο από τα δεδομένα. Το γεγονός αυτό ενισχύει τον αλγόριθμο ενάντια στην κρυπτανάλυση.

Ο RC5 χρησιμοποιείται σε διάφορα προϊόντα από την RSA Data Security, Inc.

2.1.1.2 Συμμετρικοί αλγόριθμοι ροής (Stream Ciphers) [3]

Οι stream ciphers είναι αλγόριθμοι συμμετρικής κρυπτογραφίας και είναι σχεδιασμένοι έτσι ώστε να είναι πολύ πιο γρήγοροι από τους block ciphers. Ένώ οι τελευταίοι δουλεύουν πάνω σε μεγάλα μπλοκ δεδομένων , οι stream ciphers συνήθως χρησιμοποιούν για την κρυπτογραφία μόνο ένα bit. Θα μπορούσαμε λοιπόν ότι είναι μια παραλλαγή των block ciphers μόνο που το block size τους είναι του ενός bit.
Όπως γνωρίζουμε το κρυπτογραφημένο αποτέλεσμα (ciphertext) των block ciphers είναι , για συγκεκριμένο plaintext , πάντα το ίδιο , όταν χρησιμοποιηθεί το ίδιο κλειδί. Εντούτοις , με τους stream ciphers , το ciphertext θα εξαρτηθεί από το πότε , μέσα στη διαδικασία της κρυπτογράφησης , θα χειριστεί ο αλγόριθμος το κάθε bit. Το ciphertext, συνεπώς, είναι πάντα διαφορετικό και μπορούμε να πούμε ότι οι stream ciphers έχουν *«μνήμη»*, εξαρτώνται δηλαδή από προηγούμενη κατάσταση τους (state ciphers). Αυτή όμως η διάκριση μεταξύ των block και stream ciphers δεν είναι κατ' ανάγκη οριστική. Αν προσθέσουμε την ιδιότητα της «μνήμης» σε block ciphers (όπως συμβαίνει στον CBC) μπορούμε να έχουμε μία μορφή stream cipher με μεγαλύτερα μεγέθη μπλοκ του ενός bit.
Η λειτουργία ενός stream cipher παρουσιάζει αρκετές διαφοροποιήσεις. Κάθε τέτοιος αλγόριθμος δημιουργεί μία ακολουθία bits που χρησιμεύουν ως κλειδί και ονομάζονται *keystream*. Η κρυπτογράφηση γίνεται εκτελώντας την λογική πράξη αποκλειστικό-ή (XOR) για κάθε bit του μη κρυπτογραφημένου μηνύματος και του keystream. Οι stream ciphers διακρίνονται σε δύο κατηγορίες, ανάλογα με την συμπεριφορά του keystream:

- Synchronous stream ciphers , όταν η δημιουργία του keystream είναι ανεξάρτητη του plaintext και του ciphertext
- Self-synchronizing stream ciphers, όταν το keystream εξαρτάται από τα δεδομένα και την κατάσταση της κρυπτογράφησης αυτών.

Οι περισσότεροι stream cipher που χρησιμοποιούνται ανήκουν στην πρώτη από αυτές τις δύο κατηγορίες και ο σπουδαιότερος από αυτούς είναι ο RC4.

2.1.1.2.1 Αλγόριθμος RC4. [4]

Ο RC4 είναι ένας κωδικοποιητής ροής (stream cipher) με μεταβλητό μήκος κλειδιού, ο οποίος ανήκει στην κατηγορία των συμμετρικών αλγορίθμων. Ένας συμμετρικός αλγόριθμος, είναι αλγόριθμος κρυπτογράφησης ο οποίος χρησιμοποιεί σχετιζόμενα συχνά εντελώς ίδια κλειδιά κρυπτογράφησης και για την κωδικοποίηση του αρχικού μηνύματος και για την αποκωδικοποίηση του κωδικοποιημένου μηνύματος πίσω στο αρχικό. Οι κωδικοποιητές ροής (stream ciphers) κωδικοποιούν κάθε bit του μηνύματος ξεχωριστά, σε αντίθεση με τους κωδικοποιητές τμημάτων (block ciphers) οι οποίοι κωδικοποιούν το μήνυμα ανά τμήματα από bits.

Γενικά Χαρακτηριστικά του Αλγορίθμου RC4.

- Αλγόριθμος κρυπτογράφησης συμμετρικού κλειδιού
- Σχεδιάστηκε από τον Ron Rivest
- Η λειτουργία του ήταν απόρρητη, μέχρι το 1994, όπου κάποιος τον «έσπασε»
- Χρησιμοποιείται σήμερα σε πολλά πρωτόκολλα και standards κρυπτογράφησης (SSL)

Λειτουργία

- Είναι ουσιαστικά μια γεννήτρια ψευδο-τυχαίων αριθμών
- Απλότητα του αλγορίθμου
- Ιδανικός για υλοποιήσεις με λογισμικό
- Χρησιμοποιεί 256 bytes μνήμης και μεταβλητές ακεραίων
- Αποτελείται από δύο στάδια

a) Στάδιο αρχικοποίησης

```
for i = 0 ... 255

{  S[i] = i

}

for i = 0 ... 255

{

j = (j + S[i] + key[i mod key_length]) mod 256

swap (S[i],S[j])

}
```

b) Διαδικασία κρυπτογράφησης / αποκρυπτογράφησης

```
i = 0

j = 0

loop until the entire message is encrypted/decrypted

{

i = (i + 1) mod 256

j = (j + S[i]) mod 256

swap(S[i],S[j])

k = S[(S[i] + S[j]) mod 256]

output the XOR of k with the next byte of input

}
```

Ζητήματα ασφαλείας

- Συνίσταται οι πρώτες τιμές εξόδου της γεννήτριας να απορριφθούν (προτείνεται η απόρριψη των πρώτων 256 εξόδων)
- Η κρυπτανάλυση δεν έχει δώσει ακόμα τα αναμενόμενα.
- Σε θεωρητική βάση, αν είναι γνωστά κάποια gigabytes του μηνύματος / κρυπτογράμματος,
- τότε η διαδικασία κρυπτογράφησης / αποκρυπτογράφησης μπορεί να «σπάσει».
- Στην πράξη, το παραπάνω δε δημιουργεί σοβαρά προβλήματα.

Επιθέσεις

- 2001: Επίθεση Fluhrer, Martin, Shamir

Για οποιοδήποτε από τα κλειδιά που ο αλγόριθμος RC4 χρησιμοποιεί, τα στατιστικά στοιχεία των πρώτων μερικών bytes του ρεύματος κλειδιού είναι σε ανησυχητικό βαθμό μη τυχαία. Ως αποτέλεσμα, αν κανείς επεξεργαστεί έναν μεγάλο αριθμό μηνυμάτων με το ίδιο κλειδί, είναι δυνατόν αυτό τελικά να βρεθεί.

2.1.1.2.2 Αλγόριθμος ORYX [11]

ORYX είναι ένας αλγόριθμος κρυπτογράφησης που χρησιμοποιείται σε κυψελοειδείς επικοινωνίες με σκοπό την προστασία της κυκλοφορίας των δεδομένων. Πρόκειται για ένα stream cipher που σχεδιάστηκε να έχει ένα πολύ ισχυρό 96-bit κλειδί με έναν τρόπο ώστε να μειωθεί η αντοχή σε 32-bits για την εξαγωγή. Ωστόσο, λόγω λαθών η πραγματική δύναμη είναι ένα ασήμαντο 16-bits και κάθε σήμα μπορεί να ραγίσει μετά τις πρώτες 25-27 bytes

Γενικά Χαρακτηριστικά του Αλγορίθμου ORYX

- Σχεδιασμένο για χρήση με κινητά τηλέφωνα
- Για την προστασία του απορρήτου της φωνής / δεδομένων
- Για "data channel", οχι "control channel"
- Έλεγχος καναλιών κρυπτογραφημένα με CMEA
- Πρότυπο που αναπτύχθηκε από Ένωση Βιομηχανίας Τηλεπικοινωνιών (TIA)
- Ελαττώματα ανακαλύφθηκαν το 1997

Λειτουργία

- χρησιμοποιεί 3 καταχωρητές, συμβολιζοντας τους X, A, B
- καθε cipher κατέχει 32 bits
- το κλειδί αποτελεί το πρώτο γέμισμα των καταχωρητών
- επομένως, ORYX έχει 96 bit κλειδί και
- χρησιμοποιεί έναν πίνακα αναζήτησης L
- που ενεργεί ως IV (ή MI) (το L δεν είναι μυστικό)
- δημιουργεί keystream 1 byte/βήμα
- είναι πολύ αδύναμος,το πρόβλημα με ORYX δημιουργεί 1 byte keystream και η εσωτερική κατάσταση είναι πάρα πολύ εκτεθημένη έτσι ενδέχεται να είναι ισχυρότερη εάν παράγονται μόνο από 1 bit/βήμα

2.1.1.2.3 Αλγόριθμος SEAL **[10] [4]**

Στην κρυπτογραφία, SEAL (Software-Optimized Αλγόριθμος κρυπτογράφησης) είναι μια πολύ γρήγορη κρυπτογράφηση ροής βελτιστοποιημένη για τις μηχανές με μέγεθος λέξης 32-bit και την αφθονία της μνήμης RAM. SEAL είναι στην πραγματικότητα μια ψευδοτυχαία οικογένειακή λειτουργία, δεδομένου οτι μπορεί εύκολα να δημιουργήσει αυθαίρετα τμήματα της κλειδορροής χωρίς να χρειάζεται να ξεκινήσει από την αρχή. Αυτό το καθιστά ιδιαίτερα κατάλληλο για εφαρμογές όπως η κρυπτογράφηση σκληρών δίσκων. Η πρώτη έκδοση δημοσιεύθηκε από το Phillip Rogaway και Don Coppersmith το 1994. Η τρέχουσα έκδοση, που δημοσιεύθηκε το 1997, είναι 3,0. SEAL, καλύπτεται από δύο διπλώματα ευρεσιτεχνίας στις Ηνωμένες Πολιτείες, οι οποίες έχουν εκχωρηθεί στην IBM.

Γενικά Χαρακτηριστικά του Αλγορίθμου SEAL

- Software-optimized Encryption Algorithm

- Σχεδιάστηκε από τους Coppersmith, Rogaway το 1993
- Υλοποιήθηκε με τη χρήση λογισμικού
- Σχετικά νέος αλγόριθμος
- Επικρατέστερη η έκδοση 2.0 του αλγορίθμου
- Αποδοτικός όταν εκτελείται σε 32-bit επεξεργαστές

Λειτουργία

- Ο SEAL είναι μια ψευδο-τυχαία συνάρτηση αυξανόμενου μήκους, η οποία απεικονίζει έναν 32-bit ακολουθιακό αριθμό n σε ένα L-bit ρεύμα κλειδιού, υπό τον έλεγχο ενός 160-bit μυστικού κλειδιού **α**
- Παραγωγή του ρεύματος κλειδιού (Περιλαμβάνει ένα στάδιο προ-επεξεργασίας)

Στάδιο προ-επεξεργασίας

- Το μυστικό κλειδί α «απλώνεται» σε πίνακες μεγαλύτερου μήκους, T, S και R
- Οι πίνακες T, S και R μπορούν να υπολογιστούν πριν την εφαρμογή του αλγορίθμου, αν είναι γνωστό το κλειδί **α**
- Οι πίνακες T και S έχουν μέγεθος 2K και 1K αντίστοιχα. Το μέγεθος του πίνακα **R** εξαρτάται από το επιθυμητό μήκος **L**

Παραγωγή του ρεύματος κλειδιού (i)

- Είσοδος α, ακέραιος n με $0 \leq n < 232$, L
- Έξοδος Ρεύμα κλειδιού μήκους L' bits, όπου L' το ελάχιστο κοινό πολλαπλάσιο του 128, για το οποίο ισχύει ότι είναι μεγαλύτερο ή ίσο του L

Παραγωγή του ρεύματος κλειδιού (ii)

- Χρησιμοποιούνται ως βοηθητικές για τους υπολογισμούς οι 32-bit ποσότητες A, B, C, D, Xi και Yi
- Στις περισσότερες εφαρμογές του SEAL 2.0 αναμένεται ότι $L \leq 219$
- Μεγαλύτερες τιμές του L επιβαρύνουν το συνολικό κόστος, καθώς οδηγούν σε μεγαλύτερο μήκος του πίνακα R

Επιθέσεις

- Βασίζονται στην παρατήρηση ότι οι λέξεις A, B, C και D τροποποιούνται με την χρήση του πίνακα T μόνο δύο φορές. Το γεγονός αυτό δημιουργεί «τρύπες», όπου βασικά στοιχεία των υπολογισμών, καθίστανται ορατά.
- Άλλες ιδέες βασίζονται στην εφαρμογή ταξινομήσεων στα bits των yj και η πρόβλεψη και επικύρωση μιας συσχέτισης μεταξύ των i και T[i].

Πρώτη επίθεση

- Η επίθεση στον ασθενή αλγόριθμο στηρίζεται στο ότι καθένα από τα A, B, C, D τροποποιούνται δύο φορές χρησιμοποιώντας το T.

- Οι τροποποιήσεις είναι άμεσα ορατές στον αντίπαλο.
- Καθορίζουμε ένα βασικό α και παρέχουμε τις αντίπαλες σειρές παραγωγής δειγμάτων με την μορφή y=WEAK(a,n)
- Ο αντίπαλος το n που παρήγαγε κάθε συμβολοσειρά y.
- Αποτυπώνουμε μια από τις συμβολοσειρές y που ο αντίπαλος συλλέγει και γράφουμε y=y0y1y2… για τις λέξεις του

Δεύτερη επίθεση

- Ταξινόμηση στα κομμάτια Yj
- Ταξινομούμε y τιμές σε 512 κάδους ανάλογα με την τιμή των τελευταίων 9 bits toy y3
- Θεωρούμε P1 την σχετική απόσταση του T που είναι η τιμή του P που καθορίζεται στην γραμμή 1
- Όλες οι σειρές y έχουν τις ίδιες τιμές
- Επομένως υποθέτουμε 512 διαφορετικές τιμές ότι είναι ίδιες εκτός από μια μετατόπιση και έτσι αυτές είναι οι καταχωρήσεις του T

Τρίτη επίθεση

- Η επίθεση στηρίζεται στο T που είναι μικρό και παράγεται τυχαία
- Πχ αν το λιγότερο σημαντικό κομμάτι i συμφωνήσει με το ένατο κομμάτι T[i] σε 256 από 512 λέξεις η σταθερή απόκλιση θα είναι 11.Και άρα αυτά τα 2 μπιτ συμφώνησαν 240 ή 270 φορές.
- Γίνονται ουσιαστικοί συσχετισμοί κάποιου κομματιού i και κάποιου ιδιαίτερου κομματιού T[ι]
- Ο αντίπαλος μπορεί να κάνει συσχετισμούς των σημείων αυτών βασισμένος σε ένα δείγμα y τιμών

Απόδοση Αλγορίθμων SEAL,RC4,RC5,DES,MD5. [4]

Μετράμε τους κύκλους ρολογιών σχετικά με αφηρημένο πρότυπο μηχανών

Για καθένα από τους αλγορίθμους SEAL χρησιμοποιήθηκε (ASSEMBLY) για επεξεργαστή Pentium.

Η απόδοση μετρήθηκε με μια μηχανή 90 MHz.

Το κόστος της βασικής οργάνωσης αγνοείται

Δίνονται μερικές αποδόσεις αλγορίθμων καθώς και η σχετική ταχύτητα που είναι η ταχύτητα του SEAL δια την ταχύτητα του υποδεδειγμένου αλγορίθμου

.

Αλγόριθμος	Mbit/δευτ	Σχετική ταχύτητα
SEAL	**198**	**1.0**
RC4	**110**	**1.8**
RC5-32/12	**38.4**	**5.2**
DES	**16.9**	**11.7**
MD5	**133.1**	**1.5**

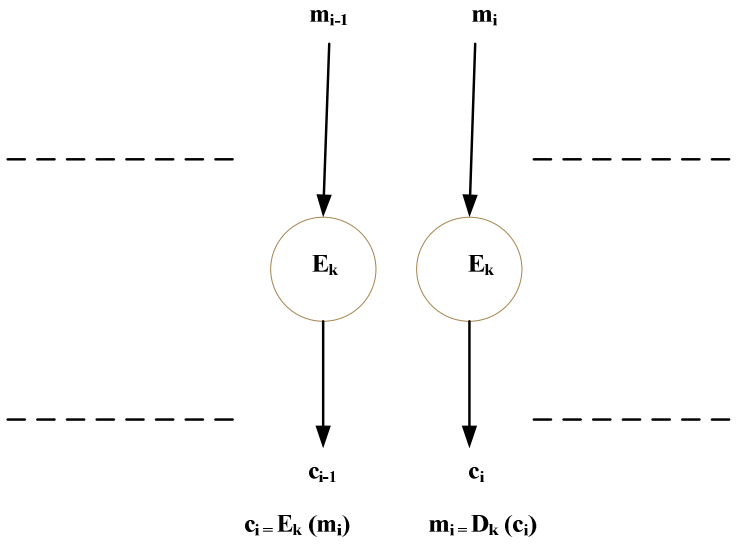
Πίνακας σύγκρισης ταχύτητας

2.1.1.3 Τρόποι λειτουργίας (Modes of Operation) [3]

Όπως έχει αναφερθεί προηγουμένως , όταν κάποιος θέλει να κρυπτογραφήσει δεδομένα με μέγεθος που δεν είναι πολλαπλάσιο του μεγέθους του μπλοκ δεδομένων τότε χρησιμοποιείται μια διαδικασία συμπλήρωσης (padding) με bits ώστε να προκύψει plaintext πολλαπλάσιου μεγέθους του block size που χρησιμοποιεί ο αλγόριθμος. Στην συνέχεια υπάρχουν δύο δυνατότητες : είτε θα κρυπτογραφηθεί κάθε μπλοκ ανεξάρτητα από τα άλλα , είτε θα ακολουθηθεί μία πιο πολύπλοκη μέθοδος , σύμφωνα με την οποία η κρυπτογράφηση κάθε μπλοκ εξαρτάται από κάποιες παραμέτρους που προέκυψαν από την κρυπτογράφηση των προηγούμενων μπλοκ. Αυτές οι διαδικασίες ονομάζονται τρόποι λειτουργίας (Modes of Operation) και είναι αναγκαίο η επιλογή κάποιου εξ' αυτών να γίνεται με γνώμονα την μέγιστη ασφάλεια και την ελαχιστοποίηση του υπολογιστικού φόρτου του αλγόριθμου. Σύμφωνα με την λίστα των αποδεχούμενων τρόπων λειτουργίας (Approved Modes of Operation) οι αλγόριθμοι DES και Triple-DES έχουν 7 αποδεχούμενους , ενώ ο AES έχει 5 .
Οι κυριότεροι τρόποι λειτουργίας, οι οποίοι εφαρμόζονται αποκλειστικά σε όλες τις πρακτικές εφαρμογές είναι ο ECB (Electronic Code Book) και ο CBC (Cipher Block Chaining Mode) .Ας αναφερθούμε σ'αυτούς συνοπτικά.

2.1.1.3.1 Λειτουργία ECB (Electronic Code Book)

Σε ECB mode, το κείμενο χωρίζεται σε ισομήκη block. Κάθε μη κρυπτογραφημένο block κρυπτογραφείται ανεξάρτητα από την συνάρτηση του βασικού block cipher. Η λειτουργία ECB (ΣΧΗΜΑ 4) είναι η πιο απλή, παρουσίαζει όμως κάποια προβλήματα.



ΣΧΗΜΑ 4 Electronic Code Book

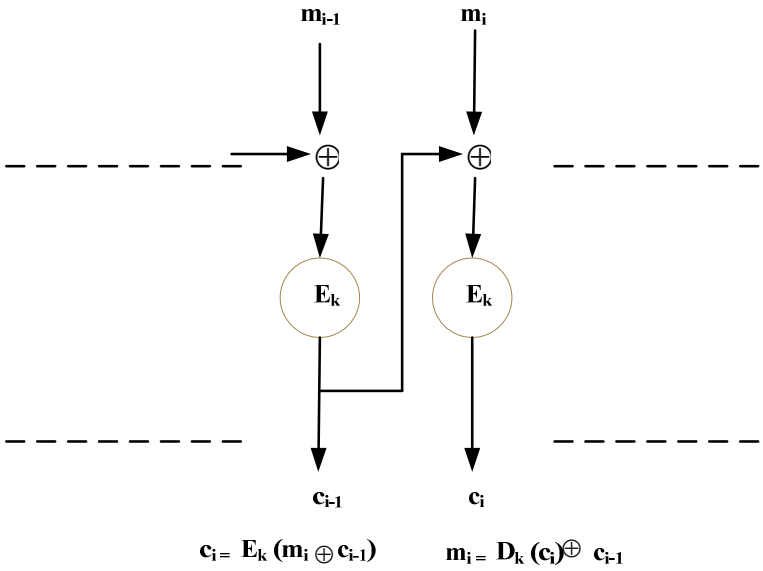
Τα κυριότερα είναι τα εξής:

1. Τα δυαδικά ψηφία που αποτελούν το κρυπτογραφημένο μήνυμα (ciphertext) είναι πάντοτε τα ίδια όταν το μη κρυπτογραφημένο μήνυμα (plaintext) παραμένει το ίδιο. Αυτό έχει επίσης ως αποτέλεσμα την εμφάνιση συγκεκριμένων συστοιχιών bits (bit patterns) μέσα στα κρυπτογραφημένα δεδομένα, γεγονός που μπορεί να εκμεταλλευτεί κάποιος κακόβουλος και να δημιουργήσει το αρχικό μήνυμα εύκολα.
2. Η αλλαγή ενός bit στο αρχικό μήνυμα θα επηρεάσει μόνο το συγκεκριμένο μπλοκ στο ciphertext. Το γεγονός αυτό είναι απευκταίο καθώς θα ήταν προτιμότερο να επηρεάζεται συνολικά το ciphertext για λόγους ασφάλειας.

Μια λύση στα παραπάνω μειονεκτήματα είναι να προστεθεί μία παράμετρος τυχαίου αριθμού στην διαδικασία ώστε η κρυπτογράφηση ενός μπλοκ να επηρεάζεται από τις τιμές των προηγούμενων μπλοκ. Αυτό επιτυγχάνεται με την λειτουργία CBC (Cipher Block Chaining Mode).

2.1.1.3.2 Λειτουργία CBC (Cipher Block Chaining Mode) [3]

Η λειτουργία αυτή (**ΣΧΗΜΑ 5**) χρησιμοποιεί ένα συμπληρωματικό μπλοκ , το οποίο χρησιμοποιείται για την αρχικοποίηση της διαδικασίας (μπλοκ c_0 του σχήματος) , καθώς και την λογική πράξη αποκλειστικό-ή (XOR). Συγκεκριμένα κάθε μπλοκ του plaintext υπόκειται στη λογική πράξη XOR με τα προηγούμενα μπλοκ του ciphertext , και στη συνέχεια κρυπτογραφείται.



ΣΧΗΜΑ 5 Cipher Block Chaining Mode

Με την λειτουργία CBC επιτυγχάνουμε δύο σημαντικά πράγματα:

1. Η κρυπτογράφηση ενός μηνύματος ποτέ δεν θα έχει το ίδιο αποτέλεσμα , το ciphertext δηλαδή δεν θα είναι ποτέ το ίδιο.
2. Η αλλαγή ενός bit στο plaintext θα επηρεάσει όχι μόνο το μπλοκ στο οποίο ανήκει , αλλά θα αλλάξει ολόκληρο το ciphertext.

Είναι, συνεπώς, κατανοητό το γεγονός ότι οι αλγόριθμοι DES, Triple-DES, και AES χρησιμοποιούνται ευρέως σε λειτουργία CBC σε πρωτόκολλα όπως IPSec, SSL και TLS.

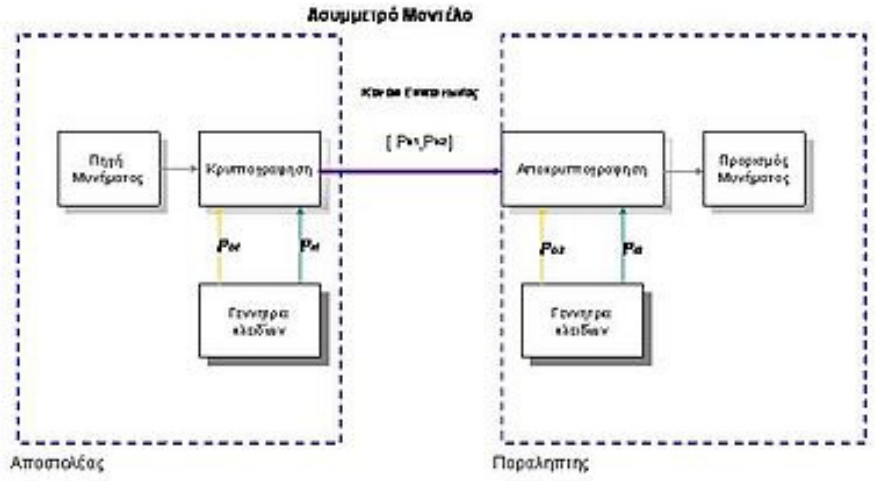
2.1.2 Ασύμμετροι αλγόριθμοι [1],[2]

Οι ασύμμετροι αλγόριθμοι ή αλγόριθμοι δημόσιου κλειδιού είναι σχεδιασμένοι έτσι ώστε το κλειδί που χρησιμοποιείται για την κρυπτογράφηση να είναι διαφορετικό από το κλειδί που χρησιμοποιείται για την αποκρυπτογράφηση. Πέρα από αυτό, το κλειδί αποκρυπτογράφησης δεν μπορεί να υπολογιστεί από το κλειδί κρυπτογράφησης. Οι αλγόριθμοι αυτοί καλούνται και "δημόσιου κλειδιού" γιατί το κλειδί κρυπτογράφησης μπορεί να δημοσιοποιηθεί. Ο καθένας μπορεί να κρυπτογραφήσει ένα μήνυμα με το δημόσιο κλειδί αλλά μόνο αυτός που διαθέτει το αντίστοιχο ιδιωτικό κλειδί μπορεί να το αποκρυπτογραφήσει.

Τα στάδια της επικοινωνίας του σχήματος **6** είναι τα ακόλουθα:

1. Η γεννήτρια κλειδιών του Κώστα παράγει 2 ζεύγη κλειδιών,
2. Η γεννήτρια κλειδιών της Μαρίας παράγει 2 ζεύγη κλειδιών
3. Η Μαρία και ο Κώστας ανταλλάσσουν τα δημόσια ζεύγη

4. Ο Κώστας δημιουργεί ένα μήνυμα όπου τα σύμβολα m ανήκουν στον χώρο των μηνυμάτων.
5. Κρυπτογραφεί το μήνυμα με το δημόσιο κλειδί της Μαρίας και η παραγόμενη κρυπτοσυμβολοσειρά αποστέλλεται
6. Η Μαρία λαμβάνει την κρυπτοσυμβολοσειρά και στη συνέχεια με το ιδιωτικό της κλειδί την αποκρυπτογραφεί και η έξοδος που παράγεται είναι το μήνυμα.



ΣΧΗΜΑ 6 Μοντέλο Ασύμμετρου Κρυπτοσυστήματος

2.1.2.1 Λίστα κυριότερων Ασύμμετρων Κρυπταλγορίθμων [1]

- RSA
- Πρωτόκολλο Diffie-Hellman
- DSA
- Προτυπο ElGamal - Υπογραφή ElGamal
- Κρυπτογραφία ελλειπτικών καμπυλών(ECC)

2.1.2.1.1 RSA [5]

Ο RSA είναι ένας κρυπταλγόριθμος ασύμμετρου κλειδιού, το όνομα του οποίου προέρχεται από τους δημιουργούς του, Ron Rivest, Adi Shamir and Len Adleman. Επιτρέπει όχι μόνο την κωδικοποίηση μηνυμάτων αλλά μπορεί επίσης να χρησιμοποιηθεί και ως ψηφιακή υπογραφή**.**

Λειτουργία

Ο RSA βασίζεται στη δυσκολία παραγοντοποίησης μεγάλων αριθμών (σήμερα, συνήθως της τάξης των 1024 με 2048 bits). Χρησιμοποιούνται δυο κλειδιά, ένα δημόσιο κατά τη διάρκεια της κρυπτογράφησης και ένα ιδιωτικό για την αποκρυπτογράφηση.

Δημιουργία των κλειδιών

1. Επιλογή δυο τυχαίων (μεγάλων) πρώτων αριθμών pκαι qέτσι ώστε $p \neq q$
2. Υπολογίζουμε $n = p.q$
3. Υπολογίζουμε την συνάρτηση του Όιλερ, $\phi(n) = (p-1)(q-1)$.
4. Επιλογή ενός αριθμού $e > 1$έτσι ώστε $e^{\phi(n)} \equiv 1 \pmod{n}$.
5. Υπολογίζουμε τον αριθμό dέτσι ώστε $d \equiv e^{-1} \pmod{\varphi(n)}$.

- Για την εύρεση πρώτων αριθμών χρησιμοποιούνται πιθανολογικοί αλγόριθμοι.
- Συνηθισμένες επιλογές για το e είναι το 3, 7 και $2^{16} + 1$. Μικροί αριθμοί οδηγούν σε ταχύτερους υπολογισμούς αλλά και σε πιο αδύνατη ασφάλεια.

Τα κλειδιά είναι τα εξής:

- δημόσιο: (n, e)
- ιδιωτικό: (n, d)

Μπορούμε τώρα να δημοσιεύσουμε το πρώτο κλειδί, δίνοντας έτσι τη δυνατότητα σε οποιονδήποτε να μας στείλει κρυπτογραφημένα μηνύματα που μόνο εμείς (χάρη στο ιδιωτικό κλειδί) μπορούμε να αποκρυπτογραφήσουμε.

Κρυπτογράφηση

Το μήνυμα μπορεί να αντιπροσωπευθεί από έναν αριθμό m(π.χ. "RSA" → 0x525341, όπου *0x52* είναι ο δεκαεξαδικός κωδικός ASCII του χαρακτήρα *R*, *0x53* του *S* και τέλος *0x41* του *A*). Το κρυπτογραφημένο μήνυμα cυπολογίζεται με τον εξής τρόπο:

$$c = m^e \mod n$$

Αποκρυπτογράφηση

Αφού ληφθεί ένα κρυπτογραφημένο μήνυμα c, για να διαβάσουμε το αρχικό μήνυμα προβαίνουμε στον ακόλουθο υπολογισμό:

$$m = c^d \mod n \equiv (m^e)^d \mod n \equiv m^{e.d} \mod n$$

Ξέρουμε πως $e.d \equiv 1 \pmod{p-1}$ και $e.d \equiv 1 \pmod{q-1}$, όποτε με το μικρό θεώρημα του Φερμά, έχουμε:

$$m^{e.d} \equiv m^1 \equiv m \mod p-1$$

και

$$m^{e.d} \equiv m^1 \equiv m \mod q-1$$

Οι αριθμοί *p* και *q* είναι πρώτοι μεταξύ τους, χρησιμοποιώντας λοιπόν το Κινέζικο Θεώρημα Υπολοίπων, έχουμε:

$$m^{e.d} \equiv m \mod n$$

Ψηφιακή υπογραφή

Ο RSA επιτρέπει την ψηφιακή υπογραφή μηνυμάτων. Αν θέλουμε να αποστείλουμε ένα υπογεγραμμένο μήνυμα, μπορούμε να το κάνουμε με τον εξής τρόπο (χρησιμοποιώντας το ιδιωτικό κλειδί *(n, d)*: $s = m^d \mod n$

Ο παραλήπτης του μηνύματος *m* και της υπογραφής *s*, υπολογίζει την τιμή s^e χάρη στο δημόσιο κλειδί *(n, e)* και τη συγκρίνει με το *m*. Αυτή η λύση, αν και λειτουργεί, δεν χρησιμοποιείται ποτέ, για λόγους ασφαλείας. Αντί να υπογραφεί το μήνυμα ως έχει, προτιμάται η χρήση μιας συνάρτησης κατακερματοποίησης *(hash function) H*:

$$s = H(m)^d \mod n$$

Ο παραλήπτης προβαίνει στη ίδια μέθοδο, αρκεί να γνωρίζει και ποία συνάρτηση κατακερματοποίησης χρησιμοποιήθηκε.

Ασφάλεια

Αν και ο αλγόριθμος θεωρείται ασφαλής όταν χρησιμοποιούνται πολύ μεγάλες παράμετροι, η κακή του χρήση μπορεί να οδηγήσει σε μεγάλες αδυναμίες ασφάλειας.

Εκτός από αυτό, μέχρι σήμερα κανένας δεν έχει αποδείξει ότι η ασφάλεια του εξαρτάται *αποκλειστικά* από την παραγοντοποίηση των ακεραίων.

Επίσης, υπάρχει πάντα η πιθανότητα να ανακαλύψει κάποιος έναν αλγόριθμο (ή να έχει ήδη ανακαλύψει) ο οποίος μπορεί να παραγοντοποιεί αριθμούς σε πολυωνυμικό χρόνο.

Επιθέσεις

Επίθεση επαναληπτικής κρυπτογράφησης

Αφού ο αλγόριθμος χρησιμοποιεί επαναληπτική συνάρτηση είναι δυνατός ένας τρόπος επίθεσης με τη χρήση επαναλαμβανόμενων κρυπτογραφήσεων. Αν έχουμε στην κατοχή μας το κρυπτογραφημένο μήνυμα και το δημόσιο κλειδί με το οποίο κρυπτογραφήθηκε τότε μπορούμε να ακολουθήσουμε την εξής διαδικασία:

Κρυπτογραφούμε το ήδη κρυπτογραφημένο μήνυμα με το δημόσιο κλειδί. Επαναλαμβάνουμε τη διαδικασία κρυπτογράφησης του αποτελέσματος μέχρι να πάρουμε κείμενο ίδιο με το πρώτο κρυπτογραφημένο μήνυμα. Η αμέσως προηγούμενη κρυπτογράφηση περιέχει το αποκρυπτογραφημένο κείμενο.

Προβλήματα που οφείλονται στην κακή χρήση ή υλοποίηση

Κοινό *n*

Αν υποθέσουμε πως έχουμε στην κατοχή μας δυο κλειδιά του τύπου (n, e_1)και (n, e_1)(το ίδιο *n*), και δυο κρυπτογραφήσεις (c_1, c_2)του ιδίου μηνύματος *m* με τα κλειδιά αυτά (π.χ. αν "κρυφακούμε" σε ένα δίκτυο):

$$c_1 = m^{e_1} \mod n$$

και

$$c_2 = m^{e_2} \mod n$$

Μπορούμε να βρούμε το αρχικό μήνυμα *m* χωρίς να έχουμε πρόσβαση στα κρυφά κλειδιά. Είναι πολύ πιθανόν να έχουμε:

$$e_1 \wedge e_2 = 1$$

οπότε και με το θεώρημα του Bézout:

$$\exists\ (u, v),\ e_1.u + e_2.v = 1$$

Για να βρούμε το αρχικό μήνυμα *m*, υπολογίζουμε λοιπόν:

$$(c_1)^u.(c_2)^v \equiv (m^{e_1})^u.(m^{e_2})^v \equiv m^{e_1.u+e_2.v} \equiv m^1 \equiv m \mod n$$

Μικρό *e* (π.χ. *e* = 3)

Ένα μήνυμα *m* κρυπτογραφείται κι αποστέλλεται από τρεις διαφορετικούς χρήστες με χρήση των δημοσίων κλειδιών $(n_1, 3)$, $(n_2, 3)$και $(n_3, 3)$. Ο κακόβουλος χρήστης έχει λοιπόν στην κατοχή του:

- $m^3 \mod n_1$
- $m^3 \mod n_2$
- $m^3 \mod n_3$

Χάρη στο Κινέζικο Θεώρημα Υπολοίπων, μπορεί να υπολογίσει:

$$m^3 \mod n_1.n_2.n_3$$

και να βρει πια εύκολα το αρχικό μήνυμα *m*.

Τυφλή υπογραφή

Υποθέτουμε πως ο Γιάννης, το ιδιωτικό (αντ. δημόσιο) κλειδί του οποίου είναι (*n*, *d*) (αντ. (*n*, *e*)), υπογράφει ότι μήνυμα του δώσουμε χωρίς δεύτερη σκέψη. Αν ένας κακόβουλος χρήστης έχει ένα κρυπτογραφημένο μύνημα *c* με τελικό παραλήπτη τον Γιάννη, μπορεί να μπερδέψει τον τελευταίο έτσι ώστε να του το αποκρυπτογραφήσει ο ίδιος ο Γιάννης. Αρκεί να διαλέξει έναν τυχαίο αριθμό *r*, πρώτο με το *n* και να ζητήσει από τον Γιάννη να του υπογράψει το μήνυμα $m' = r^e.c$. Ο Γιάννης υπολογίζει:

$$m'^d \equiv (r^e.c)^d \equiv (r^e.m^e) \equiv r^{e.d}.m^{e.d} \equiv r.m \mod n$$

Το μήνυμα *r.m* δεν είναι κατανοητό, οπότε ο Γιάννης δεν μπορεί εύκολα να καταλάβει πως πέφτει θύμα απάτης και το στέλνει στον κακόβουλο χρήστη, ο οποίος υπολογίζει τον αριθμό $r^{-1} \mod n$ και μπορεί πλέον να διαβάσει το μήνυμα *m*.

Για να αποφύγει το πρόβλημα αυτό, ο Γιάννης δεν πρέπει να χρησιμοποιεί το ίδιο κλειδί για την υπογραφή και για την αποκρυπτογράφηση μηνυμάτων, ούτε όμως και να υπογράφει ό,τι του ζητούν "στα τυφλά".

2.1.2.1.2 Πρωτόκολλο Diffie-Hellman [6]

Το πρωτόκολλο των Diffie-Hellman παρουσιάστηκε το 1976 από τους Whitfield Diffie και Martin Hellman. Πριν από τη δημιουργία αυτού κάθε κρυπτογραφική τεχνική βασιζόταν σε κάποιο προσυμφωνημένο κλειδί. Το συγκεκριμένο πρωτόκολλο είναι το πρώτο που προτάθηκε ώστε να επιτρέπει σε δυο οντότητες, χωρίς προηγούμενη επικοινωνία, να ανταλλάξουν ένα κοινό κλειδί μέσω ενός μη ασφαλούς διαύλου επικοινωνίας.

Περιγραφή πρωτοκόλλου

Η πρωτότυπη εφαρμογή του πρωτοκόλλου χρησιμοποιεί την πολλαπλασιαστική ομάδα των ακεραίων modulo p, όπου p είναι πρώτος αριθμός και g είναι γεννήτορας της πολλαπλασιαστικής ομάδας mod p.

Τα βήματα του πρωτοκόλλου για δυο οντότητες A και B, οι οποίες θέλουν να ανταλλάξουν ένα μυστικό κλειδί, είναι τα ακόλουθα:

- Ο Α υπολογίζει ένα μυστικό κλειδί a το οποίο δεν πρόκειται να αποκαλύψει σε κανένα στάδιο του πρωτοκόλλου και τους τυχαίους αριθμούς g, p επιλέγοντας για p έναν πρώτο αριθμό. Στέλνει στον Β το μήνυμα: 'g, p, g^a mod p'.
- Ο Β λαμβάνει το μήνυμα, επιλέγει με τη σειρά του ένα μυστικό κλειδί b, και στέλνει στον Α το μήνυμα: 'g^b mod p'.

Μετά το τέλος αυτών των μηνυμάτων και οι δυο οντότητες γνωρίζουν έναν αριθμό ο οποίος δεν είναι γνωστός από κανένα άλλο, τον g^{ab} mod p. Παρά το γεγονός ότι μέσα στο μη ασφαλές κανάλι έχουν περάσει οι πληροφορίες: g,p,g^amodp, g^bmodp, κανένας άλλος, εκτός των Α και Β δεν μπορεί να υπολογίσει το g^{ab}modp, καθώς κάτι τέτοιο θα σημαίνει ουσιαστικά ότι είναι δυνατό σε ρεαλιστικό χρόνο να υπολογιστεί ο διακριτός λογάριθμος.

Παράδειγμα εφαρμογής του πρωτοκόλλου

Έστω ότι ο Α διαλέγει τους παρακάτω αριθμούς:

p=563, g=5, a=9.

Στέλνει λοιπόν στον Β:

$5,563,5^9 \text{mod} 563 \equiv 5,563,1953125 \text{mod} 563 \equiv 5,563,78$

Ο Β επιλέγει b =14 και στέλνει στον Α:

$5^{14} \text{mod} 563 \equiv 6103515625 \text{mod} 563 \equiv 534$

Τώρα και οι δυο μπορούν να υπολογίσουν το

$(g^a \text{mod } p)^b \text{mod } p \equiv (g^b \text{ mod } p)^a \text{mod } p \equiv g^{ab} \text{mod } p \equiv$
$153312511596308814665178236828300148736 \text{ mod } 563 = 117$

Συνεπώς το μυστικό κλειδί είναι το 117.

Ασφάλεια πρωτοκόλλου

Η ασφάλεια του πρωτοκόλλου είναι στενά συνδεδεμένη με την επιλογή των στοιχείων p και g. Κάποιος ο οποίος θέλει να επιτεθεί στο πρωτόκολλο πρέπει ουσιαστικά να αποκτήσει τα g,a και b. Προκειμένου να αποφευχθεί κάτι τέτοιο πρέπει να προσέξουμε τις επιλογές των g, a, b, p. Η τάξη G του g πρέπει να είναι πρώτος αριθμός ή να έχει ως παράγοντα ένα πολύ μεγάλο πρώτο αριθμό για να αποφευχθεί η χρήση του αλγορίθμου των Pohlig-Hellman η όποια θα δώσει τους αριθμούς a, b. Έτσι πολύ συχνά αναζητούμε να βρούμε **πρώτους της Sophie Germain**, πρώτους δηλαδή για οποίους ισχύει ότι αν p είναι ο πρώτος μας τότε p = 2q + 1, όπου q επίσης πρώτος. Το g συνήθως το επιλέγουμε να παράγει την πολλαπλασιαστική υποομάδα τάξης q του G, και όχι ολόκληρη την G. Με αυτόν τον τρόπο εξασφαλίζουμε ότι το σύμβολο Legendre του g δεν πρόκειται να αποκαλύψει

ποτέ κανένα ψηφίο του a.
Αν οι δυο οντότητες A και B δεν χρησιμοποιούν καλές γεννήτριες τυχαίων αριθμών, τότε τα a και b είναι πιθανόν να μπορούν να προσβληθούν από κάποιον ο οποίος παρακολουθεί τα δεδομένα που περνούν στο κανάλι. Πρέπει να τονίσουμε πως μετά το τέλος του πρωτοκόλλου, οι μυστικοί ακέραιοι a και b εξαφανίζονται και από τις δυο συσκευές των δυο οντοτήτων προκειμένου να μην μείνουν στοιχεία για περαιτέρω μελέτη από κάποιο άλλο πιθανό επιτιθέμενο.

Επίθεση επανάληψης

Το πρωτόκολλο έχει ένα βασικό πρόβλημα το οποίο έχει να κάνει με το γεγονός ότι σε κανένα μήνυμα που ανταλλάσσεται μεταξύ των οντοτήτων δεν γίνεται επικύρωση της ώρας κατά της οποία έγινε η επικοινωνία. Αυτό έχει ως αποτέλεσμα ένας επιτιθέμενος να μπορεί να επαναλάβει την όλη επικοινωνία την οποία έχει προηγούμενος καταγράψει μεταξύ δυο οντοτήτων, προκειμένου να εξαπατήσει μια από αυτές. Η λύση του προβλήματος δίνεται μέσω **χρονοσφραγίδων** (timestamps), φαινομενικά τυχαίων αριθμών οι οποίοι παράγονται από το σύστημα και έχουν αποθηκευμένη μέσα τους την ώρα έκδοσης τους.

Αυθεντικοποίηση

Κατά τη διάρκεια της αρχικής περιγραφής, το πρωτόκολλο Diffie-Hellman από μόνο του δεν παρέχει πιστοποίηση της ταυτότητας των οντοτήτων της επικοινωνίας και επομένως είναι ευάλωτο σε μία Man-in-the-middle επίθεση. Ένα ενδιάμεσο άτομο μπορεί δηλαδή να συστήσει δύο διακριτές Diffie-Hellman ανταλλαγές κλειδιών, μία με τη μία οντότητα και μία με την άλλη , υποδυόμενος στην κάθε μία οντότητα την άλλη. Συνεπώς, είναι απαραίτητο κατά τη χρήση του πρωτοκόλλου να γίνεται εξακρίβωση της ταυτότητας της κάθε οντότητας για την πρόληψη αυτού του είδους των επιθέσεων

2.1.2.1.3 Πρότυπο ElGamal [7]

Ορισμός - Πρότυπο ψηφιακών υπογραφών ElGamal

Η ασφάλεια του συστήματος των ψηφιακών υπογραφών ElGamal βασίζεται στη δυσκολία του υπολογισμού του διακριτού λογάριθμου από τον αντίπαλο. Για την υλοποίηση του συστήματος ψηφιακών υπογραφών ElGamal απαιτείται κρυπτογραφική μονόδρομη hash, της οποίας η σύνοψη είναι στοιχείο του συνόλου Zp*, όπου p πρώτος αριθμός.

Η υποδομή ενός συστήματος ψηφιακών υπογραφών ElGamal απαιτεί την ακόλουθη διαδικασία δημιουργίας ζεύγους κλειδιών από τα μέλη. Αρχικά επιλέγεται ένας μεγάλος πρώτος αριθμός p και ένας ακέραιος a ο οποίος είναι γεννήτορας του συνόλου Zp*. Στη συνέχεια επιλέγεται ένας ακέραιος b τέτοιος ώστε $0 < b < p–1$, και υπολογίζεται το:
$y \equiv a^b \pmod p$

Το δημόσιο κλειδί αποτελείται από τους τρεις ακέραιους (p, a, y) ενώ το ιδιωτικό κλειδί είναι ο εκθέτης b. Η παραπάνω διαδικασία εκτελείται από κάθε μέλος. Κατά τη διαδικασία υπογραφής, εκτελείται το ακόλουθο πρωτόκολλο:

1. Επιλογή μυστικού ακεραίου k, με $0 < k < p–1$, και $\gcd(k, p–1) = 1$
2. Υπολογισμός του $r \equiv a^b \pmod p$
3. Υπολογισμός του $k^{-1} \bmod p$
4. Υπολογισμός του $s \equiv k^{-1} (h(m) – br) \pmod{p–1}$
5. Η υπογραφή για το μήνυμα m είναι το ζεύγος (r, s), το οποίο αποστέλλεται μαζί με το μήνυμα στον παραλήπτη.

Η διαδικασία επαλήθευσης πραγματοποιείται με το ακόλουθο πρωτόκολλο:
1. Έλεγχος ότι $0 < r < p–1$. Στην περίπτωση που το r δε βρίσκεται μεταξύ των ενδεδειγμένων ορίων, απορρίπτεται η ψηφιακή υπογραφή.
2.Υπολογισμός του $v \equiv y^r r^s \pmod p$.
3.Υπολογισμός της σύνοψης h(m) και υπολογισμός του $v' \equiv a^{h(m)} \pmod p$.
4.Η υπογραφή θεωρείται έγκυρη αν και μόνο αν $v = v'$.

Μπορούμε να επαληθεύσουμε την εγκυρότητα της υπογραφής με την ισοδυναμία του τελευταίου βήματος του πρωτοκόλλου επαλήθευσης ως εξής:
$s \equiv k^{-1}(h(m)-br) \pmod{p-1} \Rightarrow$
$ks \equiv h(m)-br \pmod{p-1} \Rightarrow$
$h(m) \equiv ks + br \pmod{p-1} \Rightarrow$
$a^{h(m)} \equiv a^{ks + br} \pmod p \Rightarrow$
$a^{h(m)} \equiv (a^b)^r (a^k)^s \pmod p \Rightarrow$
$a^{h(m)} \equiv y^r r^s \pmod p$
ή ισοδύναμα $v' = v$

Ασφάλεια του συστήματος ψηφιακών υπογραφών ElGamal

Αν θεωρήσουμε ότι το πρόβλημα του διακριτού αλγόριθμου είναι υπολογιστικά αδύνατο, τότε αν ο αντίπαλος επιλέξει στην τύχη έναν ακέραιο για υποψήφιο ιδιωτικό κλειδί, η πιθανότητα να επιλέξει το σωστό κλειδί είναι ίση με $1/(p–1)$, εφόσον οι επιτρεπτές τιμές του ιδιωτικού κλειδιού βρίσκονται στο διάστημα $0 < b < p–1$. Επομένως, το p θα πρέπει να είναι αρκετά μεγάλο ώστε η πιθανότητα εύρεσης του ιδιωτικού κλειδιού να είναι μικρή.
Ένα άλλο σημείο το οποίο θέτει σε κίνδυνο το σύστημα δίνοντας πλεονέκτημα για επιτυχή πλαστογραφία, είναι η επιλογή του τυχαίου ακέραιου k, κατά τη διαδικασία δημιουργίας της ψηφιακής υπογραφής. Πιο συγκεκριμένα, ο υπογεγραμμένος θα πρέπει να διατηρεί ιστορικό όλων των τυχαίων αριθμών που έχει επιλέξει, ώστε σε κάθε υπογραφή να χρησιμοποιείται διαφορετικός ακέραιος k.

Πρότυπο ElGamal

Το πρότυπο ηλεκτρονικής υπογραφής ElGamal δημιουργήθηκε το 1984 από τον Taher ElGamal παράλληλα με τον αντίστοιχο αλγόριθμο δημοσίου κλειδιού.
Για την περιγραφή του αλγορίθμου, υποθέτουμε πως διαθέτουμε μια ¨καλή¨ συνάρτηση κατακερματισμού H. Έστω τώρα ένας αρκετά μεγάλος αριθμός p και έστω g ένα στοιχείο γεννήτορας της πολλαπλασιαστικής ομάδας των ακεραίων modulo p, την Z^*_p.Επιλέγουμε ένα τυχαίο αριθμό x, με $1 < x < p - 1$ και υπολογίζουμε τον
$y = g^x \bmod p$.

Ο x θα είναι το ιδιωτικό κλειδί, ενώ η τριάδα (p, g, y) είναι το δημόσιο κλειδί.
Για να υπογραφεί ένα μήνυμα m, ο υπογράφων επιλέγει αρχικά έναν τυχαίο k, $1 < k < p - 1$ και $(k, p - 1) = 1$. Στη συνέχεια υπολογίζονται οι αριθμοί
$r \equiv g^k \bmod p$
$s \equiv (H(m) - xr)k^{-1} \bmod (p-1)$
Στην περίπτωση που $s = 0$, επιλέγεται νέο k και υπολογίζονται εκ νέου οι r και s. Το ζεύγος (r, s) είναι η ψηφιακή υπογραφή του μηνύματος m.
Για την επικύρωση της υπογραφής κάποιος ο οποίος λαμβάνει το μήνυμα m και την ψηφιακή υπογραφή (r, s), αρκεί να υπολογήσει την ποσότητα $g^{H(m)} \bmod p$ και να ελέγξει αν είναι ίση με $y^r r^s \bmod p$.
Το H(m) μπορεί να εκφραστεί ως:

$H(m) \equiv xr + sk \bmod (p - 1)$
Η ποσότητα $g^{H(m)}$ θα είναι ίση
$g^{H(m)} \equiv g^{xr + sk} \bmod p$

$\equiv g^{xr} g^{sk} \bmod p$

$\equiv (g^x)^r (g^k)^s \bmod p$

$\equiv y^r r^s \bmod p$
Συνεπώς ο αλγόριθμος είναι επιτυχής.

2.1.2.1.4 Αλγόριθμος Digital Signature Algorithm (DSA) [8]

Ο αλγόριθμος DSA (Digital Signature Algorithm) προτάθηκε τον Αύγουστο του 1991 από το NIST (National Institute of Standards and Technology) της Αμερικής. Έχει προτυποποιηθεί ως FIPS 186 (Federal Information Processing Standard). Το πρότυπο αυτό έχει ονομαστεί DSS (Digital Signature Standard) και είναι ο πρώτος αλγόριθμος ψηφιακής υπογραφής που αναγνωρίστηκε παγκόσμια. Ο DSA αποτελεί μια παραλλαγή του αλγορίθμου ElGamal για ψηφιακές υπογραφές και σχεδιάστηκε αποκλειστικά για τη δημιουργία και επαλήθευση ψηφιακών υπογραφών και κατά συνέπεια και για τον έλεγχο της ακεραιότητας των δεδομένων. Η λογική του αλγορίθμου βασίζεται σε αυτήν της ασύμμετρης κρυπτογραφίας, αφού και σε αυτήν την περίπτωση κάθε οντότητα δημιουργεί ένα ζεύγος δημοσίου και ιδιωτικού κλειδιού. Τα βήματα που ακολουθούνται για την υλοποίηση του αλγορίθμου είναι τα παρακάτω:

1. Επιλέγεται ένας πρώτος αριθμός q τέτοιος ώστε $2^{159} < q < 2^{160}$
2. Επιλέγεται ένας αριθμός t τέτοιος ώστε $0 \le t \le 8$ και ένας πρώτος αριθμός p τέτοιος ώστε $2^{511+64t} < p < 2^{512+64t}$ με την ιδιότητα ο q να διαιρεί τον $(p - 1)$
3. $g = h^{(p-1)/q} \bmod p$, όπου h είναι ένας ακέραιος $1 < h < p - 1$ έτσι ώστε $h^{(p-1)/q} \bmod p > 1$.
4. Έστω x ένας τυχαίος ακέραιος έτσι ώστε $1 \le x \le q - 1$
5. Υπολογίζεται το $y = g^x \bmod p$

6. Το δημόσιο κλειδί είναι το (*p,q,g,y*). Το ιδιωτικό κλειδί είναι το *x*.

Έχοντας υπολογίσει τις παραπάνω παραμέτρους μπορούμε να δημιουργήσουμε μία ψηφιακή υπογραφή. Τα παρακάτω βήματα περιγράφουν τον τρόπο με τον οποίο μπορεί να υπογραφεί ψηφιακά, σύμφωνα με τον αλγόριθμο DSA, ένα μήνυμα *m* τυχαίου μήκους:

1. Επιλέγεται ένας τυχαίος ακέραιος k, $0 < k < q$. Ο ακέραιος k θα πρέπει να μείνει μυστικός.
2. Υπολογίζεται το $r = (g^k \bmod p) \bmod q$.
3. Υπολογίζεται το $k^{-1} \bmod q$.
4. Υπολογίζεται το $s = k^{-1}\{h(m) + xr\} \bmod q$. Η $h(m)$ είναι μια συνάρτηση κατακερματισμού (hash function).Πρόκειται για ένα αλφαριθμητικό μήκους 160 bits που προκύπτει ως έξοδος του αλγορίθμου SHA-1 ο οποίος περιγράφεται παρακάτω.
5. Η ψηφιακή υπογραφή για το μήνυμα *m* είναι το ζεύγος *(r,s)*

Από τη στιγμή που ένας χρήστης Α ενός επικοινωνιακού συστήματος έχει υπογράψει ψηφιακά ένα μήνυμα θα πρέπει οι υπόλοιποι χρήστες του συστήματος να είναι σε θέση να επαληθεύσουν την υπογραφή του. Αυτό γίνεται με τη χρήση του δημοσίου κλειδιού του Α. Τα παρακάτω βήματα περιγράφουν τη διαδικασία της επαλήθευσης μιας ψηφιακής υπογραφής:

1. Το δημόσιο κλειδί (*p,q,g,y*) του χρήστη Α είναι διαθέσιμο.
2. Επαληθεύεται ότι $0 < r < q$ και $0 < s < q$. Αν δεν ισχύουν τα παραπάνω η υπογραφή απορρίπτεται.
3. Υπολογίζεται $w = s^{-1} \bmod q$ και την $h(m)$.
4. Υπολογίζεται το $u_1 = w \cdot h(m) \bmod q$ και το $u_2 = rw \bmod q$.
5. Υπολογίζεται το $\upsilon = (g^{u_1} y^{u_2} \bmod p) \bmod q$.
6. Η υπογραφή είναι αποδεκτή μόνο όταν $\upsilon = r$.

Ασφάλεια του DSA

Η ασφάλεια του DSA βασίζεται στη δυσκολία του υπολογισμού διακριτών λογαρίθμων μέσα σε ένα πεπερασμένο σώμα. Έρευνες πάνω στον αλγόριθμο έχουν δείξει την ύπαρξη πρώτων αριθμών οι οποίοι θα μπορούσαν να οδηγήσουν στη δημιουργία κλειδιών ευάλωτων σε επιθέσεις. Όμως, αυτοί οι αριθμοί είναι ελάχιστοι και μπορούν εύκολα να αποφευχθούν σε μία σωστή διαδικασία δημιουργίας ζεύγους κλειδιών. Όπως φαίνεται και από τα βήματα του αλγορίθμου το μέγεθος του *q* πρέπει να είναι 160 bits ενώ το μέγεθος του *p* μπορεί να είναι οποιοδήποτε πολλαπλάσιο του 64 ανάμεσα στο 512 και το 1024. Ένας πρώτος αριθμός *p* μεγέθους 512 bit προστατεύει το σύστημα οριακά από μια ενδεχόμενη επίθεση. Από το 1996 προτείνεται το μέγεθος του *p* να είναι τουλάχιστον 768 bits. Το πρότυπο FIPS 186 δεν επιτρέπει πρώτους αριθμούς p που το μέγεθός τους ξεπερνά τα 1024 bits.

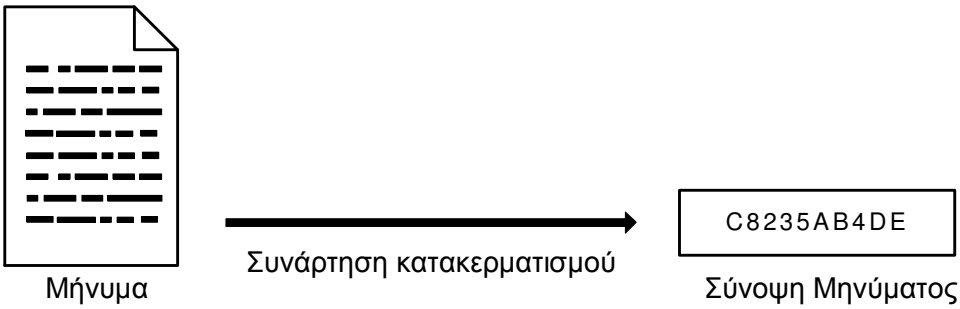
Ένα σημαντικό πλεονέκτημα του DSA είναι ότι, η εκθετοποίηση ως διαδικασία μπορεί να προηγείται της δημιουργίας της ψηφιακής υπογραφής, κάτι που δεν είναι εφικτό με τον RSA.

2.1.2.1.5 Αλγόριθμοι Ελλειπτικών Καμπυλών [8]

Το 1985, οι Neal Koblitz και V. S. Miller πρότειναν ανεξάρτητα ο ένας από τον άλλον την λεγόμενη κρυπτογραφία ελλειπτικών καμπυλών (Elliptic Curve Cryptography). Η Κρυπτογραφία Ελλειπτικών Καμπυλών βασίζεται στο πρόβλημα του διακριτού λογαρίθμου. Συγκεκριμένα δεν υπάρχει γνωστός αλγόριθμος που να επιλύει το πρόβλημα αυτό σε μια κατάλληλα επιλεγμένη ελλειπτική καμπύλη (ECDLP).
Η Κρυπτογραφία Ελλειπτικών Καμπυλών βρίσκει εφαρμογή με τη χρήση των αλγορίθμων DSA και DH ελλειπτικών καμπυλών (ECDSA και ECDH). Οι αλγόριθμοι ECDSA και ECDH υλοποιούνται κάνοντας χρήση ενός συνόλου σημείων, που προκύπτουν ως λύση της εξίσωσης μιας ελλειπτικής καμπύλης πάνω σε ένα πεπερασμένο σώμα (finite field). Η ασφάλειά τους βασίζεται στη δυσκολία του υπολογισμού λογαρίθμων πάνω σε ένα σύνολο σημείων ελλειπτικής καμπύλης. Λαμβάνοντας υπόψη της το πρόβλημα του διακριτού λογαρίθμου σε μια ελλειπτική καμπύλη, η μέχρι σήμερα έρευνα στον τομέα της κρυπτογραφίας έχει δείξει ότι το μήκος των κλειδιών που παράγονται από τους αλγόριθμους ECDSA και ECDH θα πρέπει να είναι τουλάχιστον 192 bits προκειμένου να εξασφαλίζεται επαρκής ασφάλεια στα συστήματα επικοινωνίας τα επόμενα χρόνια.
Ο αλγόριθμος ECDSA είναι ένας κατά FIPS (Federal Information Processing Standard) εγκεκριμένος αλγόριθμος για δημιουργία και επαλήθευση ψηφιακών υπογραφών. Ο ECDSA περιγράφεται στο ANSI X9.62 .Ας σημειωθεί επίσης ότι είναι δυνατόν να υλοποιηθεί και ο RSA ως αλγόριθμος ελλειπτικής καμπύλης. Όμως, η βάση της ασφάλειας αυτού του αλγορίθμου είναι η δυσκολία παραγοντοποίησης μεγάλων ακεραίων και όχι το πρόβλημα του διακριτού λογαρίθμου, με αποτέλεσμα το μέγεθος των κλειδιών να μην είναι σημαντικά μικρότερο από αυτό του συνηθισμένου RSA. Έτσι, η πρόσθετη πολυπλοκότητα δεν έχει κάποιο σημαντικό όφελος, με αποτέλεσμα η χρήση του ECRSA να είναι ιδιαίτερα περιορισμένη.

2.1.3 Συναρτήσεις Κατακερματισμού (Hash Functions) [9]

Ο όρος συνάρτηση κατακερματισμού (hash function) υποδηλώνει ένα μετασχηματισμό H ο οποίος παίρνει ως είσοδο ένα μήνυμα m ανεξαρτήτου μήκους και δίνει ως έξοδο μία ακολουθία χαρακτήρων h, είναι δηλαδή . Η έξοδος h μιας συνάρτησης κατακερματισμού ονομάζεται τιμή κατακερματισμού (hash value) ή σύνοψη μηνύματος (message digest) και έχει συγκεκριμένο μήκος ανάλογα με το είδος του αλγόριθμου κατακερματισμού που χρησιμοποιείται, συνήθως πολύ μικρότερο από αυτό του αρχικού μηνύματος (**ΣΧΗΜΑ 7**). Μπορούμε να φανταστούμε την σύνοψη μηνύματος ως το "ψηφιακό αποτύπωμα" ("digital fingerprint") του εγγράφου.



ΣΧΗΜΑ 7: Συνάρτηση κατακερματισμού

Οι σημαντικότερες ιδιότητες των συναρτήσεων κατακερματισμού με μορφή είναι:

• Η είσοδος x μπορεί να έχει οποιοδήποτε μήκος

• Η έξοδος y έχει περιορισμένο μήκος

• Δεδομένου του x και της συνάρτησης H είναι εύκολος ο υπολογισμός του H(x)

• Η H(x) είναι μονόδρομη (one way function)

• Η H(x) είναι αμφιμονοσήμαντη (συνάρτηση ένα προς ένα)

Μια μονόδρομη συνάρτηση κατακερματισμού είναι μία συνάρτηση κατακερματισμού για την οποία είναι υπολογιστικά ανέφικτο να υπολογιστεί η αντίστροφή της, δηλαδή το αρχικό μήνυμα δεν μπορεί να ανακτηθεί από τη σύνοψή του. Όταν επιπλέον η συνάρτηση είναι αμφιμονοσήμαντη, τότε είναι πολύ δύσκολο να βρεθούν δύο διαφορετικά μηνύματα με την ίδια σύνοψη. Στην περίπτωση που κάτι τέτοιο συμβεί τότε υπάρχει σύγκρουση (collision) .

Οι πιο γνωστοί αλγόριθμοι κατακερματισμού είναι οι MD5 με σύνοψη 128 bit, ο SHA-1 με σύνοψη 160 bits και ο RIPEMD-160 [17] με σύνοψη 160 bits. Οι νέες εκδόσεις του αλγορίθμου SHA, SHA-256, SHA-384 και SHA-512 [18] δίνουν σύνοψη μηνύματος 256, 384 και 512 bits αντίστοιχα.

2.1.3.1 MD5 (Message-Digest algorithm 5) [9]

Γενικά στοιχεία

Ο κρυπταλγόριθμος σχεδιάστηκε το 1992 από τον Rivest, αποτελεί συνέχεια της μονόδρομος hash MD4 λόγω της εύρεσης συγκρούσεων στην MD4 με 220 υπολογισμούς . Προς το παρόν η μόνη αξιόλογη κριτική είναι το μικρό μήκος της σύνοψης που είναι 128 bits, και υπάρχει το ενδεχόμενο επιτυχούς εξαντλητικής αναζήτησης.

Ο MD5 έχει τους εξής στόχους ασφαλείας όπως διατυπώθηκαν από το Rivest **Ασφάλεια**. Θα πρέπει να είναι υπολογιστικά αδύνατο να βρεθούν δύο μηνύματα τα οποία να δίνουν το ίδιο αποτέλεσμα σύνοψης.

Άμεση ασφάλεια. Ο αλγόριθμος δε θα βασίζεται σε υποθέσεις, όπως για παράδειγμα στη δυσκολία παραγοντοποίησης ακεραίων. -26-Cracking LM hash –MD5 –SHA1 Password using FPGA Platforms–Θεοχαρούλης Κ.

Ταχύτητα. Ο αλγόριθμος θα είναι βασισμένος σε απλές λογικές πράξεις και ο σχεδιασμός του θα είναι βελτιστοποιημένος για 32-bit αρχιτεκτονικές υπολογιστών.

Απλότητα και κατάληψη μικρού χώρου. Ο αλγόριθμος θα πρέπει να είναι σχετικά απλός στην περιγραφή του, χωρίς να απαιτεί μεγάλους πίνακες αντικατάστασης τιμών, ή μεγάλα σε μήκος προγράμματα.

Εύνοια αρχιτεκτονικής little-endian. Η αρχιτεκτονική little-endian που είναι βασισμένοι οι επεξεργαστές της Intel x386, αποθηκεύουν το λιγότερα σημαντικό bit σε χαμηλή διεύθυνση μνήμης του byte, σε αντίθεση με την αρχιτεκτονική big-endian που είναι βασισμένοι οι επεξεργαστές Sparc. Έτσι ένας little-endian επεξεργαστής μπορεί να χρησιμοποιεί απ' ευθείας τις αποθηκευμένες δυαδικές λέξεις, ενώ στην περίπτωση του big-endian απαιτείται αντιστροφή. Επειδή γενικά οι big-endian επεξεργαστές είναι γρηγορότεροι στην εκτέλεση πράξεων, θεωρήθηκε ότι η προτίμηση έκφρασης του αλγόριθμου στη μορφή little endian εξισορροπεί τη διαφορά ταχύτητας μεταξύ των δύο οικογενειών επεξεργαστών.

2.1.3.2 Αλγόριθμος Κατακερματισμού (SHA-1) [9]

Ο ασφαλής αλγόριθμος κατακερματισμού SHA-1 (Secure Hash Algorithm-1) αποτελεί μια βελτιωμένη έκδοση του αρχικού αλγορίθμου κατακερματισμού SHA. Αυτός ο αλγόριθμος κατακερματισμού (hash algorithm) σχεδιάστηκε αποκλειστικά για χρήση σε συνδυασμό με τον DSA και συνεπώς δεν μπορεί να χρησιμοποιηθεί με τον RSA ή οποιοδήποτε άλλο αλγόριθμο δημοσίου κλειδιού για ψηφιακή υπογραφή. Οι σχεδιαστικές αρχές του SHA-1 είναι παρεμφερείς με αυτές των συναρτήσεων κατακερματισμού MD2 , MD4 , και κυρίως της συνάρτησης MD5 .

Ο αλγόριθμος μπορεί να έχει ως είσοδο μηνύματα μήκους μικρότερου από bits. Η έξοδος του αλγορίθμου ονομάζεται σύνοψη μηνύματος (message digest ή hash value ή message fingerprint) και έχει μήκος 160 bit. Είναι πιο αργός από τον MD5 αλλά το μεγαλύτερο message digest που παράγει (ο MD5 παράγει message digest μήκους 128 bits) τον καθιστούν πιο ισχυρό σε προσπάθειες αντιστροφής του.

Overheads [13]

Στην κρυπτογραφία απαιτείται συνεχής προσπάθεια για την επίτευξη της ασφάλειας. Αρχικά απαιτούνται προσπάθειες για την παραγωγή των κλειδιών. Μόλις τα κλειδιά παραχθούν, το επόμενο βήμα είναι η κρυπτογράφηση των δεδομένων και η αποστολή τους στο διαδίκτυο. Υπάρχουν διάφορα overheads που συνδέονται στην κρυπτογραφία και δίδονται ως εξής:

- **Οικονομικά Overhead:**

Πολλά χρήματα έχουν επενδυθεί για να κρατηθούν τα έγγραφα ασφαλή από τον εχθρό .

• **Less Channel bandwidth:**
Οι χρήστες είναι σε θέση να χρησιμοποιούν μόνο περιορισμένο εύρος ζώνης , λόγω της παρουσίας πρόσθετων δυαδικών ψηφίων που προκαλούνται από τα κλειδιά .

• **Μεγαλύτερη διάχυση θερμότητας:**
Η κρυπτογράφηση δεδομένων με πολλαπλά κλειδιά που έχουν μεγάλα μήκη εχει παρατηρηθέι σημαντική ποσότητα απαγωγής θερμότητας. Αυτό θέτει έναν περιορισμό σχετικά με τη χρήση συστατικών on-chip το οποίο είναι ιδιαίτερα επιθυμητό για τη γρήγορη διαδικασία κρυπτογράφησης .

• **Κατανάλωση ρεύματος:**
Οι ισχυροί επεξεργαστές καταναλώνουν περισσότερη δύναμη για τη δημιουργία του κλειδιού, ως αποτέλεσμα να έχουμε χωρητικότητα κόμβου , φορτο διαμοιρασμού και διαρροή ρεύματος .Αυτές οι παράμετροι είναι υπεύθυνες για την απώλεια δεδομένων και αιτία βλάβης του σταθμού .

• **Καθυστέρηση :**
Η διαδικασία κρυπτογράφησης χρειάζεται χρόνο για τη μετατροπή του απλού κειμένου σε κρυπτογραφημένο κείμενο που προκαλεί καθυστέρηση και αυξάνει την λανθάνουσα κατάσταση .Κάποιοι αλγόριθμοι κρυπτογράφησης απαιτούν επίσης πρόσθετες τεχνικές παραγεμίσματος που καταναλώνει περισσότερο ενέργεια και χρόνο.

Symmetric cryptography(secret key)	**Characteristics**
BLOCK CIPHER	Χωρίζουν το μήνυμα σε κομμάτια και κρυπτογραφούν κάθε ένα από τα κομμάτια αυτά χωριστά.
Blowfish	Είναι εύκολος στην υλοποίηση και παρουσιάζει μικρή ταχύτητα εκτέλεσης
DES	Ασθενής αντοχή,μέτρια ανάγκη επεξεργασίας, μέτρια ανάγκη RAM,μικρος αριθμός κλειδιών
IDEA	Χρήση κυκλικών μετατοπίσεων, καθιστά πολύ δύσκολη τη διαδικασία κρυπτανάλυσης
RC2	Βασικό χαρακτηριστικό του είναι οτι υποστηρίζει κλειδιά μεταβλητού μεγέθους
RC5	Ταχύς,μεταβλητό μήκος κλειδιού,απλός,χαμηλή απαίτηση μνήμης,υψηλή ασφάλεια,κατάλληλο για υλοποίηση σε υλικό ή λογισμικό
Triple-DES	Ισχυρή αντοχή,υψηλή ανάγκη επεξεργασίας, υψηλή ανάγκη RAM,ισχυρή ασφάλεια,αργός σε υλοποιήσεις με χρήση λογισμικού
AES	Ισχυρή αντοχή,υψηλή ανάγκη επεξεργασίας, υψηλή ανάγκη RAM ,απλός, ευέλικτος, ρωμαλέος
STREAM CIPHER	Κρυπτογραφούν μία ροή μηνύματος (stream) χωρίς να τη διαχωρίζουν σε τμήματα
ORYX	Χρησιμοποιεί 3 καταχωρητές ολίσθησης, χρησιμοποιεί επίσης έναν πίνακα αναζήτησης, είναι πολύ αδύναμος,

RC4	Είναι ουσιαστικά μια γεννήτρια ψευδο-τυχαίων αριθμών,Απλός αλγόριθμος,Ιδανικός για υλοποιήσεις με λογισμικό,Χρησιμοποιεί 256 bytes μνήμης και μεταβλητές ακεραίων,Αποτελείται από δύο στάδια
SEAL	Software-optimized αλγόριθμος κρυπτογράφησης,Σχεδιάστηκε από τους Coppersmith, Rogaway το 1993,Υλοποιήθηκε με τη χρήση λογισμικού,Σχετικά νέος αλγόριθμοςΑποδοτικός όταν εκτελείται σε 32-bit επεξεργαστές,Επικρατέστερη η έκδοση 2.0 του αλγορίθμου
Asymmetric cryptography(public key)	
Πρωτόκολλο Diffie-Hellman	Επιτρέπει σε δύο μέρη, δίχως προηγούμενη επικοινωνία, να καταλήξουν σε κάποιο μυστικό κλειδί μέσω ενός μη ασφαλούς καναλιού
RSA	Βασίζεται στη δυσκολία παραγοντοποίησης μεγάλων αριθμών . Χρησιμοποιούνται δυο κλειδιά, ένα δημόσιο κατά τη διάρκεια της κρυπτογράφησης και ένα ιδιωτικό για την αποκρυπτογράφηση.
DSA	Μπορεί να χρησιμοποιηθεί από τον παραλήπτη ενός μηνύματος για να βεβαιωθεί ότι το μήνυμα δεν έχει αλλοιωθεί κατά τη μεταφορά, καθώς και να εξακριβώσει την ταυτότητα του εντολέα.
Προτυπο ElGamal	Το κρυπτόγραμμα είναι δύο φορές πιο μεγάλο από το μήνυμα,η ασφάλεια του στηρίζεται στη δυσκολία επίλυσης των προλημάτων DLP
Hash algorgorithms	
MD5	Ασφάλεια,Απλότητα και κατάληψη μικρού χώρου,Ταχύτητα.Δέχεται ως είσοδο ένα μήνυμα αυθαίρετου μήκους και παράγει σύνοψη μήκους 128 bits
SHA-1	Παράγει μια αφομοίωση μηνυμάτων, σχηματίζει έναν δεκαεξαδικό αριθμό, μήκους 40 ψηφίων,είναι η πιο ευρέως χρησιμοποιημένη λειτουργία hash.

Πίνακας 1 : Βασικοί αλγόριθμοι και τα χαρακτηριστικά τους

Αναφορές Κεφάλαιο ΙΙ

[1] http://el.wikipedia.org/wiki/κρυπτογραφία
[2] http://www.image.ntua.gr/meleti172KTP/?q=node/23
[3] ΚΩΝΣΤΑΝΤΙΝΟΥ Η. ΠΑΠΑΔΗΜΗΤΡΙΟΥ διπλωματική εργασία "Διαχείρηση Ασφάλειας και Εμπιστοσύνης σε Περιβάλλοντα Εργασίας με Χρήση Τεχνικών Κρυπτογραφίας Δημοσίου Κλειδιού",2006
[4] http://en.wikipedia.org/wiki/RC4 , http://www.wisdom.weizmann.ac.il/~itsik/RC4/rc4.html , http://www.soterisdemetriou.com/related_rc4_gr.php stream_Seal_RC4_A5
[5] http://el.wikipedia.org/wiki/RSA
[6] http://el.wikipedia.org/wiki/Πρωτόκολλο_Diffie-Hellman
[7] http://el.wikipedia.org/wiki/BF_ElGamal
[8] ΤΣΙΓΓΕΝΟΠΟΥΛΟΥ ΒΕΡΑ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ "Ασφάλεια σε RFID και Smart Cards. Μελέτη των Βασικών Αρχών και των Κρυπτογραφικών Μεθόδων που χρησιμοποιούνται για να Ενισχύσουν τα Επίπεδα Ασφάλειας"
[9] Χρυσούλα Π. Σκλιά ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ Επέκταση περιηγητή για διαχείριση πιστοποιητικών σε τεχνολογία έξυπνων καρτών,2006
[10] http://en.wikipedia.org/wiki/SEAL_(cipher)
[11] http://en.wikipedia.org/wiki/ORYX, http://cs.sjsu.edu/~stamp/crypto/PowerPoint_PDF/6_ORYX.pdf
[12] Δρ. Καλλονιάτης Χρήστος "ΒΑΣΙΚΑ ΘΕΜΑΤΑ ΚΡΥΠΤΟΓΡΑΦΙΑΣ"

[13] International Journal of Engineering and Technology "Comparison of Various Encryption Algorithms and Techniques for Secured Data Communication in Multinode Network ",January, 2012

ΚΕΦΑΛΑΙΟ ΙΙΙ

Υλοποίηση και σύγκριση βασικών αλγόριθμων κρυπτογράφησης

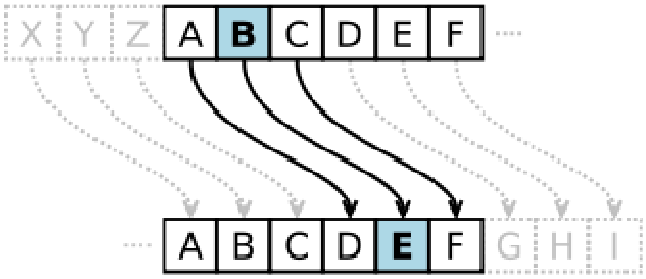
3.1 Βήματα λειτουργίας κρυπτογραφησης αλγορίθμων

Για τα περισσότερα παραδείγματα και βήματα πέρνουμε ως λέξη προς κρυπτογράφηση το κείμενο : **university patras** η οποία είναι 128bit.

3.1.1 Αλγόριθμος Caesar [1]

Είναι κώδικας αντικατάστασης στον οποίο κάθε γράμμα του κειμένου αντικαθίσταται από κάποιο άλλο γράμμα με σταθερή απόσταση κάθε φορά στο αλφάβητο. Για

παράδειγμα, με μετατόπιση 3, το A θα αντικαθιστούνταν από το Δ, το B από το E, και ούτω καθεξής.



Οπότε για τον συγκεκριμένο αλγόριθμο και λέξη έχουμε τα παρακάτω βήματα.

Α. Έχουμε τη λέξη **university patras.** Υπολογίζουμε και βρίσκουμε το 3^{ο} επόμενο γράμμα κατα το αγγλικό αλφάβητο για κάθε γράμμα των λέξεων ξεχωριστά.
Β. Άρα έχουμε το κρυπτογαφημένο αποτέλεσμα **xqlyhuvlwb sdwudv.**

3.1.2. Αλγόριθμος Vigenere [2]

Η κρυπτογράφηση Vigenère αποτελείται από πολλούς αλγόριθμους κρυπτογράφησης του Καίσαρα σε ακολουθία με διαφορετικές τιμές μετατόπισης. Για την κρυπτογράφηση, ένας πίνακας του αλφάβητου μπορεί να χρησιμοποιηθεί, ως tabula Recta, Vigenère square, ή Vigenère table. Αποτελείται από το αλφάβητο, που αναγράφεται 26 φορές σε διαφορετικές γραμμές, κάθε αλφάβητο μετατοπίζεται κυκλικά προς τα αριστερά σε σχέση με την προηγούμενη αλφάβητο, που αντιστοιχούν στους 26 πιθανούς αλγόριθμους κρυπτογράφησης του Καίσαρα. Σε διάφορα σημεία κατά τη διαδικασία κρυπτογράφησης, η κρυπτογράφηση χρησιμοποιεί διαφορετικό αλφάβητο σε κάθε μια από τις σειρές. Το αλφάβητο που χρησιμοποιείται σε κάθε σημείο εξαρτάται από μια επαναλαμβανόμενη λέξη-κλειδί.

	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

A. Έχουμε το κείμενο **UNIVERSITYPATRAS.** Ως λέξη κλειδί πέρνουμε τη λέξη **LEMON** και την επαναλαμβάνουμε μέχρι να ταιριάζει με το μήκος του απλού κειμένου, δηλαδή **LEMONLEMONLEMONL.**

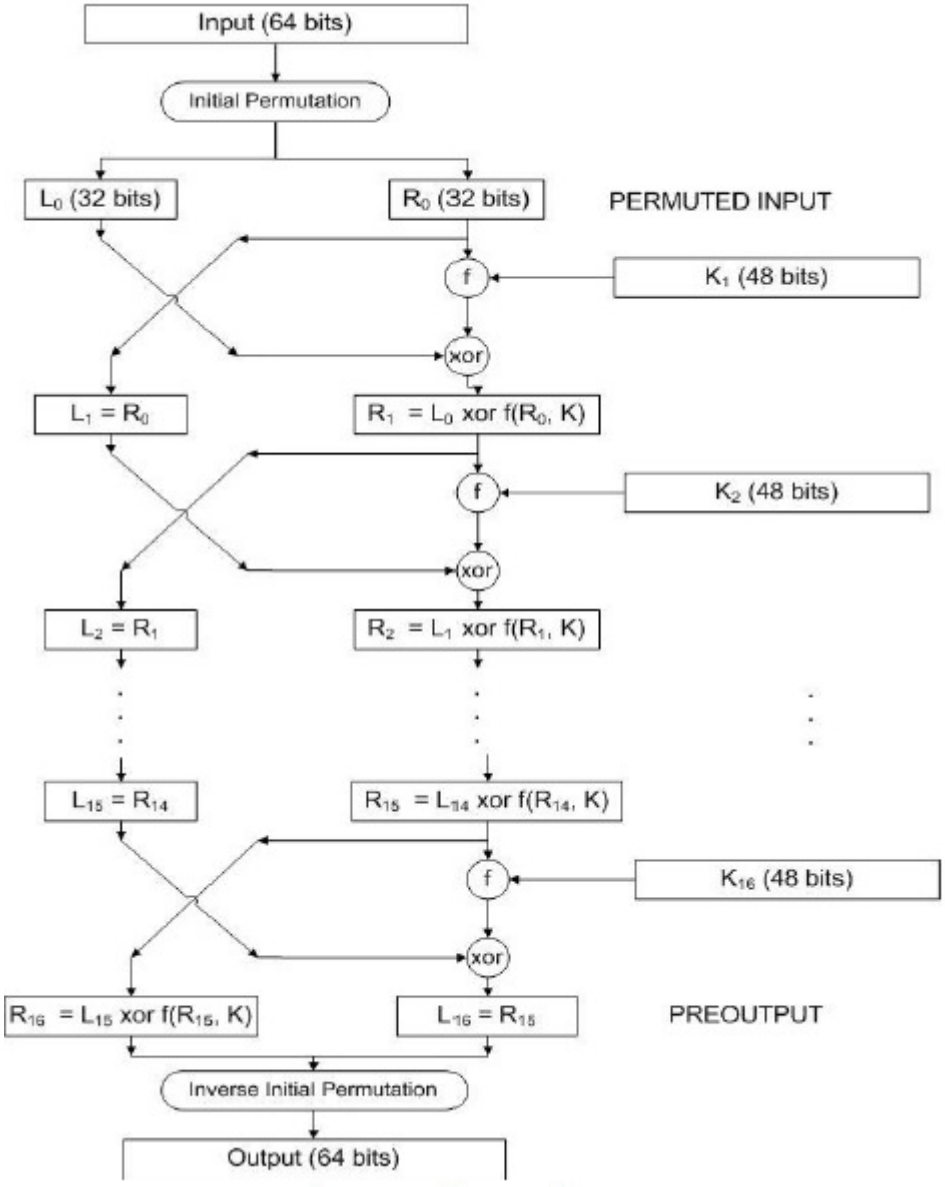
B. Κάθε γραμμή ξεκινά με ένα πλήκτρο του γράμματος. Το υπόλοιπο της σειράς κατέχει τα γράμματα A έως το Z(σε σειρά μετατόπισης). Παρά το γεγονός ότι υπάρχουν 26 βασικές, εμείς θα χρησιμοποιήσουμε τόσα κλειδιά (διαφορετικά αλφάβητα) όσα και τα μοναδικά γράμματα που υπάρχουν στο κλειδί σειρά, εδώ μόλις 5 κλειδιά {L, E, M, O, N}. Για διαδοχικά γράμματα από ένα μήνυμα, θα πάρουμε τα διαδοχικά γράμματα από τη σειρά κλειδί, και η κρυπτογράφηση για κάθε μήνυμα θα γίνει χρησιμοποιώντας το αντίστοιχο πλήκτρο της γραμμής. Επιλέγουμε το επόμενο γράμμα του κλειδιού, πάμε κατά μήκος της στήλης αυτό το γράμμα για να βρούμε την κεφαλίδα της στήλης που ταιριάζει με το μήνυμα του χαρακτήρα , το γράμμα στη διασταύρωση των [key-row ,msg-col] είναι το κρυπτογραφημένο γράμμα.

C. Άρα βρίσκοντας με τη βοήθεια του πίνακα-αλφάβητου τον συνδυασμό, έχουμε ως αποτέλεσμα το κρυπτογραφημένο κείμενο:
FRUJRCWUHLAEFFND .

3.1.3 Αλγόριθμος DES [3]

A . Αρχικά σαν είσοδο έχουμε καθαρό κείμενο σε block των 64 bit (8 bytes),κλειδί των 56 bit συν 1 bit ισοτιμίας για κάθε 7 bit (Σύνολο 64 bit)

B.Ένα τμήμα – block για να κρυπτογραφηθεί πρέπει να περάσει από μία αρχική μετάθεση (Initial Permutation - IP), μετά από ένα πολύπλοκο υπολογισμό που εξαρτάται από το κλειδί, και τελικά από μία τελική μετάθεση (Final Permutation – IP-1) που είναι αντίστροφη της αρχικής. Ο ενδιάμεσος υπολογισμός χρησιμοποιεί μία συνάρτηση f , που ονομάζεται cipher function, και τη συνάρτηση δημιουργίας κλειδιού (Key Schedule - KS) Εν συνεχεία παρουσιάζεται το σχήμα (ΕΙΚΟΝΑ 1) που απεικονίζει τη διαδικασία κρυπτογράφησης (encryption) του DES.



ΕΙΚΟΝΑ 1. Διαδικασία κρυπτογράφησης DES

Στην αριστερή πλευρά της εικόνας 1 παρουσιάζονται τα τρία στάδια της επεξεργασίας του αρχικού κειμένου. Στην αρχή, το κείμενο των 64-bit ακολουθεί την αρχική μετάθεση (IP) στα πλαίσια του οποίου τα bits αναδιατάσσονται για να παραχθεί η μετασχηματισμένη είσοδος. Γίνεται αντιμετάθεση σύμφωνα με τον παρακάτω πίνακα 1, όπου οι είσοδοι στον πίνακα δείχνουν την νέα αναδιάταξη των bits από την αρχική. Το 58ο bit της εισόδου γίνεται το 1ο του IP. Το 50ο γίνεται 2ο κτλ.

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

ΠΙΝΑΚΑΣ.1 IP(Αρχική μετάθεση μηνύματος)

C.Ακολουθεί ένα στάδιο που αποτελείται από 16 επαναλήψεις της ίδιας λειτουργίας. Η έξοδος της τελευταίας επανάληψης, δηλαδή της δέκατης έκτης, αποτελείται από 64-bit που αποτελούν συνάρτηση του αρχικού κειμένου και του κλειδιού. Το αριστερό (L) μισό τμήμα και το δεξί (R) μισό τμήμα της εξόδου αντιμετατίθενται, ώστε να παραχθεί η αρχική έξοδος (PREOUTPUT).

D. Η τιμή αυτή τροποποιείται με βάση την αντίστροφη μετάθεση (IP-1) ώστε να παραχθεί το κρυπτογράφημα των 64-bit.

3.1.4 Αλγόριθμος AES (CBC) Rijandel [4]

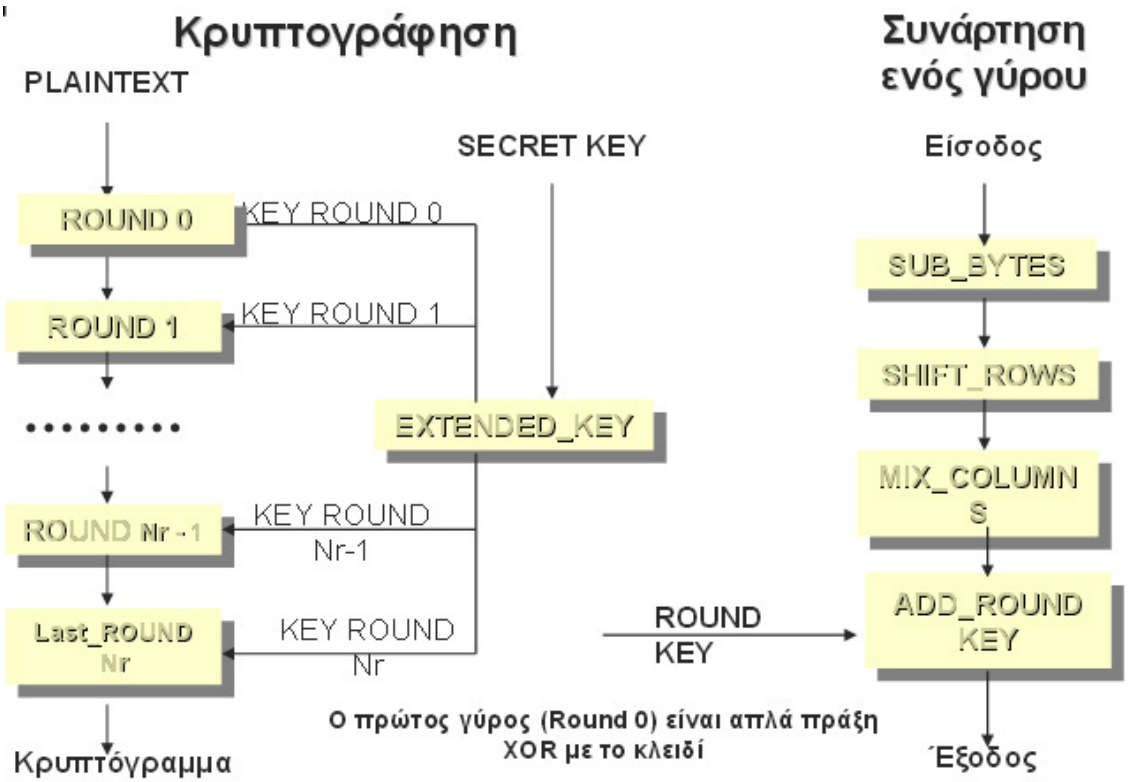
A. Έχουμε μήκος κλειδιού 128bits άρα χρειάζονται 10 γύροι. Κάθε γύρος έχει 4 βήματα:
-Αντικατάσταση byte (Byte substitution) – χρήση s-boxes με καλά χαρακτηριστικά
-Ολίσθηση (Shift row)
-Συνδυασμός πολλών bit (Mix Column)
-Πρόσθεση (XOR) του κλειδιού

B. Για μήκος 128 bits: τοκάθε μπλοκ θεωρείται σαν ένας 4 x 4 πίνακας, όπου κάθε στοιχείο του πίνακα είναι 1 byte.
Ο αλγόριθμος αποτελείται από έναν αρχικό γύρο, και άλλους r - 1 τυπικούς γύρους (όπου το r είναι είτε 10 είτε 12 είτε14 αναλόγως τα μήκη των μπλοκ), καθώς επίσης και από έναν τελευταίο γύρο.

C. Στο επόμενο βήμα βρίσκουμε το πλήθος γύρων που χρειάζονται στον πίνακα μπλόκ και τον πίνακα κλειδιού.Επίσης το σχηματικό διάγραμμα των γύρων εκτέλεσης.

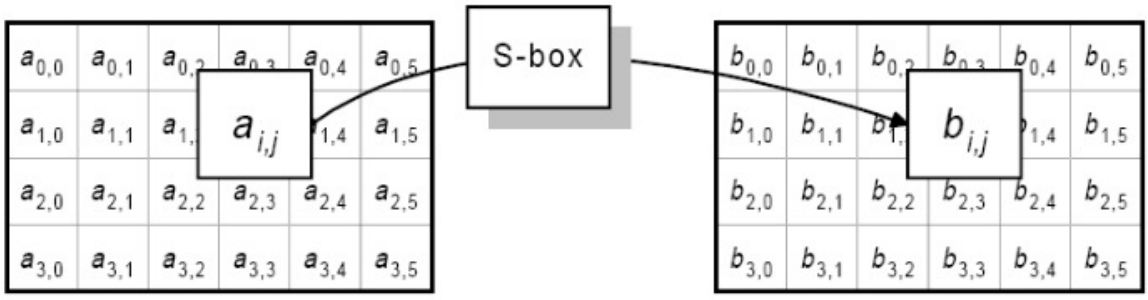
	Nb=4	Nb=6	Nb=8
Nk=4	10	12	14
Nk=6	12	12	14
Nk=8	14	14	14

Πίνακας μπλοκ και κλειδιού.

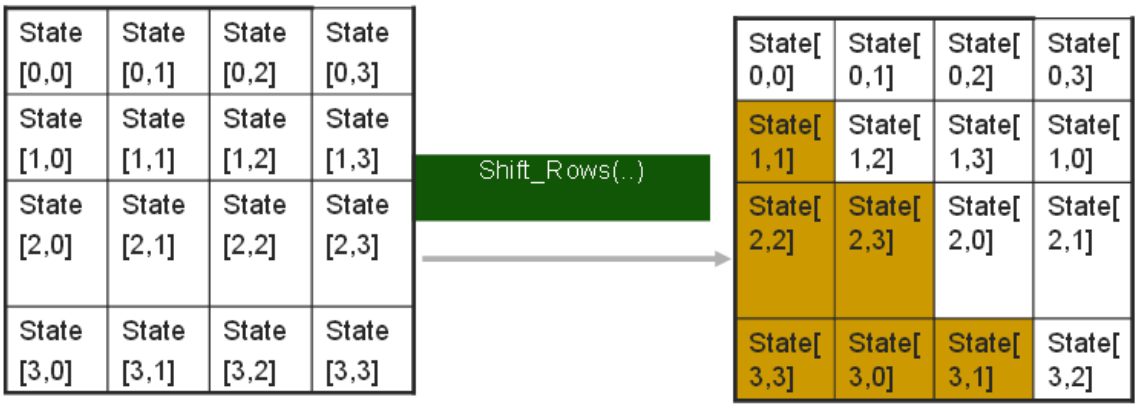


Σχηματικό διάγραμμα AES.

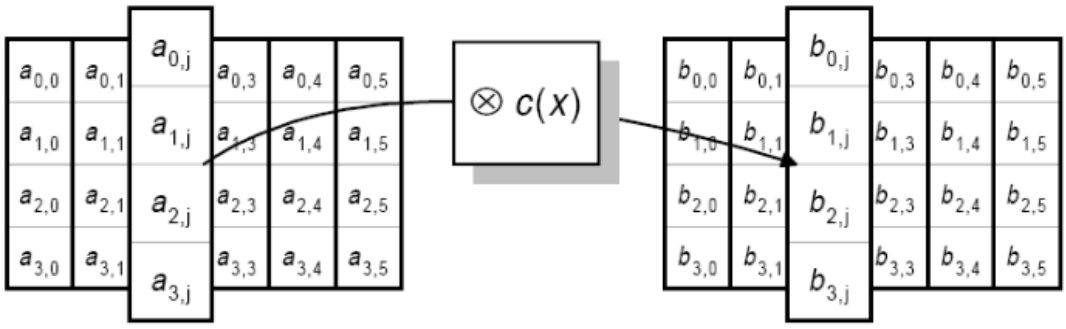
D. Στη συνέχεια κάθε byte μετατρέπεται σε ενα άλλο , μέσω μιας γραμμικής συνάρτησης. Παρακάτω βλέπουμε τον τρόπο μετατροπής S-Box (Sub_bytes):



E. Έπειτα έχουμε την ολίσθηση γραμμών (shift_rows):



F. Τέλος γίνεται το ανακάτωμα των στηλών (mixcolumn) , ώστε να λάβουμε το κρυπτογραφημένο μήνυμα.



Πολλαπλασιασμός με έναν κατάλληλο (πάντα σταθερό και συγκεκριμένο) πίνακα C Διαστάσεων 4x4 (όπου κάθε στοιχείο του C είναι 1 Byte).Η MixColumn πράξη είναι αυτή η οποία δεν συντελείται στον τελευταίο γύρο του AES. (Κατά τα άλλα, ο τελευταίος γύρος είναι ίδιος με τους υπόλοιπους)

3.1.5 Αλγόριθμος RSA [4]

A. Στον αλγόριθμο RSA έχουμε ως πρωτο βήμα τις παρακάτω παραμέτρους:

1. Επιλέγονται δύο μεγάλοι πρώτοι αριθμοί, p και q (συνήθως πολύ μεγαλύτεροι από 10^{100})
2. Υπολογίζεται $n = p \times q$ και $z = (p-1) \times (q-1)$. Ο n ονομάζεται *υπόλοιπο RSA (RSA modulus)*
3. Επιλέγεται ένας πρώτος αριθμός ως προς τον z ο οποίος ονομάζεται e
4. Υπολογίζεται ο d έτσι ώστε $d \times e = 1 \bmod z$
5. Το δημόσιο κλειδί αποτελείται από το ζευγάρι (e,n) και το ιδιωτικό κλειδί από το ζευγάρι (d,n).

Τα βήματα που ακολουθούνται για την κρυπτογράφηση ενός κειμένου m περιγράφονται παρακάτω

B. Το κείμενο το οποίο θα κρυπτογραφηθεί (που θεωρείται ως συρμός bit) διαιρείται σε μπλοκ, έτσι ώστε κάθε μήνυμα κειμένου, m, να πέφτει στο διάστημα $0 \le m < n$. Αυτό μπορεί να γίνει με ομαδοποίηση του κειμένου σε μπλοκ των k bit, όπου το k είναι ο μεγαλύτερος ακέραιος για τον οποίο η σχέση $2^k < n$ εί

C. Υπολογίζεται το $c = m^e \pmod n$ όπου το c είναι το κρυπτογραφημένο κείμενο.

Παράδειγμα

Για δοθέντα a,b με α>= b, εύρεση του g=gcd(a,b) και ακεραίων x,y με την ιδιότητα a.x+b.y=g [στον RSA: a= φ(N),b=e και επίσης g=1. Στο d θα αποδοθεί η τιμή του y,δλδ, φ(N).x+e.d=1, οπότε e.d=1mod φ(N)]

- **ΒΗΜΑ1:** Αν b=0,τότε: c=a,x=1,y=0, δλδ, a=a.1+0.0 διαφορετικά
- **ΒΗΜΑ2:** x2=1,x1=0,y2=0,y1=1
- **ΒΗΜΑ3:** Για όσο ισχύει b>0 q=[a/b], r=a-q .b , x=x2-q .x1 , y=y2-q .y1
  a=b, b=r, x2=x1, x1=x, y2=y1, y1=y
- **ΒΗΜΑ4:**g=a, x=x2, y=y2 και επέστρεψε (g,x,y) Τέλος

Έστω p=37 q=73 , N= p . q=2701 Φ(N)=(p-1) (q-1)=2592 κι e=77

Q	**r**	**X**	**Y**	**a=Φ(N)**	**b=e**	**X2**	**X1**	**Y2**	**Y1**
-	-	-	-	2592	77	1	0	0	1
33	51	1	-33	77	51	0	1	1	-33
1	26	-1	34	51	26	1	-1	-33	34
1	25	2	-67	26	25	-1	2	34	-67
1	1	-3	101	25	1	2	-3	-67	101
25	0	77	-2592	1	0	-3	77	101	-2592

Άρα x=x2=-3 , y=y2=101 =d , οπότε 2592 .(-3)+77. 101=1

Για το μήνυμα **university patras** μετατρέπουμε κάθε γράμμα σε αριθμό, με βάση τη θέση του στο αλφάβητο.

U N I V E R S I T Y P A T R A S

20 13 08 21 04 17 18 08 19 24 15 00 19 17 00 18

Σπάμε το μήνυμα σε blocks των δύο γραμμάτων(το δύο αυτό είναι τυχαίο-αναλόγως την υλοποίηση του RSA αλλάζει)

2013 0821 0417 1808 1924 1500 1917 0018 (τα ονομάζουμε P1,P2,P3,P4,P5,P6,P7,P8) και υπολογίζουμε το $Ci = Pi^{77} (mod\ 2701)$ για κάθε block

$C1 = 2013^{77} (mod\ 2701)$ = 135
$C2 = 821^{77} (mod\ 2701)$= 2007
$C3 = 417^{77} (mod\ 2701)$= 322
$C4 = 1808^{77} (mod\ 2701)$= 723
$C5 = 1924^{77} (mod\ 2701)$= 407
$C6 = 1500^{77} (mod\ 2701)$= 2608

$C7 = 1917^{77} (mod\ 2701)$= 1545
$C8 = 18^{77} (mod\ 2701)$= 1569
Άρα στέλνουμε τους αριθμούς 135,2007,322,723,407,2608,1545,1569

3.1.6 Αλγόριθμος Hill [4]

Η κρυπτογράφηση αποτελεί επέκταση του γραμμικού αλγορίθμου. Το μήνυμα κωδικοποιείται ανα block L στοιχείων(Lτυχαίος ακέραιος) με βάση L γραμμικές εξισώσεις(αντί για μία που είναι στο γραμμικό κρυπταλγόριθμο).

A. Συγκεκριμένα: c=Km mod n όπου K πίνακας διαστάσεων LxL(το κλειδί),όπου n το πλήθος των γραμμάτων της αλφαβήτου,κι αφού είμαστε στο αγγλικό αλφάβητο n=26,παρακάτω θα δείτε ένα παράδειγμα αλγορίθμου Hill σε μπλοκ μήκους 2.

B. Έστω ότι προτείνω για K=$\begin{pmatrix} 2 & 3 \\ 1 & 7 \end{pmatrix}$,ένας πίνακας για να μπορεί να χρησιμοποιηθεί για την κρυπτογράφηση Hill θα πρέπει ο gcd(a,n)=1 Βρίσκω την ορίζουσα K= =>2x7-1x3=14-3=11=a και gcd(11,26)=1 άρα μπορώ να τον χρησιμοποιήσω.

C. Για την κρυπτογράφηση της λέξης **university patras** ξεκινάω παίρνοντας:

το ζευγάρι γραμμάτων **"UN"** ,το γράμμα U αντιστοιχεί στο νούμερο 20 κι το N στο 13.

$\begin{pmatrix} 2 & 3 \\ 1 & 7 \end{pmatrix} \begin{pmatrix} 20 \\ 13 \end{pmatrix} = \begin{pmatrix} 79 \\ 111 \end{pmatrix} \begin{pmatrix} 1 \\ 7 \end{pmatrix}$mod 26 άρα το ζευγάρι **UN** γίνεται **BH**

το ζευγάρι γραμμάτων **"IV"** , το γράμμα I αντιστοιχεί στο νούμερο 8 κι το V στο 21.

$\begin{pmatrix} 2 & 3 \\ 1 & 7 \end{pmatrix} \begin{pmatrix} 8 \\ 21 \end{pmatrix} = \begin{pmatrix} 79 \\ 155 \end{pmatrix} \begin{pmatrix} 1 \\ 25 \end{pmatrix}$mod 26 άρα το ζευγάρι **UN** γίνεται **BZ**

το ζευγάρι γραμμάτων **"ER"** ,το γράμμα E αντιστοιχεί στο νούμερο 4 κι το R στο 17.

$\begin{pmatrix} 2 & 3 \\ 1 & 7 \end{pmatrix} \begin{pmatrix} 4 \\ 17 \end{pmatrix} = \begin{pmatrix} 59 \\ 53 \end{pmatrix} \begin{pmatrix} 7 \\ 1 \end{pmatrix}$mod 26 άρα το ζευγάρι **ER** γίνεται **HB**

το ζευγάρι γραμμάτων **"SI"** , το γράμμα S αντιστοιχεί στο νούμερο 18 κι το I στο 8.

$\begin{pmatrix} 2 & 3 \\ 1 & 7 \end{pmatrix} \begin{pmatrix} 18 \\ 8 \end{pmatrix} = \begin{pmatrix} 60 \\ 74 \end{pmatrix} \begin{pmatrix} 8 \\ 22 \end{pmatrix}$mod 26 άρα το ζευγάρι **SI** γίνεται **IW**

το ζευγάρι γραμμάτων **"TY"** , το γράμμα T αντιστοιχεί στο νούμερο 19 κι το Y στο 24.

$\begin{pmatrix} 2 & 3 \\ 1 & 7 \end{pmatrix} \begin{pmatrix} 19 \\ 24 \end{pmatrix} = \begin{pmatrix} 110 \\ 187 \end{pmatrix} \begin{pmatrix} 6 \\ 5 \end{pmatrix}$mod 26 άρα το ζευγάρι **TY** γίνεται **GF**

.

το ζευγάρι γραμμάτων **"PA "** , το γράμμα P αντιστοιχεί στο νούμερο 15 κι το A στο 0.

$\begin{pmatrix} 2 & 3 \\ 1 & 7 \end{pmatrix} \begin{pmatrix} 15 \\ 0 \end{pmatrix} = \begin{pmatrix} 30 \\ 15 \end{pmatrix} \begin{pmatrix} 4 \\ 15 \end{pmatrix}$mod 26 άρα το ζευγάρι **PA** γίνεται **EP**

το ζευγάρι γραμμάτων **"TR "** , το γράμμα T αντιστοιχεί στο νούμερο 19 κι το R στο 17

$\begin{pmatrix}2 & 3\\ 1 & 7\end{pmatrix}\begin{pmatrix}19\\17\end{pmatrix}=\begin{pmatrix}89\\138\end{pmatrix}\begin{pmatrix}11\\8\end{pmatrix}$mod 26 άρα το ζευγάρι **TR** γίνεται **LI**

.

Και τέλος,το ζευγάρι γραμμάτων **"AS "** , το γράμμα Α αντιστοιχεί στο νούμερο 0 κι το S στο 18

$\begin{pmatrix}2 & 3\\ 1 & 7\end{pmatrix}\begin{pmatrix}0\\18\end{pmatrix}=\begin{pmatrix}54\\126\end{pmatrix}\begin{pmatrix}2\\22\end{pmatrix}$mod 26 άρα το ζευγάρι **AS** γίνεται **CW**

Άρα η λέξη **university patras** γίνεται **bhbzhbiwgfeplicw** σε κρυπτοφραφημένη μορφή.

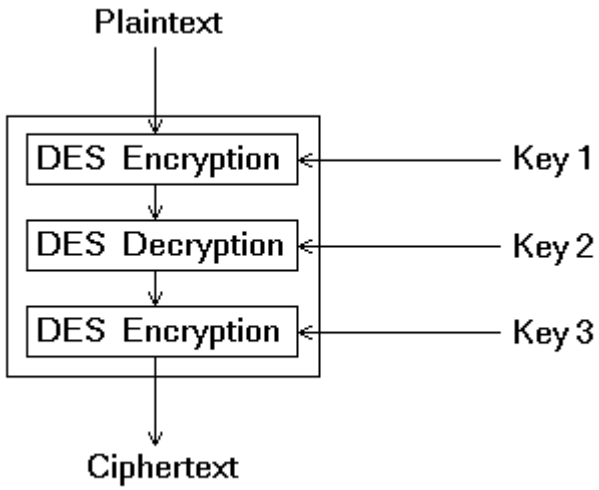
3.1.7 Αλγόριθμος Triple DES [5]

Ο αλγόριθμος Triple DES λειτουργεί με τον τρόπο του απλού DES , με τη διαφορά πως ο DES χρησιμοποιεί ενα κλειδί μήκου 56 bit . ενώ ο Triple 3 κλειδιά ίδιου μήκους, ώστε να είναι πιο ανθεκτικός σε επιθέσεις τύπου brutal force.

A. Ο triple-DES μπορεί να εφαρμοστεί και με δύο διαφορετικά κλειδιά αντί για τρία. Στην γενική περίπτωση, το συνολικό εύρος του κλειδιού είναι 2^112.
Ο αλγόριθμος ακολουθεί τη διαδοχή: κρυπτογράφηση, αποκρυπτογράφηση, κρυπτογράφηση (EDE – encryption – decryption - encryption) :

C=EK3[DK2[EK1[P]]]
όπου:
C = κρυπτογράφημα
P = αρχικό κείμενο
EK[X] = κρυπτογράφηση του X με χρήση του κλειδιού K
DK[Y] = αποκρυπτογράφηση του X με χρήση του κλειδιού K

B. Βήματα εισαγωγής κειμένου και έξοδος του , μετά την κρυπτογράφηση.
Αρχικά το κέιμενό μας κρυπτογραφείται με το πρώτο κλειδί , στη συνέχεια αποκρυπτογραφείται με το δεύτερο κλειδί και τέλος κρυπτογραφείται και πάλι με το τρίτο κλειδί ωστέ να πάρει την τελική του μορφή.



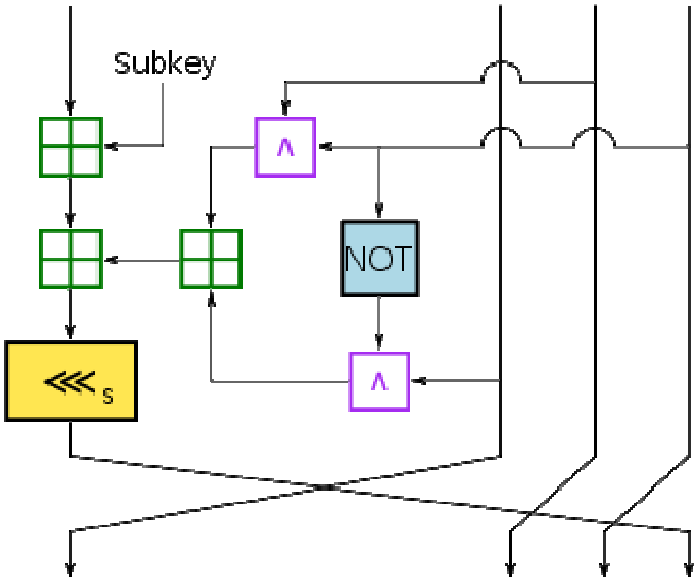
Το πρότυπο καθορίζει τις ακόλουθες επιλογές για το κλειδί για τη δέσμη (K1, K2, K3):

- K1, K2 και K3 ανεξάρτητα κλειδιά.
- K1 και K2 ανεξάρτητα κλειδιά και K3 = K1.
- K1 = K2 = K3

Αξίζει να σημειωθεί ότι η ύπαρξη της αποκρυπτογράφησης στο δεύτερο στάδιο της κρυπτογράφησης TDES δεν παρουσιάζει κάποια κρυπτογραφική χρησιμότητα, απλώς επιτρέπει στους χρήστες του TDES να αποκρυπτογραφήσουν τα στοιχεία που κρυπτογραφούνται από τους χρήστες του απλού DES: C=EK1[DK1[EK1[P]]]

3.1.8 Αλγόριθμος RC2 [6]

Ο RC2 είναι ένας block-cipher 64-bit, με μεταβλητό μέγεθος κλειδιού. Οι 18 γύροι του είναι διευθετημένοι ως ενα source-heavy Feistel δίκτυο, με 16 γύρους ενός τύπου (MIXING) και διακόπτονται από δύο γύρους του άλλου τύπου (mashing). Ένας γύρος Mixing αποτελείται από τέσσερις εφαρμογές του μετασχηματισμού MIX, όπως φαίνεται στο διάγραμμα.



Εικόνα του MIX μετασχηματισμού της RC2.

3.1.9 Αλγόριθμος RC4 [7]

Είναι ένα ιδιωτικός κρυπτογραφικός αλγόριθμος που ανήκει στην RSA DSI. Έχει μεταβλητό μέγεθος κλειδιού, είναι ενας byte-oriented κρυπτογραφικος αλγοριθμος ροης και χρησιμοποιειται ευρυτατα (web SSL/TLS, wireless WEP/WPA) .
Βήματα λειτουργίας του αλγορίθμου:

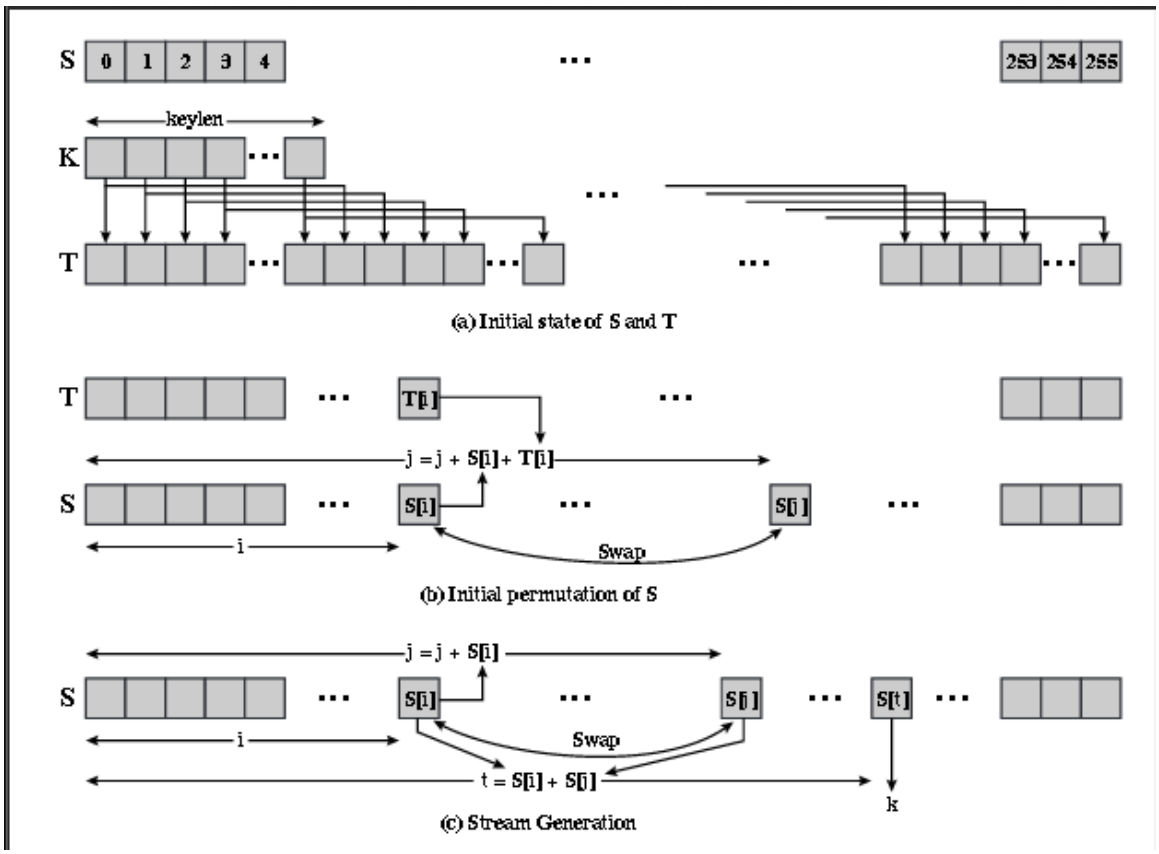
A. Αρχιζει με ενα array S με αριθμους: 0..255
• Χρησιμοποιουμε το κλειδι για να ανακατεψουμε καλα
• Το S σχηματιζει την εσωτερικη κατασταση του αλγορίθμου

```
for i = 0 to 255 do
S[i] = i
T[i] = K[i mod keylen])
j = 0
for i = 0 to 255 do
j = (j + S[i] + T[i]) (mod 256)
swap (S[i], S[j])
```

B. Η κρυπτογραφηση συνεχιζεται ανακατευοντας τις τιμες του array. Το αθροισμα του ανακατεμενου ζευγους επιλεγει την τιμη του "stream key" απο τη μεταθεση.

C. Κανουμε XOR το S[t] με το επομενο byte του μημυματος για να κρυπτογραφησουμε
/αποκρυπτογραφησουμε

```
i = j = 0
for each message byte M
i
i = (i + 1) (mod 256)
j = (j + S[i]) (mod 256)
swap(S[i], S[j])
t = (S[i] + S[j]) (mod 256)
C
i
= M
i
XOR S[t]
```



Σχηματικό διάγραμμα κρυπτογράφησης RC4.

3.1.10 Αλγόριθμος Playfair [4]

Ένας 5x5 πίνακας συμπληρώνεται με τη λέξη-κλειδί(δύο ίδια γράμματα δεν εμφανίζονται δυο φορές)και οι υπόλοιπες θέσεις του πίνακα συμπληρώνονται από τα εναπομείναντα γράμματα του αλφάβητου.Στον παρακάτω πίνακα,κλειδί είναι η λέξη "CRYPTOGRAPHY"(επιλέξαμε την περίπτωση όπου παραλείπουμε το Q από τον πίνακα,που είναι η συνηθέστερη περίπτωση.)

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

A. Για την κρυπτογράφηση χωρίζουμε το μήνυμα σε ζεύγη γραμμάτων.Κάθε ένα ζεύγος το κρυπτογραφούμε με βάση τους ακόλουθους κανόνες:

- Αν υπάρχει ζευγάρι με δύο ίδια γράμματα,τότε ανάμεσά τους προστίθεται ένα X.
- Αν τα δύο γράμματα του ζεύγους εμφανίζονται στην ίδια γραμμή στον πίνακα,τότε το καθένα αντικαθιστάται από το δεξιότερο του(αν κάποιο από αυτά είναι το τελευταίο στη γραμμή, τότε αντικαθιστάται από το πρώτο της γραμμής.
- Αν τα δύο γράμματα του ζεύγους εμφανίζονται στην ίδια στήλη στον πίνακα,τότε το καθένα αντικαθίσταται από αυτόπου βρίσκεται αμέσως κάτω του(αν κάποιο από αυτά είναι το τελευταίο στη στήλη τότε αντικαθιστάται από το πρώτο της στήλης.)
- Τελος,αν δεν βρίσκονται ούτε στην ίδια γραμμή ούτε στην ίδια στήλη,τότε φανταζόμαστε το νοητό ορθογώνιο που ορίζουν τα δύο γράμματα και τα αντικαθιστούμε από τα άλλα δύο γράμματα που αντιστιχούν στις γωνίες του ορθογωνίου(έχει σημασία η σειρά-κάθε γράμμα(γωνία του ορθογωνίου)θα αντικατασταθεί από εκείνο το γράμμα(γωνία)που βρίσκεται στην ίδια γραμμή.

B. Στη συνέχεια πάμε να κρυπτογραφήσουμε τη λέξη «**university patras»,** σπάμε το μήνυμα σε δυάδες δηλαδή UN,IV,ER,SI,TY PA,TR,AS.

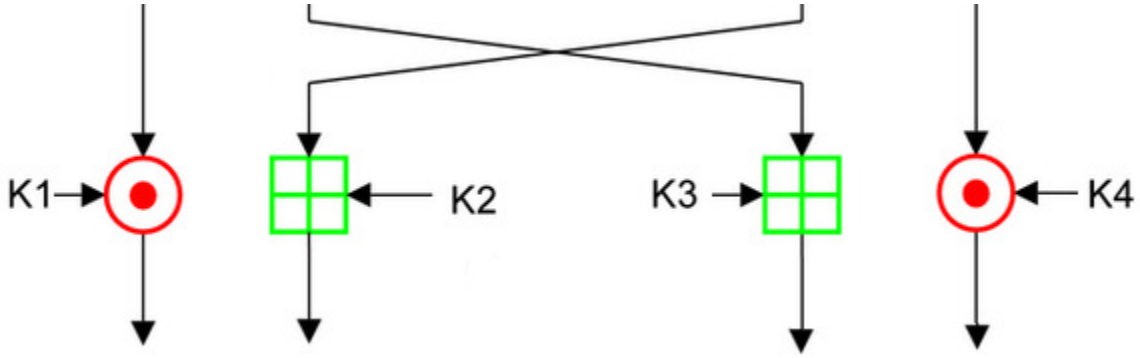
Η κρυπτογράφηση γίνεται ως εξής:

1. το UN σχηματίζουν ορθογώνιο στον πίνακα,άρα κρυπτογραφούνται στα XK
2. το IV σχηματίζουν επίσης ορθογώνιο ,άρα κρυπτογραφούνται στα EX
3. το ER εμφανίζονται στην ίδια στήλη,αρα κρυπτογραφούνται στα LG
4. το SI σχηματίζουν ορθογώνιο ,άρα κρυπτογραφούνται στα NJ
5. το TY εμφανίζονται στην ίδια γραμμή,άρα κρυπτογραφούνται στα CP
6. το PA σχηματίζουν ορθογώνιο ,άρα κρυπτογραφούνται στα YH
7. το TR εμφανίζονται στην ίδια γραμμή,άρα κρυπτογραφούνται στα CY
8. το AS σχηματίζουν ορθογώνιο ,άρα κρυπτογραφούνται στα BM

Άρα η λέξη **university patras** γίνεται **xkexlgnjcp yhcybm** σε κρυπτοφραφημένη μορφή.

3.1.11 Αλγόριθμος IDEA [8]

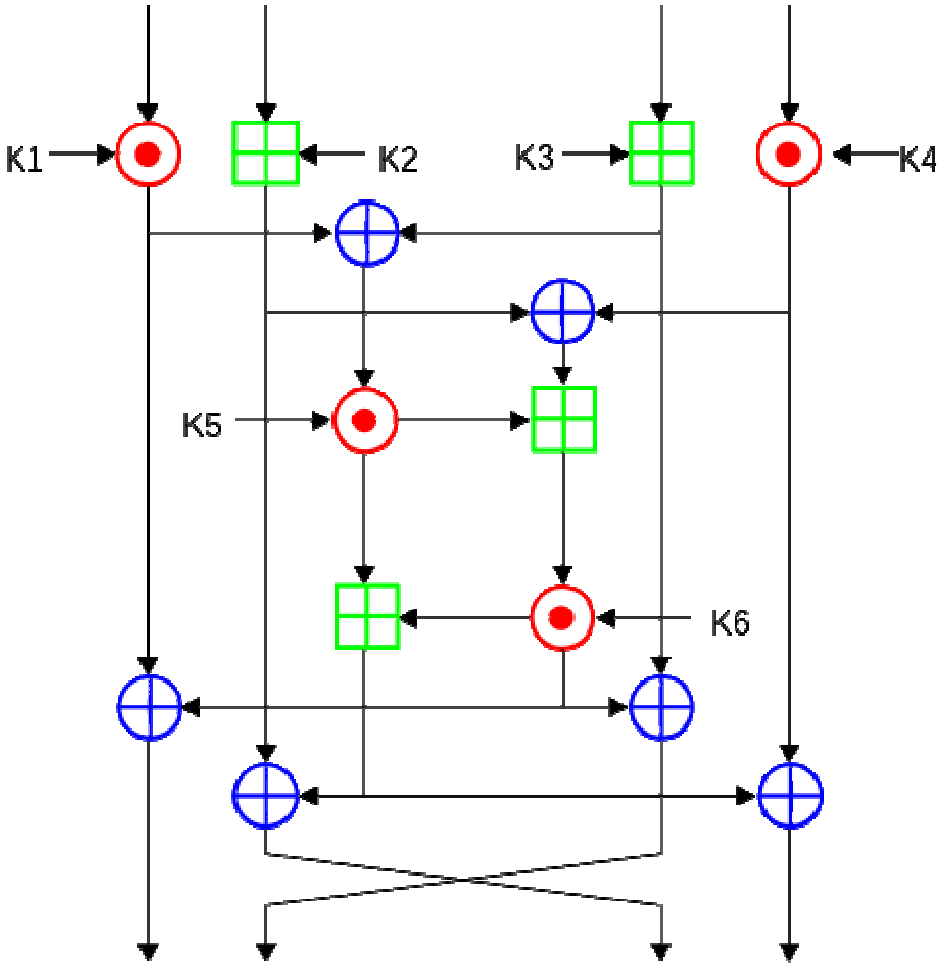
Ο IDEA λειτουργεί με μπλοκ 64-bit χρησιμοποιώντας ένα κλειδί 128-bit, και αποτελείται από μια σειρά από οκτώ πανομοιότυπες μεταμορφώσεις (ενας γύρος,) και μιας μεταμόρφωσης εξόδου (μισός γύρος,εικόνα). Η διαδικασία για την κρυπτογράφηση και την αποκρυπτογράφηση είναι παρόμοια. Ο IDEA αντλεί μεγάλο μέρος της ασφάλειάς του μέσω της παρεμβολής εργασιών από διάφορες ομάδες - modular πρόσθεσης και πολλαπλασιασμό, και bitwise eXclusive Ή (XOR) - τα οποία είναι αλγεβρικά "ασυμβίβαστα" με κάποια έννοια.



Εικόνα γύρων μεταμορφώσεων.

A. Η συνολική δομή του IDEA ακολουθεί το σχήμα Lai-Massey. Η πράξη XOR χρησιμοποιείται τόσο για αφαίρεση οσο και την προσθήκη.Ο IDEA χρησιμοποιεί μια λειτουργία μισού γύρου που εξαρτάται απο το κλειδί. Για να συνεργαστεί με τις λέξεις 16 bit (δηλαδή τέσσερις εισόδους αντί για δύο για το μέγεθος του μπλοκ 64 bit), ο IDEA χρησιμοποιεί το σύστημα Lai-Massey δύο φορές παράλληλα, με τις δύο παράλληλες λειτουργίες γύρων που είναι συνυφασμένες με την άλλη. Για να εξασφαλιστεί επαρκής διάχυση, δύο από τα υπο-μπλοκ αντιστρέφονται μετά από κάθε γύρο.

B. Κάθε γύρος χρησιμοποιεί έξι 16-bit υπο-κλειδιά, ενώ ο μισός γύρος χρησιμοποιεί τέσσερις, συνολικά 52 για 8.5 γύρους. Τα πρώτα οκτώ sub-keys εξάγονται απευθείας από το κλειδί, με K1 από τον πρώτο γύρο να είναι τα χαμηλότερα δεκαέξι bits. Περαιτέρω ομάδες των οκτώ κλειδιών δημιουργούνται περιστρέφοντας το κύριο κλειδί αριστερά 25 bits ανάμεσα σε κάθε ομάδα των οκτώ. Αυτό σημαίνει ότι περιστρέφεται λιγότερο από μία φορά ανά γύρο, κατά μέσο όρο, για ένα σύνολο έξι περιστροφών.



Διάγραμμα λειτουργίας κρυπτογράφησης IDEA.

3.1.12 Αλγόριθμος ADFGVX [9]

Πήρε το όνομά του από τα έξι πιθανλα γράμματα που χρησιμοποιούνται στο κρυπτογράφημα: A, D, F, G, V και X. Τα γράμματα αυτά επιλέχθηκαν σκόπιμα επειδή ακούγονται πολύ διαφορετικά το ενα με το άλλο, όταν μεταδίδονται μέσω του κώδικα Μορς. Η πρόθεση ήταν να μειωθεί η πιθανότητα λάθους του χειριστή.

A. Σε αυτή τη περίπτωση έχουμε το κείμενο «universitypatras» . Αρχικά γεμίζουμε εναν πίνακα διαστάσεων 5x5 με μια ανακατεμένη αλφάβητο.Τα γράμματα i & j έχουν συνυαστεί , ώστε να γίνει ο πίνακας 5x5.

	A	**D**	**F**	**G**	**X**
A	B	t	A	l	p
D	D	h	O	z	k
F	Q	f	v	s	n
G	G	j	c	u	x
X	M	r	e	w	y

B. Με τη βοήθεια του παραπάνω πίνακα συνδυάζουμε τα γράμματα που βρίσκονται σην πρώτη στήλη και στην πρώτη γραμμή, ώστε να δημιουργήσουμε κρυπτογραφώντας παράλληλα το αρχικό κείμενό μας. Έτσι έχουμε:

u n i v e r s i t y p a t r a s
GG FX GD FF XF XD FG GD AD XX AX AF AD XD AF FG

3.1.13 Αλγόριθμος XOR [10]

Στην κρυπτογραφία, ο απλός κρυπταλγόριθμος XOR είναι ένα είδος προσθετικού κρυπταλγόριθμου,ένας αλγόριθμος κρυπτογράφησης που λειτουργεί σύμφωνα με τις αρχές:

$A \oplus 0 = A,$

$A \oplus A = 0,$

$(A \oplus B) \oplus C = A \oplus (B \oplus C),$

$(B \oplus A) \oplus A = B \oplus 0 = B,$

όπου $\oplus$ δηλώνει την αποκλειστική διάζευξη (XOR) λειτουργίας. Αυτή η λειτουργία ονομάζεται μερικές φορές modulus 2 προσθήκης (ή αφαίρεσης, η οποία είναι πανομοιότυπη.)

A. Με αυτή τη λογική, μια συμβολοσειρά κειμένου μπορεί να είναι κρυπτογραφημένη με την εφαρμογή του XOR σε κάθε χαρακτήρα, χρησιμοποιώντας ένα συγκεκριμένο κλειδί. Για παράδειγμα η λέξη «**University Patras**» (01010101,01101110,01101001,01110110,01100101,01110010,01110011,01101001, 01110100,01111001, 01010000,01100001,01110100,01110010,01100001,01110011 στο 8-bit ASCII) μπορεί να κρυπτογραφηθεί με την επανάληψη του **κλειδιού 11110011** όπως βλέπουμε στο επόμενο βήμα.

B. Κάνουμε πράξη XOR ανάμεσα σε κάθε γράμμα με το συγκεκριμένο κλειδί .

$$\begin{array}{r} 01010101 \\ \oplus\ 11110011 \\ \hline \end{array} \rightarrow 10100110$$

$$\begin{array}{r} 01101110 \\ \oplus\ 11110011 \\ \hline \end{array} \rightarrow 10011101$$

$$\begin{array}{r} 01101001 \\ \oplus\ 11110011 \\ \hline \end{array} \rightarrow 10011010$$

$$\begin{array}{r} 01110110 \\ \oplus\ 11110011 \\ \hline \end{array} \rightarrow 10000101$$

$$\begin{array}{r} 01100101 \\ \oplus\ 11110011 \\ \hline \end{array} \rightarrow 10010110$$

$$\begin{array}{r} 01110010 \\ \oplus\ 11110011 \\ \hline \end{array} \rightarrow 10000001$$

$$\begin{array}{r} 01110011 \\ \underline{11110011}\oplus \end{array} \rightarrow 10000000$$

$$\begin{array}{r} 01101001 \\ \underline{11110011}\oplus \end{array} \rightarrow 10011010$$

$$\begin{array}{r} 01110100 \\ \underline{11110011}\oplus \end{array} \rightarrow 10000111$$

$$\begin{array}{r} 01111001 \\ \underline{11110011}\oplus \end{array} \rightarrow 10001010$$

$$\begin{array}{r} 01010000 \\ \underline{11110011}\oplus \end{array} \rightarrow 10100011$$

$$\begin{array}{r} 01100001 \\ \underline{11110011}\oplus \end{array} \rightarrow 10010010$$

$$\begin{array}{r} 01110100 \\ \underline{11110011}\oplus \end{array} \rightarrow 10000111$$

$$\begin{array}{r} 01110010 \\ \underline{11110011}\oplus \end{array} \rightarrow 10000001$$

$$\begin{array}{r} 01100001 \\ \underline{11110011}\oplus \end{array} \rightarrow 10010010$$

$$\begin{array}{r} 01110011 \\ \underline{11110011}\oplus \end{array} \rightarrow 10000000$$

Έτσι έχουμε το κρυπτογραφημένο αποτέλεσμα XOR μορφή ASCII.

3.1.14 Αλγόριθμος Homophonic cipher [11]

Ο Homophonic είναι ενας αλγόριθμος στον οποίο τα γράμματα του απλού μας κειμένου μπορούν να αντικατασταθούν απο οποιαδήποτε διαφορετικά γράμματα ciphertext. Αυτός ο τρόπος είναι γενικά πιο δύσκολος να σπάσει σε σχέση με τους βασικούς αλγόριθμους κρυπτογράφησης υποκατάστασης.
Ο αριθμός των χαρακτήρων που αντικαθίσταται κάθε γράμμα είναι μέρος του κλειδιού, π.χ. το γράμμα «E» μπορεί να αντικατασταθεί από 5 διαφορετικά σύμβολα, ενώ το γράμμα «Q» μπορεί να αντικατασταθεί μόνο από 1 σύμβολο.

A. Κρυπτογραφούμε το κείμενο "**UNIVERSITY PATRAS**" . Βλέπουμε σύμφωνα με τον παρακάτω πίνακα με το αλφάβητο του κρυπταλγόριθμου , τα γράμματα που μπορούν να αντικαταστήσουν αυτά του κειμένου μας.

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
D	X	S	F	Z	E	H	C	V	I	T	P	G	A	Q	L	K	J	R	U	O	W	M	Y	B	N
9				7				3					5	0				4	6						
				2																					
				1																					

Αλφάβητο Homophonic Cipher

B. Όπως βλέπουμε κάποια γράμματα μπορούν να αντικατασταθούν με παραπάνω απο ενα , ή ακόμη και αριθμό.Επιλέξαμε τυχαία τα γράμματα αντικατάστασης. Οπότε το κρυπτογραφημένο κείμενό μας θα είναι:

UNIVERSITY PATRAS
O5VW2J43UB LD6J9R

3.2 Σύγκριση του size overhead

Στο κεφάλαιο αυτό χρησιμοποήσαμε το πρόγραμμα **Cryptool**, το οποίο είναι ένα ανοικτού κώδικα εργαλείο ηλεκτρονικής μάθησης για την επεξήγηση κρυπτογραφικών εννοιών. Συγκρίνουμε το μέγεθος των byte του κάθε αλγόριθμου και βρίσκουμε το size overhead του βέλτιστου.

Χρησιμοποιήσαμε τους συμμετρικούς αλγορίθμους Caesar, Vigenere, Hill, Playfair, ADFGVX, XOR, IDEA, DES (CBC) , DES (EBC) ,Triple DES, Homophone RC2 , RC4, Rijndael AES και τον Ασύμμετρο RSA.Τα αρχεία txt που κρυπτογραφίσαμε., είχαν μέγεθος 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000 Kbyte. Παρακάτω βλέπουμε τον Πίνακα με τα αποτελέσματα της σύγκρισης και το μέγεθος των kbyte που αλλάζει σε κάθε αλγόριθμο ξεχωριστά.

3.2.1 Πίνακας σύγκρισης Size Overhead

Αλγόριθμος / kb	200kb	400kb	600kb	800kb	1000kb	1200kb	1400kb	1600kb	1800kb	2000kb
Caesar	200	400	600	800	1000	1200	1400	1600	1800	2000
Vigenere	200	400	600	800	1000	1200	1400	1600	1800	2000
Hill	200	400	600	800	1000	1200	1400	1600	1800	2000
Playfair	**100**	**200**	**301**	**401**	**505**	**606**	**709**	**809**	**910**	**1011**
ADFGVX	**133**	**267**	**401**	**534**	**670**	**804**	**939**	**1072**	**1206**	**1340**
XOR	200	400	600	800	1000	1200	1400	1600	1800	2000
IDEA	200	400	600	800	1000	1200	1400	1600	1800	2000
RC2	200	400	600	800	1000	1200	1400	1600	1800	2000
RC4	200	400	600	800	1000	1200	1400	1600	1800	2000
DES(ECB)	200	400	600	800	1000	1200	1400	1600	1800	2000
DES(CBC)	200	400	600	800	1000	1200	1400	1600	1800	2000
TripleDES(ECB)	200	400	600	800	1000	1200	1400	1600	1800	2000
TripleDES(CBC)	200	400	600	800	1000	1200	1400	1600	1800	2000
AES (CBC)	200	400	600	800	1000	1200	1400	1600	1800	2000
Homophone	**63**	**126**	**189**	**252**	**317**	**380**	**444**	**507**	**570**	**633**
RSA (Asymmetric)	**203**	**406**	**609**	**812**	**1016**	**1219**	**1422**	**1625**	**1828**	**2031**

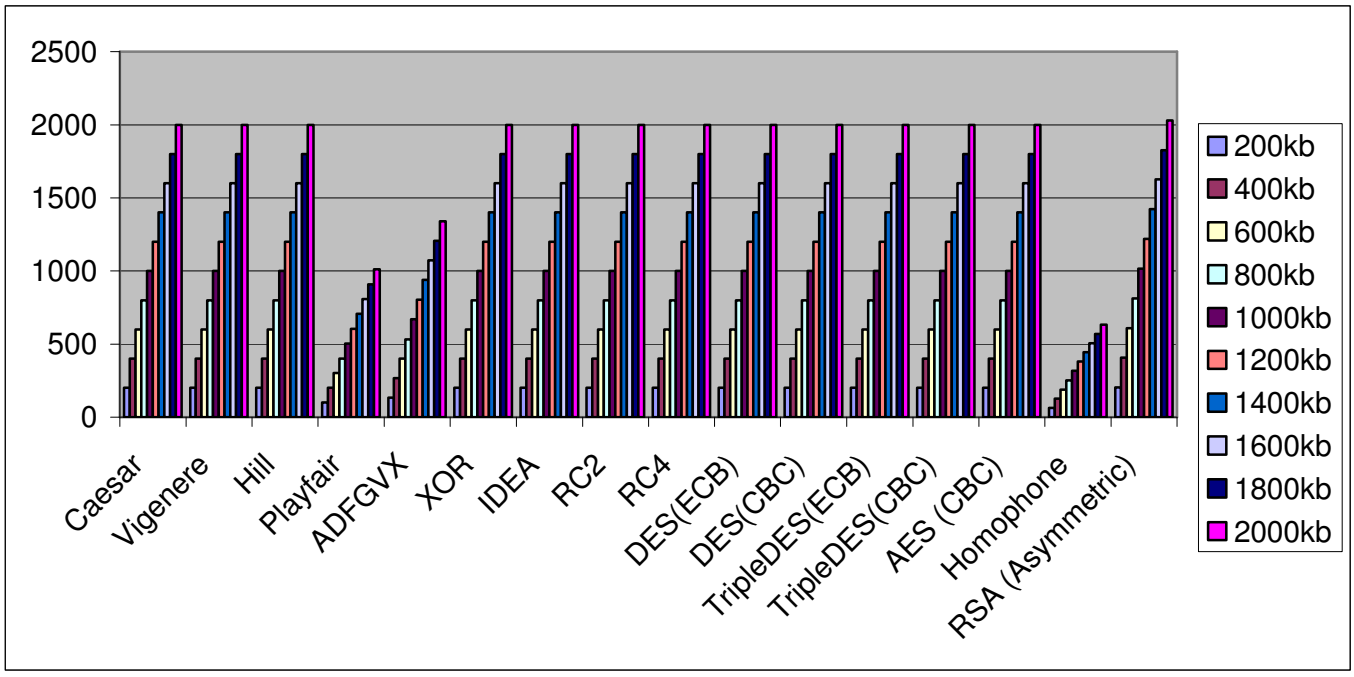
Παρατηρούμε πως στον αλγόριθμο **Playfair,** υπάρχει μια μείωση του μεγέθους του αρχείου περίπου **50%**.
Στον αλγόριθμο **ADFGVX** ,η μείωση είναι περίπου **67%** .
Για τον αλγόριθμο **Homophone**, βλέπουμε μείωση μεγέθους περίπου **31%**.
Στον ασύμμετρο αλγόριθμο **RSA** , το μέγεθος του κρυπτογραφημένου αρχείου αυξάνεται κατα **1,6%** περίπου.

Συμπέρασμα

Συμπεραίνουμε πως απο τους 15 συμμετρικούς αλγορίθμους που χρησιμοποιήσαμε , μόνον 3 απο αυτούς έχουν διαφορετικό Size Overhead , το οποίο είναι μικρότερο κάθε φορά.

Αντίθετα στον ασύμμετρο αλγόριθμο RSA παρατηρήσαμε αύξηση του Size Overhead κατα 1,6%.
Οπότε ,η συμμετρική κρυπτογράφηση, είναι προτιμότερη και συγκεκριμένα με τον αλγόριθμο Playfair, όπου η μείωση του μεγέθους του αρχείου είναι περίπου 50%.

3.2.2 Διάγραμμα σύγκρισης Size Overhead

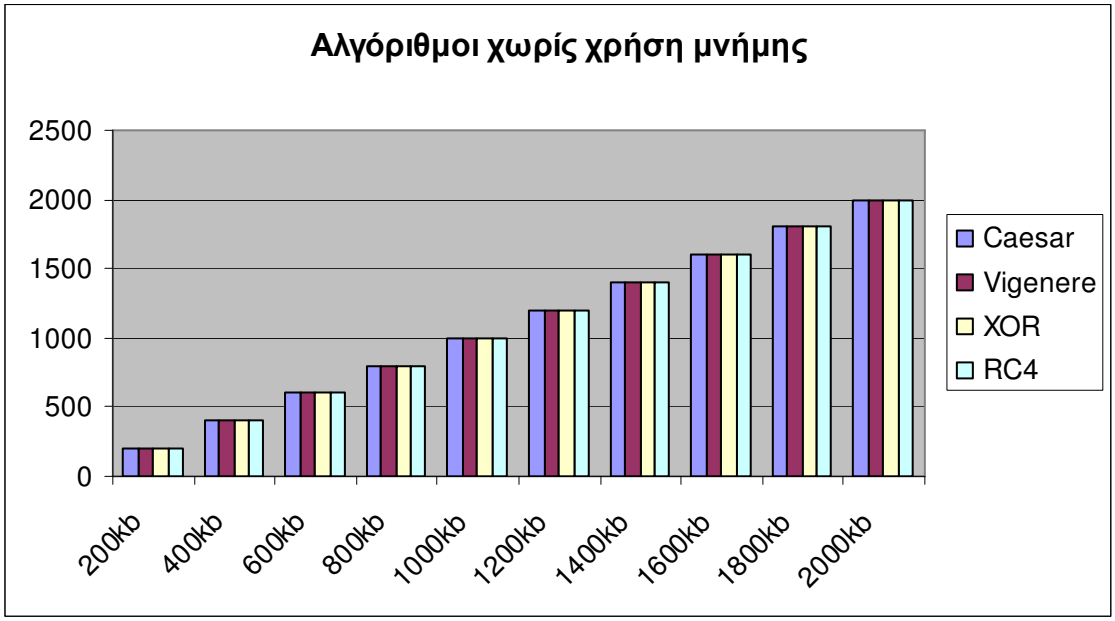


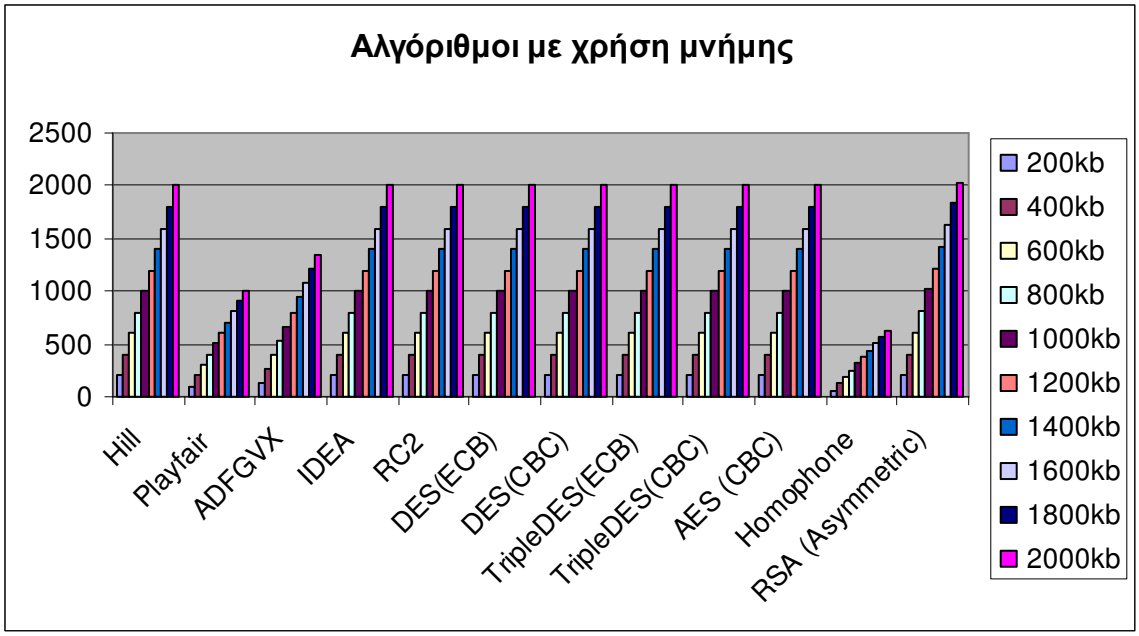
3.2.3 Διαχωρισμός αλγορίθμων σε σχέση χρήσης μνήμης κατά τη διάρκεια κρυπτογράφησης

Αλγόριθμοι με χρήση μνήμης
Hill
Playfair
ADFGVX
Homophone
IDEA
RC2
AES (CBC)
DES(ECB)
DES(CBC)
TripleDES(ECB)
TripleDES(CBC)
RSA (Asymmetric)

Αλγόριθμοι χωρίς χρήση μνήμης
Vigenere
Caesar
XOR
RC4

Ακολουθούν διαγράμματα με τις μετρήσεις, χωρισμένα σύμφωνα με τον τύπο μνήμης.





3.3 Συμπεράσματα και χρήσεις

Στο κεφάλαιο αυτό μελετήσαμε και συγκρίναμε τους συνηθέστερους αλγορίθμους κρυπτογράφησης.
Είδαμε λοιπόν οτι κάποιοι δουλέυουν με παρόμοιο τρόπο , κάποιοι είναι πιο ασφαλείς απο άλλους και κάποιοι είναι πιο πολύπλοκοι απο τους υπόλοιπους.
Παρατηρήσαμε πως ως πιο απλοί αλγόριθμοι, οι οποίοι είναι πιο κατανοητοί και εύκολοι στη χρήση , μπορούν να θεωρηθούν αρχικά ο αλγόριθμος Caesar , ο Vigenere οποίος είναι παρόμοιος, ο Playfair, ο ADFGVX , ο Homophone , ο RSA και ο Hill. Αυτό γίνεται αντιληπτό επίσης απο τα βήματα που χρειάστηκαν για τον κάθε ενα ωστε να κρυπτογραφήσουμε το μήνυμα μας.
Οι υπόλοιποι αλγόριθμοι , δηλαδή ο IDEA, DES, TripleDes, AES , RC4 και RC2 , έχουν κατα κάποιο τρόπο πιο πολύπλοκη διαδικασία κρυπτογράφησης ενος μηνύματος , οπότε είναι και πιο ασφαλείς.

Στο κομμάτι της χρήσης του κάθε αλγορίθμου , αρχίζουμε με τον αλγόριθμο **Caesar** , ο οποίος χρησιμοποιούνταν συνήθως για αλληλογραφίες και για να προστατεύσει μηνύματα στρατιωτικής σημασίας απο τον Ιουλιο Καίσαρα. Οπότε θα λέγαμε πως αν υπήρχε περίπτωση να χρησιμοποιήσουμε τον Caesar θα ήταν για κάποιο κωδικοποιημένο μήνυμα αλληλογραφίας ή κάποιο σύνθημα.

Ο **Vigenere** είναι παρόμοιος με τον Caesar και δημιουργήθηκε για τον ίδιο περίπου σκοπό. Άρα η χρήση του είναι για μετάδοση κωδικοποιημένων μηνυμάτων και συνθηματικών σε καιρούς πολέμου.

Ο **ADFGVX** είναι ένας ακόμη αλγόριθμος που η χρήση του αποσκοπούσε σε ανταλλαγή μυνημάτων εν κατάσταση πολέμου και χρησιμοποιήθηκε αρχικά απο τον Γερμανικό στρατό.

Ενας ακόμη αλγόριθμος που σχετίζεται η χρήση του σε πολεμικές καταστάσεις είναι ο **Playfair**. Χρησημοποιούνταν απο τους Άγγλους σε καιρούς πολέμου για κρυπτογράφηση σημαντικών εγγράφων, για αλληλογραφίες και για συνθηματικά.

Ο αλγόριθμος **Homophone** χρησιμοποιήθηκε επίσης για κρυπτογράφηση σημαντικών μηνυμάτων μεταξύ αρχηγών και κυβερνητικών εγγράφων κατα το παρελθόν.

Ο **Hill** ήταν ο πρώτος πολυγραφικός αλγόριθμος που μπορούσε να επεξεργαστεί πάνω απο 3 σύμβολα την ίδια στιγμή. Η χρήση του αποσκοπούσε σε κρυπτογράφηση μυνημάτων , με πιο περίπλοκο τρόπο απ'οτι οι προηγούμενοι αλγόριθμοι που αναφέραμε που τον καθιστά και πιο ασφαλή.

Ο αλγόριθμος **XOR** είναι αρκετά νεότερης εκδοσης αλγόριθμος και χρησιμοποιείται για κρυπτογράφηση διαφόρων αρχείων μέσω προγραμματισμού.πχ c++ . Μετατρέπει ενα συνθηματικό απο απλά γράμματα αλφαβήτου σε κώδικα ASCII. Κατα τη γνώμη μας , μια περίπτωση που ο XOR θα μας ήταν χρήσιμος είναι η κρυπτογράφηση ενός αρχείου ηλεκτρονικού μηνύματος , για την αποφυγή ανάγνωσης απο ανεπιθύμητους.

Ο **RSA** είναι ένας απο τους πρώτους κρυπταλγόριθμους δημοσίου κλειδιού και χρησιμοποιείται για την ασφαλη μετάδοση δεδομένων στο Internet. Ενα παράδειγμα χρήσης του είναι σε Web browsers για την ασφαλή μετάδωση δεδομένων μεταξύ χρηστών και διαδικτύου.. Όπως και χρησιμοποιείται ήδη απο τη Microsoft για αυτό.

Ο **DES** άρχισε να αναπτύσεται απο τις αρχές του 1970 απο την εταιρία IBM και χρησιμοποιήθηκε για την προστασία σημαντικών και απόρρητων κυβερνητικών εγγράφων. Έχει ομως αντικατασταθεί ήδη λόγω της εύκολης αποκρυπτογράφησης του και κρίθηκε μη ασφαλής πλέον.

Ο IDEA είναι μια καλύτερη έκδοση του DES αλγορίθμου.Χρησιμοποιήθηκε για τη κρυπτογράφηση δεδομένων PGP η οποίο λειτουργεί για την επικοικωνία μέσω email , μετάδωση πληροφοριών και για ολόκληρους σκληρούς δίσκους.Οπότε αν είχαμε μια πλατφόρμα ανταλλαγης email και μηνυμάτων , ο αλγόριθμος IDEA θα μας ήταν χρήσιμος.

Ο αλγόριθμος **Triple DES** είναι μια πιο ισχυρή εκδοση του DES και δημιουργήθηκε ετσι ώστε να μη σπάει όπως θα έσπαγε ο απλος DES και επομένως για την καλύτερη και ασφαλέστερη κρυπτογράφηση. Η χρήση του Triple DES γίνεται γα πληρωμή με ηλεκτρονικό τρόπο σε διάφορους ιστοτόπους , όπως επίσης και σε ορισμένα λογισμικά της Microsoft για την καλύτερη ασφάλεια των χρηστών σε email και δεδομένα.

Ο αλγόριθμος **AES** είναι ο αντικαταστάτης του DES και χρησιμοποιέιται απο την Αμερικάνικη κυβέρνηση κυρίως , για την κρυπτογράφηση απόρρητων αρχείων και δεδομένων. Ο τρόπος λειτουργίας του καθιστά την απόπειρα παραβίασης κωδικών πρόσβασης και δεδομένων αδύνατη.

Ο **RC2** είναι ένας συμμετρικός αλγόριθμος , γρηγορότερος απο τον DES και μπορεί να γίνει ασφαλέστερος ή και λιγότερο ασφάλης απο τον DES ανάλογα με το μήκος του κλειδιού. Χρησιμοποιήθηκε επίσης για την κρυπτογράφηση σημαντυκών αρχειων και δεδομένων, αλλά πλέον δεν είναι αρκετά ασφαλής.

Τέλος , ο **RC4** είναι ενας ακόμη συμμετρικός αλγόριθμος ο οποίος είναι σχετικά γρήγορος και χρησιμοποιείται σε λειτουργίες του πρωτοκόλλου SSL και στις ασύρματες επικοινωνίες IEEE 802.11a/b/g. Θα μπορούσαμε δηλαδη να τον χρησιμοποιήσουμε για να ασφαλήσουμε ενα δικτυο με ασφάλεια WPA ή WEP για παράδειγμα.

Αναφορές Κεφάλαιο ΙΙΙ

[1] http://el.wikipedia.org/wiki/Κώδικας_του_Καίσαρα
[2] http://el.wikipedia.org/wiki/Αλγόριθμος_κρυπτογράφησης_Vigenère
[3] http://www.eap-pli.com/index.php/component/content/article/25-2012-07-12-17-57-42/195-des
[4] Κώστας Λιμνιώτης,Εργαστηριακός/Επιστημονικός Συνεργάτης του Τμήματος Πληροφορικής και Τεχνολογίας Υπολογιστών, Κρυπτογραφία - Θεωρία-Εργαστήριο- Εργαστηριακές Ασκήσεις http://users.teilam.gr/~klimn/
[5] Φλωκατούλα Δώρα "Συμμετρικοί Αλγόριθμοι Κρυπτογράφησης Δεδομένων – Οι περιπτώσεις των αλγορίθμων DES και TDEA",2012
[6] http://en.wikipedia.org/wiki/RC2
[7] Cryptography and Network Security Chapter 7 Fifth Edition by William Stallings
[8] http://en.wikipedia.org/wiki/International_Data_Encryption_Algorithm
[9] http://en.wikipedia.org/wiki/ADFGVX_cipher
[10] http://en.wikipedia.org/wiki/XOR_cipher
[11] http://practicalcryptography.com/ciphers/homophonic-substitution-cipher/

ΚΕΦΑΛΑΙΟ IV

4.1 ΚΑΤΑΜΕΤΡΗΣΗ ΚΑΙ ΣΥΓΚΡΙΣΗ ΒΗΜΑΤΩΝ ΥΠΟΛΟΓΙΣΜΟΥ ΚΡΥΠΤΟΓΡΑΦΗΣΗΣ ΚΕΙΜΕΝΟΥ 128BIT.

4.1.1 Αλγόριθμος Caesar

Η εφαρμογή του κώδικα Καίσαρα συνίσταται στην αντικατάσταση κάθε γράμματος του κειμένου με ένα άλλο το οποίο έχει σταθερή απόσταση από αυτό στο αλφάβητο. Στο παράδειγμα χρησιμοποιείται μετατόπιση τριών θέσεων, έτσι ώστε το Β του κειμένου να γίνεται Ε στο κρυπτογραφημένο κείμενο.

Α. Έχουμε τη λέξη **university patras.** Υπολογίζουμε και βρίσκουμε το $3^{ο}$ επόμενο γράμμα κατα το αγγλικό αλφάβητο για κάθε γράμμα των λέξεων ξεχωριστά.
Β. Άρα χρειαζόμαστε **16 βηματα** συνολικά ώστε να βρούμε το κρυπτογραφημένο αποτέλεσμα.

4.1.2 Αλγόριθμος Vigenere

Ο αλγόριθμος κρυπτογράφησης Vigenere χρησιμοποιεί ένα πινάκα κλειδιών για την κρυπτογράφηση κειμένου. Το κάθε γράμμα κρυπτογραφείτε σύμφωνα με τον αριθμό που υπάρχει στην θέση του μυστικού πινάκα.Στην ουσία η κρυπτογράφηση Vigenere αποτελείτε από πολλούς αλγορίθμους κρυπτογράφησης Καίσαρα με πολλές διαφορετικές τιμές μετατόπισης.

Α. Αποτελείται από το αλφάβητο, που αναγράφεται 26 φορές σε διαφορετικές γραμμές, κάθε αλφάβητο μετατοπίζεται κυκλικά προς τα αριστερά σε σχέση με την προηγούμενη αλφάβητο, που αντιστοιχούν στους 26 πιθανούς αλγόριθμους κρυπτογράφησης του Καίσαρα .
Β. Οπότε σε συνδυασμό με το κλειδί του , χρειαζόμαστε **16 βήματα** για το κρυπτογραφημένο αποτέλεσμα

4.1.3 Αλγόριθμος DES

Α. Στον Des ουσιαστικά έχουμε τα παρακάτω βήματα:

Έχουμε:

- Καθαρό κείμενο σε block των 64 bit (8 bytes)

- Κλειδί των 56 bit συν 1 bit ισοτιμίας για κάθε 7 bit (Σύνολο 64 bit)

Στη συνέχεια αρχίζει η διαδικασία:

Μετατρέπεται το απλο κείμενο μας σε μορφή ASCI.

Διαχωρίζεται σε 2 μπλοκ των 32 bit.

Διαχωρίζεται το κλειδί σε 16 υποκλειδιά.

Στη συνέχεια γίνεται μια μετάθεση των bit του κειμένου 64bit.

Διαχωρίζεται σε 32bit.

Έπειτα αρχίζει ο συνδυασμός κλειδιού και κειμένου και κατά την i-οστή φορά συμβαίνουν τα εξής:

Το δεξί κείμενο (έστω R_i) εισάγεται σε μία συνάρτηση $f(k_i,R_{i)}$

Στη συνέχεια λαμβάνει χώρα η πράξη L_i $f(k_i,R_i)$

Έχουμε $L_{i+1} = R_i$ και $R_{i+1} = L_i$ $f(k_i,R_i)$ 16 φορες.

Β. Οπότε για την μετατροπή του συγκεκριμέου κειμένου χρειαζόμαστε **37 υπολογιστικά βήματα.**

4.1.4 Αλγόριθμος AES (CBC) Rijandel

Ο AES είναι ένας κρυπταλγόριθμος τμήματος με F = G = {0, 1}^128, ενώ το κλειδί έχει μεταβλητό μέγεθος και μπορεί να είναι ίσο με 128, 192 ή 256 bits. Ο εξαιρετικά μεγάλος κλειδοχώρος καθιστά την εξαντλητική αναζήτηση πρακτικώς αδύνατη. Ο AES είναι επαναληπτικός κρυπταλγόριθμος και βασίζεται σε κρυπτογράφηση γινομένου. Ο αριθμός των γύρων r εξαρτάται από το μέγεθος του κλειδιού. Έτσι ο αριθμός των γύρων είναι ίσος με r128 = 10, r192 = 12 ή r256 = 14 για μεγέθη κλειδιών 128, 192 και 256 bits αντίστοιχα.

Α. Έχουμε μήκος κλειδιού 128bits άρα χρειάζονται 10 γύροι. Κάθε γύρος έχει 4 βήματα:
-Αντικατάσταση byte (Byte substitution) – χρήση s-boxes με καλά χαρακτηριστικά
-Ολίσθηση (Shift row)
-Συνδυασμός πολλών bit (Mix Column)
-Πρόσθεση (XOR) του κλειδιού

Β. Για το τελικό αποτέλεσμα θα χρειαστούμε **40 υπολογιστικά βήματα.**

4.1.5 Αλγόριθμος RSA.

Α. Στον αλγόριθμο RSA έχουμε ως πρωτο βήμα τις παρακάτω παραμέτρους:

1. Επιλέγονται δύο μεγάλοι πρώτοι αριθμοί, p και q (συνήθως πολύ μεγαλύτεροι από 10^{100})
2. Υπολογίζεται $n = p \times q$ και $z = (p-1) \times (q-1)$. Ο n ονομάζεται *υπόλοιπο RSA (RSA modulus)*
3. Επιλέγεται ένας πρώτος αριθμός ως προς τον z ο οποίος ονομάζεται e
4. Υπολογίζεται ο d έτσι ώστε $d \times e = 1 \bmod z$
5. Το δημόσιο κλειδί αποτελείται από το ζευγάρι (e, n) και το ιδιωτικό κλειδί από το ζευγάρι (d, n).

B. Σπάμε το μήνυμα σε blocks των δύο γραμμάτων, δηλαδή 8 και κάνουμε τους υπολογισμούς.

C. Για το τελικό αποτέλεσμα επομένως χρειάστηκαν **13 βήματα.**

4.1.6 Αλγόριθμος Hill .

Κάθε γράμμα αντιστοιχεί σε έναν αριθμό από modulo 26. Για να κρυπτογραφήσετε ένα μήνυμα, κάθε μπλοκ των n γραμμάτων (που θεωρείται ως n-διανυσματική συνιστώσα), πολλαπλασιασμένο με ένα αντιστρέψιμο n × n πίνακα, και πάλι modulo 26.Το πλέγμα που χρησιμοποιείται για την κρυπτογράφηση είναι το κλειδί κρυπτογράφησης, και θα πρέπει να επιλέγεται τυχαία από το σύνολο των αντιστρέψιμων n × n πινάκων (modulo 26). Η κρυπτογράφηση μπορεί, φυσικά, να προσαρμοστεί σε ένα αλφάβητο με οποιονδήποτε αριθμό από γράμματα

A. Στον Hill αλγόριθμο έχουμε το απλό κείμενο και τον πίνακα κλειδί .
B. Διαχωρίζονται τα γράμματα ανα ζευγάρια και γίνονται οι υπολογισμοί για την εύρεση του κρυπτογραφημένου μηνύματος.Ετσι χρειάζονται **8 βήματα.**

4.1.7 Αλγόριθμος Triple DES

Ο TDEA είναι απλά η εφαρμογή του DES τρεις φορές με τρία κλειδιά τα οποία χρησιμοποιούνται με συγκεκριμένη σειρά. Ο αλγόριθμος ακολουθεί τη διαδοχή: κρυπτογράφηση, αποκρυπτογράφηση, κρυπτογράφηση (EDE – encryption – decryption - encryption)

A. Ο Triple-DES χρησιμοποιεί ένα "πακέτο κλειδιών" που περιλαμβάνει τρία DES κλειδιά, K1, K2 και K3, καθένα από τα οποία είναι 56 bits.
B. Σύμφωνα με τη λειτουργία του DES προσθέτοντας τη διαδικασία των τριών κλειδιών , θα χρειαστούν **68 βήματα** για την κρυτπογράφηση του κειμένου μας.

4.1.8 Αλγόριθμος RC2

Υπάρχουν δύο διακριτά μέρη για τη χρήση RC2. Πρώτον, ένα κλειδί επέκτασης βασικό για την διαδικασία παίρνει ένα κλειδί του χρήστη που παρέχεται μεταξύ ενός και 128 bytes σε μήκος, μαζί με μια παράμετρο που καθορίζει την αποτελεσματική κρυπτογράφηση του μήκους του κλειδιού.
Από αυτές τις πληροφορίες μια σειρά από K 64,16-bit γύροι κλειδιών λαμβάνονται. Στη συνέχεια, ένα 64-bit plaintext μπλοκ κρυπτογραφείται χρησιμοποιώντας συστοιχία K. Η κρυπτογράφηση αποτελείται από δύο μορφές γύρων.

A. Στον RC2 έχουμε 16 γύρους τύπου MIXING ΚΑΙ 2 γύρους τύπου MASHING. Ένας γύρος Mixing αποτελείται από τέσσερις εφαρμογές του μετασχηματισμού MIX.
B. Άρα για το τελικό μας αποτέλεσμα χρειάζονται **70 υπολογιστικά βήματα.**

4.1.9 Αλγόριθμος Playfair

Η λογική του Playfair είναι να αντικαθιστά περισσότερα γράμματα τη φορά με σκοπό να καταστρέψει τη δομή αυτή δηλαδή την κρίσιμη πληροφορία του plaintext να διαρρέεται απο το ciphertext.Στον αλγόριθμο Playfair, πρώτα επιλέγουμε ένα κλειδί.

A.Ένας 5x5 πίνακας συμπληρώνεται με τη λέξη-κλειδί(δύο ίδια γράμματα δεν εμφανίζονται δυο φορές)και οι υπόλοιπες θέσεις του πίνακα συμπληρώνονται από τα εναπομείναντα γράμματα του αλφάβητου.
B. Στη συνέχεια το κείμενο σπάει σε δυάδες και υπολογίζεται 8 φορες εφόσον η λέξη μας αποτελείται απο 16 γράμματα.Άρα ακολουθούνται **9 βήματα** συνολικά.

4.1.10 Αλγόριθμος ADFGVX

Το κρυπτογράφημα ADFGVX χρησιμοποιεί ταυτόχρονα αντικατάσταση και μετάθεση. Για την αντικατάσταση χρησιμοποιείται ένας πίνακας από επτά γραμμές και επτά στήλες του οποίου η πρώτη γραμμή και η πρώτη στήλη συμπληρώνονται από τους χαρακτήρες ADFGVX. Στην συνέχεια ο πίνακας συμπληρώνεται με 36 στοιχεία αποτελούμενα από 26 γράμματα και 10 πρώτους αριθμούς τοποθετημένα με τυχαίο τρόπο.Η διάταξη των στοιχείων αποτελεί το κλειδί του κρυπτογραφήματος. Η κρυπτογράφηση συνίσταται στη μετάφραση κάθε χαρακτήρα του μηνύματος με ένα ζεύγος γραμμάτων από την ομάδα ADFGVX.

A. Σε αυτή τη περίπτωση έχουμε το κείμενο «universitypatras» . Αρχικά γεμίζουμε εναν πίνακα διαστάσεων 5x5 με μια ανακατεμένη αλφάβητο.
B. Στη συνέχεια συνδυάζουμε το κείμενό μας , καθε γράμμα ξεχωριστά με τη λέξη κλειδί και πέρνουμε το αποτέλεσμα μετά απο **17 βήματα** συνολικά για τη συγκεκριμένη περίπτωση.

4.1.11 Αλγόριθμος XOR

A. Στη διαδικασία του αλγορίθμου XOR , αρχικά το κείμενο μας μετατρέπεται σε μορφή ASCII μαζί με το κλειδί του.

B. Έπειτα ακολουθεί η πράξη XOR μεταξύ τους , εως ώστε να βρεθεί το κρυπτογραφημένο αποτέλεσμα. Στη συγκεκριμένη περίπτωση χρειάστηκαν **37 υπολογιστικά βήματα** συνολικά.

4.1.12 Αλγόριθμος Homophonic cipher

Στην κρυπτογραφία, ένα κρυπτογράφημα υποκατάστασης είναι μια μέθοδος με την οποία κωδικοποιεί μονάδες απλού κρυπτοκειμένου που έχουν αντικατασταθεί,σύμφωνα με ένα κανονικό σύστημα.Οι «μονάδες» μπορεί να είναι μεμονωμένα γράμματα (η πιο συνηθισμένα), ζεύγη γραμμάτων, τριάδες γραμμάτων, μείγματα των παραπάνω, και ούτω καθεξής. Ο δέκτης αποκρυπτογραφεί το κείμενο, εκτελώντας μια αντίστροφη υποκατάστασης.

A. Ο Homophonic είναι ενας αλγόριθμος στον οποίο τα γράμματα του απλού μας κειμένου μπορούν να αντικατασταθούν απο οποιαδήποτε διαφορετικά γράμματα ciphertext.
B. Για την κρυπτογράφηση του κειμένου μας αυτή τη φορά θα χρειαστούν **18 βήματα** , όσο και τα γράμματα που έχουμε.

4.1.13 Αλγόριθμος IDEA

A. Ο IDEA λειτουργεί με μπλοκ 64-bit χρησιμοποιώντας ένα κλειδί 128-bit, και αποτελείται από μια σειρά από οκτώ πανομοιότυπες μεταμορφώσεις (ενας γύρος,) και μιας μεταμόρφωσης εξόδου
B. Ο IDEA χρησιμοποιεί μια λειτουργία μισού γύρου που εξαρτάται απο το κλειδί. Για να συνεργαστεί με τις λέξεις 16 bit (δηλαδή τέσσερις εισόδους αντί για δύο για το μέγεθος του μπλοκ 64 bit), ο IDEA χρησιμοποιεί το σύστημα Lai-Massey δύο φορές παράλληλα, με τις δύο παράλληλες λειτουργίες γύρων που είναι συνυφασμένες με την άλλη. Για να εξασφαλιστεί επαρκής διάχυση, δύο από τα υπο-μπλοκ αντιστρέφονται μετά από κάθε γύρο.
C. Κάθε γύρος χρησιμοποιεί έξι 16-bit υπο-κλειδιά, ενώ ο μισός γύρος χρησιμοποιεί τέσσερις, συνολικά 52 για 8.5 γύρους. Τα πρώτα οκτώ sub-keys εξάγονται απευθείας από το κλειδί, με K1 από τον πρώτο γύρο να είναι τα χαμηλότερα δεκαέξι bits. Περαιτέρω ομάδες των οκτώ κλειδιών δημιουργούνται περιστρέφοντας το κύριο κλειδί αριστερά 25 bits ανάμεσα σε κάθε ομάδα των οκτώ. Αυτό σημαίνει ότι περιστρέφεται λιγότερο από μία φορά ανά γύρο, κατά μέσο όρο, για ένα σύνολο έξι περιστροφών.
D. Στη περίπτωση μας δηλαδη θα χρειαστούν **51 βήματα** συνυπολογίζοντας τους γύρους και τις περιστροφές.

4.1.14 Αλγόριθμος RC4

Ο αλγόριθμος RC4 είναι εξαιρετικά απλο και πολύ εύκολο να εξηγηθεί. Ένα κλειδί μεταβλητού μήκους από 1 έως 256 byte (8-2048 bits) χρησιμοποιείται για να προετοιμάσει ένα 256-byte διάνυσμα κατάστασης S, με στοιχεία S [0], S [1], ..., S [255].Ανά πάσα στιγμή,το S περιέχει μια μετάθεση όλων των 8-bit αριθμών από 0

έως 255. Για την κρυπτογράφηση και την αποκρυπτογράφηση, ένα byte k παράγεται από το S επιλέγοντας μία από τις 255 καταχωρήσεις με ένα συστηματικό τρόπο.

A. Αρχιζει με ενα array S με αριθμους: 0..255
• Χρησιμοποιουμε το κλειδι για να ανακατεψουμε καλα
• Το S σχηματιζει την εσωτερικη κατασταση του αλγορίθμου
for i = 0 to 255 do
S[i] = i
T[i] = K[i mod keylen])
j = 0
for i = 0 to 255 do
j = (j + S[i] + T[i]) (mod 256)
swap (S[i], S[j])

B. Η κρυπτογραφηση συνεχιζεται ανακατευοντας τις τιμες του array. Το αθροισμα του ανακατεμενου ζευγους επιλεγει την τιμη του "stream key" απο τη μεταθεση.

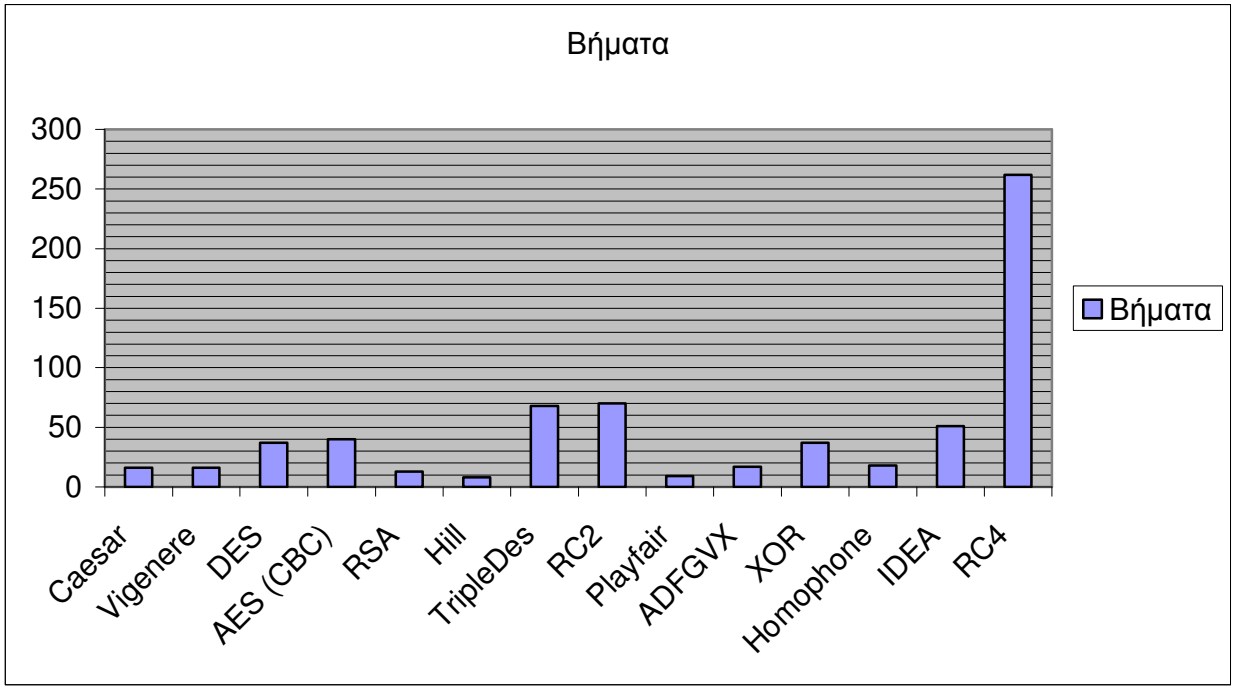
C. Κανουμε XOR το S[t] με το επομενο byte του μημυματος για να κρυπτογραφησουμε
/αποκρυπτογραφησουμε
i = j = 0
for each message byte M
i
i = (i + 1) (mod 256)
j = (j + S[i]) (mod 256)
D. Άρα για το δικό μας κείμενο μεγεθους 16byte , θα χρειαστούν συνλικά **262 βήματα** για το τελικό κρυπτογραφημένο αποτέλεσμα.

Συμπέρασμα RC4

Οπως βλέπουμε, ο αλγόριθμος RC4 χρειάζεται τα περισσότερα βήματα και με διαφορά , για να κρυπτογραφήσει ενα μήνυμα σε σύγκριση με τους υπόλοιπους αλγορίθμους που συγκρίναμε στο πείραμά μας.
Αυτο οφείλεται στο οτι ο RC4 ανήκει στους κωδικοποιητές ροής (stream ciphers) ο οποίοι κωδικοποιούν κάθε bit του μηνύματος ξεχωριστά, σε αντίθεση με τους κωδικοποιητές τμημάτων (block ciphers) οι οποίοι κωδικοποιούν το μήνυμα ανά τμήματα από bits.Χρησιμοποιεί εναν πίνακα 256Byte καθε φορά και για αυτό αυξάνονται τα βήματα κρυπτογράφησης σε καθε επανάληψη.Επίσης γίνεται χρήση XOR κρυπτογράφησης στη διαδικασία αυτή.
Έτσι λόγω του πιο πολύπλοκου τρόπου κρυπτογράφησης τα βήματα είναι περισσότερα και βλέπουμε χρήσεις του σε ασφάλειες WEP, WPA , SSL/TSL και σε διάφορες άλλες εφαρμογές.

Πίνακας σύγκρισης βημάτων

Αλγόριθμος	Βήματα
Caesar	16
Vigenere	16
DES	37
AES (CBC)	40
RSA	13
Hill	8
TripleDes	68
RC2	70
Playfair	9
ADFGVX	17
XOR	37
Homophone	18
IDEA	51
RC4	262



Διάγραμμα σύγκρισης βημάτων.

4.2 Γενικό συμπέρασμα

Συγκρίνοντας τα βήματα όλων των αλγορίθμων είδαμε πως κάποιοι χρειάζονται πάνω κάτω τον ίδιο αριθμό βημάτων για την κρυπτογράφηση ενός κειμένου και άλλοι έχουν μεγαλύτερη διαφορά μεταξύ τους.
Την μεγαλύτερη διαφορά την έκανε ο αλγόριθμος RC4 όπου αποδείχθηκε και ως ο πιο 'αργός' σύμφωνα με τα βήματα του. Οι αλγόριθμοι RSA, Playfair και Hill έχουν σχετικά παρόμοιο και τον μικρότερο αριθμό βημάτων για την εκτέλεση της κρυπτογράφησης.

ΚΕΦΑΛΑΙΟ V

5.1 ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ ΓΙΑ ΠΑΡΟΜΟΙΕΣ ΜΕΤΡΗΣΕΙΣ ΤΩΝ ΑΛΓΟΡΙΘΜΩΝ ΚΡΥΠΤΟΓΡΑΦΗΣΗΣ

5.1.1 Έρευνα για την μυστικότητα και την ανάλυση των συμμετρικών αλγορίθμων κρυπτογράφησης [1]

Σε αυτή την έρευνα, επιλέξαμε τους Block ciphers DES, Triple DES και AES (σε λειτουργία ECB), καθώς και τους Stream ciphers RC2 και RC4
Η ανάλυση γίνεται με βάση δύο κριτήρια μέτρησης σε δύο περιπτώσεις.
Οι δύο περιπτώσεις:
1) Πρώτη περίπτωση: η μεταβλητή είναι το μέγεθος εισόδου plaintext (ως είσοδος δίνεται αρχεία κειμένου, το μέγεθος μετράται σε kilobyte)
2) Δεύτερη περίπτωση: η μεταβλητή είναι το μήκος εισόδου plaintext, δηλαδή, το μήκος χαρακτήρων (ως είσοδοι θεωρούται οι κωδικοί πρόσβασης)
Δύο κριτήρια μέτρησης:

(i) Η μυστικότητα των Ciphers

(ii) Ο χρόνος Κρυπτογράφησης

Στοχοι έρευνας:
Προσδιορισμός καλύτερου αλγόριθμου με βάση την απόδοση του αλγορίθμου και τη μυστικότητα της κρυπτογράφησης
Συμπέρασμα έρευνας
Οι αποδόσεις όλων των Stream ciphers είναι υψηλότερες από των block ciphers ενώ όσον αφορά την μυστικότητα των αλγορίθμων, οι blocks cipher είναι καλύτερες από των stream cipher.

5.1.2. Ανάλυση απόδοσης των αλγορίθμων κρυπτογράφησης [2]

Σε αυτή την έρευνα, επιλέξαμε τους DES,3DES, Blowfish, AES
Η εργασία αυτή προσπαθεί να παρουσιάσει μια δίκαιη σύγκριση μεταξύ των πιο κοινών αλγορίθμων στο πεδίο της κρυπτογράφησης.Κύριο μέλημά μας εδώ είναι η απόδοση αυτών των αλγορίθμων κάτω από διαφορετικές μετρήσεις.
Κριτήρια μέτρησης:
1) Ταχύτητα κρυπτογράφησης
2) Απόδοση σε σχέση με την κρυπτογράφηση αρχείων εισόδου διάφορων περιεχομένων και μεγεθών

Συμπεράσματα

Φαίνεται ότι ο Blowfish και ο AES έχουν την καλύτερη επίδοση ταχύτητας μεταξύ των άλλων. Και οι δύο είναι γνωστό ότι έχουν καλύτερη κρυπτογράφηση (δηλαδή ισχυρότερη έναντι επιθέσεων δεδομένων) από ό, τι τους άλλους δύο.
Τα αποτελέσματα έδειξαν ότι ο Blowfish έχει μια πολύ καλή απόδοση σε σύγκριση με τους άλλους αλγορίθμους. Επίσης έδειξε ότι ο AES έχει καλύτερες επιδόσεις από

ό, τι οι 3DES και DES. Τέλος δείχνει επίσης ότι ο 3DES χρειάζεται 3 φορές περισσότερο χρόνο από τον DES για να επεξεργαστεί την ίδια ποσότητα δεδομένων.

5.1.3. Ανάλυση και μοντελοποίηση κρυπτογράφησης overheads για sensor network nodes [3]

Αυτή η έρευνα διερευνά τις υπολογιστικές απαιτήσεις για μια σειρά από δημοφιλείς κρυπτογραφικούς αλγορίθμους που εμφανίζονται σε ενσωματωμένες αρχιτεκτονικές. Επιλέχτηκαν οι αλγοριθμοι RC4, IDEA, RC5, MD5, SHA1 σε συγκεκριμένες πλατφόρμες επεξεργαστών δίκτυων αισθητήρων οι οποιες είναι Atmega 103/Atmega 128, M16C/10, UltraSPARC II ,StrongARM SA-1110,XScale PXA250

Συμπέρασμα

Οι πειραματικές μετρήσεις δείχνουν ομοιόμορφο κρυπτογραφικό κόστος για κάθε κατηγορία κρυπτογράφησης και για κάθε κατηγορία αρχιτεκτονικής αμελητέα επίπτωση κρυφής μνήμης.Διαπιστώθηκε ότι ο RC4 φαίνεται να έχει υψηλότερες επιδόσεις από τον RC5 σε χαμηλή ποιότητα επεξεργαστών σε αντίθεση με την επιλογή του RC5 σε αισθητήρες ρεύματος.

5.1.4. Συγκριτική ανάλυση της αποτελεσματικής απόδοσης και μέτρα ασφάλειας μερικών αλγορίθμων κρυπτογράφησης [4]

Το συσγκεκριμένο έγγραφο παρέχει μια δίκαιη σύγκριση των επιδόσεων μεταξύ των διάφορων αλγορίθμων κρυπτογράφησης, των AES, RSA, RC2, DES, 3DES,Blowfish, DSA. Έχουμε συγκρίνει τις εξης παραμέτρους: απόσβεση(Tunability) , το μέτρο της ποσότητας των δεδομένων που πρόκειτε να κρυπτογραφηθεί(encryption ratio), μήκος κλειδιού, υπολογιστική ταχύτητα, και το είδος των επιθέσεων σχετικά με τα ζητήματα ασφαλείας που παρέχονται.

Συμπέρασμα

Το ποσοστό κρυπτογράφησης είναι υψηλό σε χρήση της συμμετρικής κρυπτογράφησης, η απόσβεση(Tunability) είναι υψηλότερη στην τεχνική ασύμμετρης κρυπτογράφησης, το μήκος του κλειδιού είναι υψηλό στην ασύμμετρη κρυπτογράφηση, να παραβιαστεί ο κωδικός είναι σύνθετο στον RSA. Στο θέμα της ταχύτητας η συμμετρική κρυπτογράφηση θεωρείται ως καλή. Τέλος, στην συμμετρική κρυπτογράφηση ο αλγόριθμος AES ορίζεται ως η καλύτερη λύση, στη συνέχεια, ακολουθεί ο αλγόριθμος blowfish. Στην ασύμμετρη τεχνική κρυπτογράφησης ο αλγόριθμος RSA είναι πιο ασφαλείς δεδομένου ότι χρησιμοποιεί "πρακτορείο" υψηλών πρώτων αριθμών για την παραγωγή κλειδιών. Ως εκ τούτου, ο αλγόριθμος RSA είναι η καλύτερη λύση σε αυτή τη μέθοδο.

5.1.5. Συγκριτική ανάλυση των κρυπτογραφικών αλγορίθμων[5]

Αναλύθηκαν τρεις αλγόριθμοι DES, Triple DES και RSA, έχουν αναλυθεί σχετικά με την ικανότητά τους για ασφαλή δεδομένα, το χρόνο που απαιτείται για την κρυπτογράφηση των δεδομένων και το throughput. Οι αποδόσεις των διαφορετικών αλγορίθμων είναι διαφορετικές ανάλογα με τις εισόδους τους.

Συμπεράσματα

Η ταχύτητα της κρυπτογράφησης του DES είναι δύο φορές την ταχύτητα της κρυπτογράφησης του RSA,ο DES καταναλώνει επίσης λιγότερη ισχύ σε σύγκριση με την ισχύ του RSA.Ο Triple-DES εξακολουθεί να απαιτεί περισσότερο χρόνο από ό, τι ο DES επειδή ο DES κρυπτογραφεί τα δεδομένα μόνο μία φορά και ο Triple DES κρυπτογραφεί τα δεδομένα τρεις φορές.Ο Triple-DES έχει περισσότερη κατανάλωση ισχύος και λιγότερο throughput από τον DES λόγω των χαρακτηριστικών της τριπλής φάσης. Αλλά αυτό που λείπει στους DES και RSA και κάνει τον Triple DES, την επιλογή του αλγορίθμου μας είναι ασφάλεια.

5.1.6 Αξιολογώντας τις επιδόσεις των συμμετρικών αλγορίθμων κρυπτογράφησης[6]

Το παρόν έγγραφο παρέχει αξιολόγηση των έξι από τους πιο κοινούς αλγόριθμους κρυπτογράφησης και συγκεκριμένα: AES (Rijndael), DES, 3DES, RC2, Blowfsh και RC6.Μια σύγκριση έχει διεξαχθεί για τους εν λόγω αλγόριθμους κρυπτογράφησης σε διαφορετικές ρυθμίσεις για κάθε αλγόριθμο, όπως διαφορετικά μεγέθη των μπλοκ δεδομένων, διαφορετικούς τύπους δεδομένων, την κατανάλωση ισχύος της μπαταρίας, διαφορετικό μέγεθος του κλειδιού και, τέλος, την ταχύτητα κρυπτογράφησης.

Συμπεράσματα

Στην περίπτωση της αλλαγής μέγεθους του πακέτου, εξήχθη το συμπέρασμα ότι Blowfish έχει καλύτερη απόδοση από ό, τι οι άλλοι κοινοί αλγόριθμοι κρυπτογράφησης που χρησιμοποιούνται, ακολουθεί ο RC6. Διαπιστώθηκε ότι ο 3DES εξακολουθεί να έχει χαμηλή απόδοση σε σύγκριση με τον αλγόριθμο DES, βρέθηκε ότι ο RC2, έχει μειονέκτημα έναντι όλων των άλλων αλγορίθμων όσον αφορά την κατανάλωση του χρόνου,ακόμα ότι ο AES έχει καλύτερη απόδοση από ό, τι οι RC2, DES, 3DES. Τέλος,στην περίπτωση των αρχείων ήχου και βίντεο βρέθηκε το ίδιο αποτέλεσμα,όπως στο κείμενο και το έγγραφο.

5.1.7.Επισκόπιση συμμετρικών αλγορίθμων:συγκριτική ανάλυση [7]

Σε αυτή την ενότητα, έχουν αξιολογηθεί διαφορετικοί τύποι υφιστάμενων συμμετρικών αλγορίθμων. Για να εφαρμοστεί ο κατάλληλος αλγόριθμος σε μια συγκεκριμένη εφαρμογή είναι υποχρεωτικο να γνωρίζουμε τη δύναμη και τον περιορισμό του. Κατά συνέπεια, είναι απαραίτητη η αξιολόγηση αλγόριθμων με βάση συγκεκριμένες παραμέτρους. Οι παράμετροι μπορεί να περιλαμβάνουν την

αρχιτεκτονική,την ασφάλεια,την επεκτασιμότητα (από άποψη ρυθμού κρυπτογράφησης,τη χρήση μνήμης, την απόδοση του hardware και software σε υπολογιστικό χρόνο),τους περιορισμούς, και την ευελιξία.
Οι αλγόριθμοι που αναλύθηκαν είναι οι DES, 3DES, IDEA, Serpent, Blowfish, Rijndeal, RC6, CAST, RSA, PGP, MARS, TEA, Twofish

Κριτήρια

Τα κριτήρια βάσει των οποίων οι διαφορετικοί αλγόριθμοι αναλύονται είναι:

- Ασφάλεια
- Αρχιτεκλτονική
- Ευελιξία
- Περιορισμοί
- Επεκτασιμότητα

Μεθοδολογία αξιολόγησης

Η μεθοδολογία αξιολόγησης ήταν εύκολη και απλή, διαφορετικοί κωδικες αλγόριθμων κρυπτογράφησης λήφθηκαν, καθώς και πόροι, όπως εγχειρίδια, πηγαίου κώδικα και ερευνητικές εργασίες μελετήθηκαν.Κάθε αλγόριθμος αξιολογείται βάσει των παραπάνω παραμέτρων. Ούτε ένας αλγόριθμος δεν πληρεί τα κριτήρια αξιολόγησης, με κάποιους να έχουν περισσότερες ελλείψεις απο τους υπόλοιπους.

	Algorithm Structure	Plain Text/Cipher Text Length	Key Size	# S boxes	# of Rounds
DES	Festial structure	64 bits	56	8	16
3DES	Festial structure	64 bits	168	8	48
Blowfish	Festial structure	64 bits	128-448	4	16
IDEA	Substitution-Permutation Structure	64 bits	128	N/A	8
TEA	Festial structure	64 bits	128	N/A	64 (32 cycles)
CAST	Festial structure	64 bits	40-128	4	12 – 16
Rijndael	Festial structure	128 Bits	128,192,256	1	10,12,14
RC6	Festial structure	128 Bits	128,192,256	N/A	20
Serpent	Festial structure	128 Bits	128,192,,256	8	32
Twofish	Festial	128 Bits	128,192,256	4	16
MARS	Festial	128 Bits	128-448	1	32

Πίνακας 1: Χαρακτηριστικά αλγορίθμων

Συγκριτική Ανάλυση

Μετά την ανάλυση των δημοφιλέστερων συμμετρικών αλγόριθμων, ο AES (Rijndael) βρέθηκε ο πιο ασφαλής, πιο γρήγορος και καλύτερος ανάμεσα σε όλους τους αλλους αλγόριθμους, χωρίς σοβαρές αδυναμίες . Υπάρχουν κάποιες ατέλειες σε συμμετρικούς αλγόριθμους όπως αδύναμα κλειδιά , ανασφαλής μετάδοση μυστικού κλειδιού,η ταχύτητα , ευελιξία , έλεγχος ταυτότητας και η αξιοπιστία .Δηλαδή στον DES , τέσσερα κλειδιά κρυπτογράφησης η οποία είναι ακριβώς η ίδια με την αποκρυπτογράφηση. Αυτό σημαίνει ότι το Original απλό κείμενο μπορεί να ανακτηθεί , αν η κρυπτογράφηση εφαρμόζεται δύο φορές με ένα από αυτά τα αδύναμα κλειδία. Ο DES είναι πολύ αργός , όταν εφαρμόζονται σε λογισμικό και είναι καταλληλότερος για την υλοποίηση σε hardware . Παρόμοια είναι η περίπτωση στον IDEAπου περιλαμβάνει μεγάλη κατηγορία των αδύναμων κλειδιών που διευκολύνουν την κρυπτανάλυση για την ανάκτηση του κλειδιού .Οι DES και IDEA έχουν την ίδια ταχύτητα κρυπτογράφησης . Ο Triple-DES δεν παρέχει πάντα την επιπλέον ασφάλεια που αναμένεται κάνοντας χρήση της διπλής και τριπλής κρυπτογράφησης , καθώς επίσης είναι πολύ αργος , όταν εφαρμόζεται σε λογισμικό , όπως προκύπτει από τον DES και ο DES για το λογισμικό είναι ήδη αργός , έτσι ο TripleDES θα μπορούσε να θεωρηθεί ασφαλέστερος αλλά και πιο αργός . Στον Blow Fish υπάρχουν ορισμένες αδυναμίες κλειδίων που προσβάλλουν την έκδοση τριων γύρων , είναι επίσης εκτεθειμένος σε διαφορες επίθεσεις σε ορισμένες παραλλαγές του , εχει επίσης αργή ταχύτητα , αλλά είναι πολύ πιο γρήγορος από ό, τι ο DES και ο IDEA . Κοιτάζοντας τους πέντε τελευταίους, στον AES καμια αδυναμία δεν διαπιστώθηκε , ωστόσο, λίγες αδύναμες πλευρες τονίστηκαν πως θα μπορούσαν να εκτεθούν στο συντομο μέλλον , όπως στον AES (Rijndael) ενα αριθμητικό κομμάτι της κρυπτογράφησης μπορεί να εκτεθεί σε μια επίθεση .Πλήρη αυθαιρεσία RC6 δεν επιτυγχάνεται , ο Serpent είναι λίγο πιο αργος και πολύπλοκος , ο Twofish ενδεχομένως έχει επιλεγεί για βασικές επιθέσεις και ο MARS σχετικά πολύπλοκος για ανάλυση.

Συμπέρασμα

Σε αυτή την εργασία παρουσιάζεται μια λεπτομερή ανάλυση των συμμετρικών αλγόριθμων κρυπτογράφησης μπλοκ με βάση διαφόρων παραμέτρων. Ο κύριος στόχος ήταν να αναλυθεί η απόδοση από τους δημοφιλέστερους αλγορίθμους με συμμετρικό κλειδί από την άποψη της ταυτότητας, της ευελιξίας, της αξιοπιστίας, της αντοχής,της επεκτασιμότητας, της ασφάλειας, και να τονίστει η μεγάλη αδυναμία των προαναφερθέντων αλγορίθμων, καθιστώντας τη δύναμη του κάθε αλγορίθμου και τους περιορισμούς, διαφανης για τις εφαρμογες . Κατά τη διάρκεια αυτής της ανάλυσης παρατηρήθηκε ότι ο AES (Rijndael) ήταν ο καλύτερος μεταξύ όλων από την άποψη της ασφάλειας, Ευελιξίας, χρήσης μνήμης, και την απόδοση κρυπτογράφησης .Παρόλα ταυτα οι άλλοι αλγόριθμοι ήταν επίσης ικανοί, αλλά οι περισσότεροι από αυτούς έχουν μια ανταλλαγή, μεταξύ της χρησης μνήμης και την απόδοση κρυπτογράφησης με λίγους αλγόριθμους να έχουν παραβιαστεί.

5.1.8.Ανάλυση των επιπτώσεων των συμμετρικών κρυπτογραφικών αλγορίθμων για την κατανάλωση ισχύος σε διάφορους τύπους δεδομένων [8]

Στην ανάλυση αυτή θα δούμε την σύγκριση κατανάλωσης πόρων ανάμεσα στους αλγορίθμους 3DES, AES, Blowfish, Computer Security, DES, Encryption Techniques, RC2, RC5 κατα τη διάρκεια κρυπτογράφησης δεδομένων.

Θα χρησιμοποιηθεί συμμετρική και ασύμμετρη κρυπτογράφηση.
Ο συμμετρικός μηχανισμός περιέχει τα παρακάτω στοιχεία:

- Απλό κείμενο
- Αλγόριθμος κρυπτογράφησης
- Μυστικό κλειδί
- Κρυπτογραφημένο κείμενο
- Αλγόριθμος αποκρυπτογράφησης

Ο ασύμμετρος μηχανισμός περιέχει:

- Απλό κείμενο
- Αλγόριθμος κρυπτογράφησης
- Μυστικό κλειδί και δημόσιο κλειδί
- Κρυπτογραφημένο κείμενο
- Αλγόριθμος αποκρυπτογράφησης

Για το πείραμα συλλεχθηκαν πληροφορίες για:

1. Power Consumption
2. Encryption Time
3. CPU Process Time
4. CPU Clock Cycles

Χρησιμοποιήθηκε ασύρμματη συσκευή και λάπτοπ με 1.5 GHZ CPU **.**

Συμπεράσματα

Αυτή η εργασία παρουσιάζει μια αξιολόγηση των επιδόσεων των διαφόρων συμμετρικών αλγορίθμων . Οι επιλεγμένοι αλγόριθμοι είναι οι AES , DES , 3DES , RC6 , Blowfish και RC2.Πολλά συμπεράσματα μπορεί κανείς να βγάλει από τα αποτελέσματα της προσομοίωσης και διαπιστώνεται ότι Blowfish παρέχει την καλύτερη απόδοση μεταξύ όλων των αλγορίθμων , στη συνέχεια, μετά από αυτό ο καλύτερος αλγόριθμος που καταναλώνει λιγότερη ενέργεια και λιγότερο χρόνο είναι ο RC6 και η χειρότερη προσέγγιση μεταξύ όλων των αλγορίθμων όσον αφορά το φόρτο της CPU είναι ο RC2 διότι οδηγεί σε βαρύ φόρτο εργασίας της CPU , καθώς είναι παράγοντας κατανάλωσης μεγάλου χρόνου

5.1.9. Μελέτη και ανάλυση απόδοσης αλγορίθμων κρυπτογράφησης [9]

Περιγραφή πειράματος

Σε αυτή την εργασία λαμβάνοντας υπόψη τις επιδόσεις του αλγορίθμου κρυπτογράφησης για αρχεία κειμένου , χρησιμοποιείται ο AES , ο αλγόριθμος RSA και ο DES και αξιολογείται από τις ακόλουθες παραμέτρους όπως το χρόνο Υπολογισμού , Η χρήση της μνήμης , τα bytes εξόδου . Ο χρόνος κρυπτογράφησης υπολογίζεται πρώτα. Ο χρόνος που απαιτείται για να μετατραπεί το απλό κείμενο σε ciphertext είναι γνωστός ως χρόνος κρυπτογράφησης . Συγκρίνοντας αυτούς τους τρεις αλγορίθμους , ο RSA χρειάζεται περισσότερο χρόνο για τη διαδικασία υπολογισμού . Η χρήση της μνήμης του κάθε αλγορίθμου θεωρείται ως επίπεδο byte μνήμης . Ο RSA παίρνει μεγαλύτερη μνήμη από τον AES και τον DES. Εν τέλει , το byte εξόδου υπολογίζεται από το μέγεθος του byte εξόδου του κάθε αλγορίθμου.Το επίπεδο του byte εξόδου είναι το ίδιο για AES και DES , αλλά ο αλγόριθμος RSA παράγει χαμηλό επίπεδο byte εξόδου . Σε αυτό το έγγραφο, οι επιλεγμένοι αλγόριθμοι είναι οι AES , 3DES , Blowfish και DES . Με τη χρήση αυτών των αλγορίθμων η απόδοση της κρυπτογράφησης και της διαδικασίας αποκρυπτογράφησης αρχείων κειμένου υπολογίζεται μέσω της παραμέτρου απόδοσης. Ο χρόνος κρυπτογράφησης υπολογίζεται ως το σύνολο απλό κείμενο σε bytes που κρυπτογραφούνται διαιρούμενο δια του χρόνου κρυπτογράφησης..Ως ένα αποτέλεσμα που αναφέρεται , λέγεται ότι ο Blowfish αλγόριθμος δίνει την καλύτερη απόδοση από ό, τι όλοι οι άλλοι αλγόριθμοι από την άποψη της απόδοσης. Ο λιγότερο αποδοτικός αλγόριθμος είναι ο 3DES .

Στη συνέχεια , μελετάται η αξιολόγηση των επιδόσεων των AES και Blowfish αλγορίθμων και παρατηρείται πως ο Blowfish έχει καλύτερη απόδοση σε όλες τις μετρήσεις.

Αποτελέσματα πειράματος

Audio Files(KB)	**Χρονος κρυπρογράφησης Blowfish (ms)**	**Χρόνος κρυπτογράφησης AES (ms)**
8282	**970**	**1025**
387	**38**	**55**
33	**16**	**20**
2826	**348**	**370**
Μέσπς χρόνος	**343**	**347,5**

Μέσος χρόνος κρυπτογράφησης

Audio Files(KB)	**Χρονος αποκρυπρογράφησης**	**Χρόνος αποκρυπτογράφησης**

	Blowfish (ms)	AES (ms)
8282	**300**	**433**
387	**120**	**220**
33	**21**	**28**
2826	**55**	**97**
Μέσπς χρόνος	**124**	**194,5**

Μέσος χρόνος αποκρυπτογράφησης

Συμπέρασμα

Ο αλγόριθμος κρυπτογραφίας είναι η επιστήμη σε μυστικό κωδικό. Η ταχεία αύξηση της εγκληματικότητας στον κυβερνοχώρο και η αυξανόμενη προοπτική του διαδικτύου που χρησιμοποιείται ως μέσο για επιθέσεις, δημιουργεί μια μεγάλη πρόκληση για την ασφάλεια του δικτύου.
Τα αποτελέσματα που παρουσιάζονται απο την προσομοίωση των αρχείων ήχου δείχνουν το τελικό νόημα .Το συμπέρασμα είναι ότι ο Blowfish έχει καλύτερη απόδοση από τον AES σε μέσους χρόνους.

5.1.10.Συγκριτική ανάλυση αλγορίθμων κρυπτογράφησης για επικοινωνία δεδομένων [10]

Πείραμα

Τα πέντε αρχεία κειμένου των διαφόρων μεγεθών που χρησιμοποιούνται για τη διεξαγωγή πέντε πειραμάτων, όπου εκτελείται η συγκριτική τριών αλγορίθμων AES, DES και RSA. Η κρυπτογράφηση χρησιμοποιείται επίσης για τη διεξαγωγή πειραμάτων.

Παράμετροι αξιολόγησης

Η απόδοση του αλγορίθμου κρυπτογράφησης αξιολογείται λαμβάνοντας υπόψη τις ακόλουθες παραμέτρους.

Α. Περίοδος Υπολογισμού
Β. Η χρήση της μνήμης
Γ. Bytes εξόδου

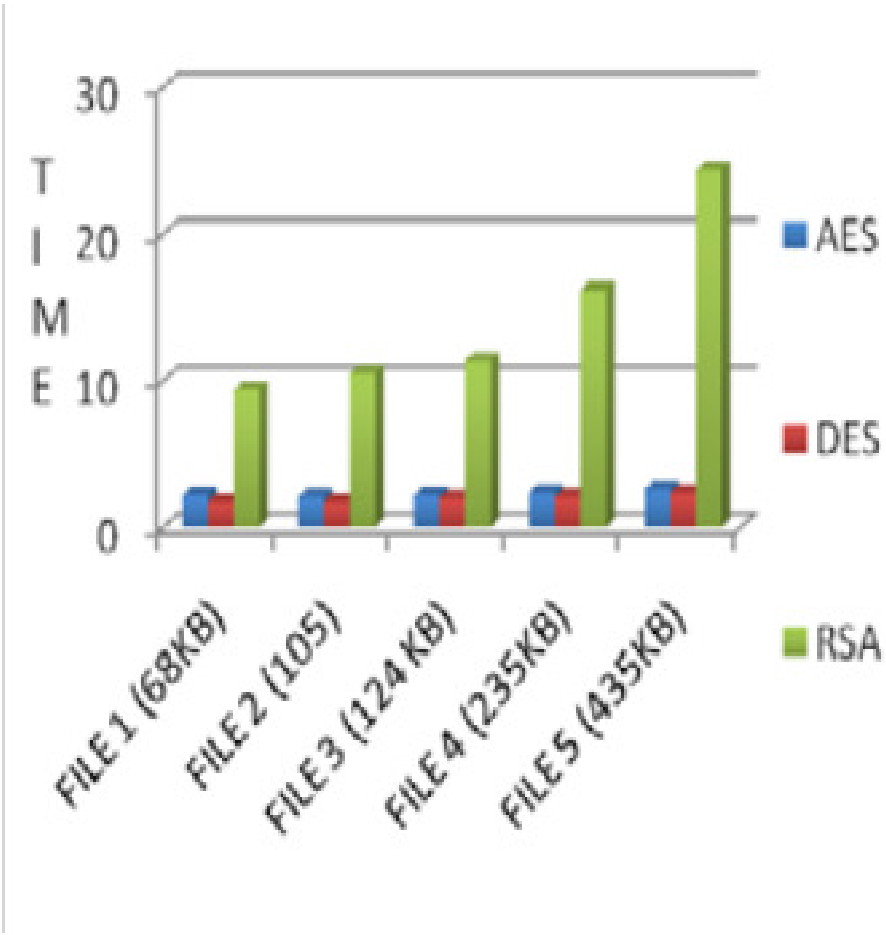
Ως χρόνος κρυπτογράφησης θεωρείται ο χρόνος που ένας αλγόριθμος κρυπτογράφησης χρειάζεται για να παράγει ένα κρυπτογραφημένο κείμενο από ένα απλό κείμενο. Ο χρόνος κρυπτογράφησης που χρησιμοποιείται για τον υπολογισμό της απόδοσης ενός συστήματος κρυπτογράφησης, υπολογίζεται ως το σύνολικό απλό κείμενο σε bytes κρυπτογραφημένο, διαιρούμενο με τον χρόνο κρυπτογράφησης. Εκτελούνται συγκριτικές αναλύσεις των αποτελεσμάτων των επιλεγμένων διαφορετικων σχήματων κρυπτογράφησης .

Πειραματικά Αποτελέσματα και Ανάλυση

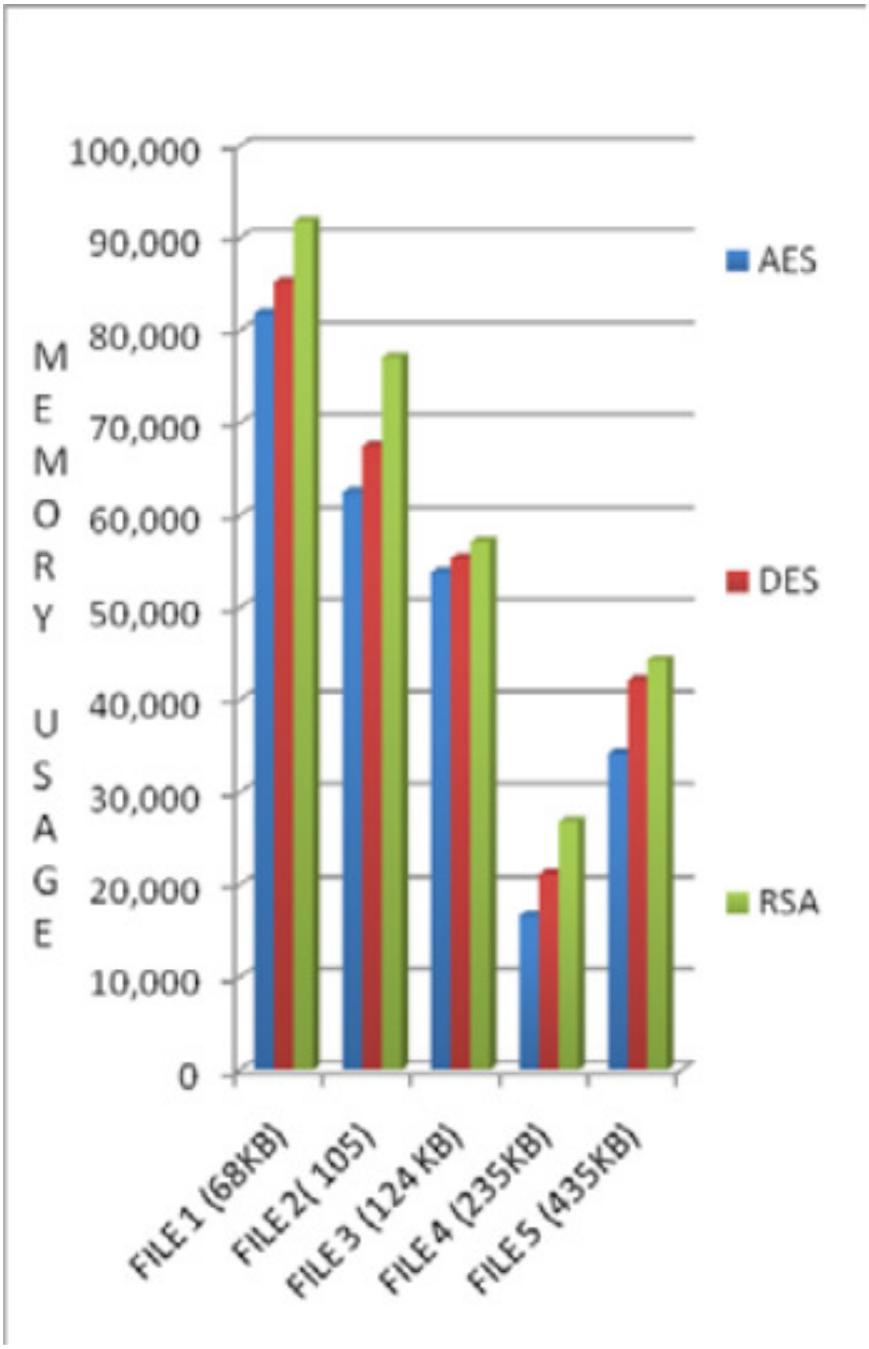
Τα πειραματικά αποτελέσματα για τον AES αλγόριθμο κρυπτογράφησης, τον DES και τον RSA φαίνονται στο πίνακα 1, ο οποίος δείχνει τη σύγκριση των τριών αλγορίθμων AES, DES και RSA χρησιμοποιώντας ίδιο αρχείο κειμένου για τα πέντε πειράματα, τα byte εξόδου για τους AES και DES είναι ίδια για διαφορετικά μεγέθη αρχείων. Με την ανάλυση του πίνακα 1, παρατηρήσαμε ότι ο RSA έχει πολύ μικρότερα byte εξόδου σε σύγκριση με τους AES και DES αλγόριθμους. Ο χρόνος που απαιτείται από τον αλγόριθμο RSA είναι πολύ υψηλότερος σε σύγκριση με το χρόνο που απαιτείται από τους AES και DES. Παρατηρείται μεταβολή της χρήσης της μνήμης. Δεν αυξάνεται ανάλογα με το μέγεθος του αρχείου σε όλους τους αλγορίθμους. Με την ανάλυση στο Σχήμα1 . αυτό που δείχνει το χρόνο Λήψης για κρυπτογράφηση σε διάφορες μέγεθος του αρχείου κειμένου από τρεις αλγορίθμους i: e AES, DES και RSA, έχει παρατηρήθεί ότι ο αλγόριθμος RSA παίρνει πολύ περισσότερο χρόνο σε σύγκριση με το χρόνο που λαμβάνονται από τους AES και DES αλγόριθμους. Ο αλγόριθμος DES καταναλώνει το λιγότερο χρόνο για κρυπτογράφηση.Οι AES και DES αλγόριθμοι παρουσιάζουν πολύ μικρή διαφορά στο χρόνο που απαιτείται για την κρυπτογράφηση. Το ΣΧ. 2 δείχνει χρήσεις μνήμης από τους AES, DES και RSA αλγορίθμους. Το ΣΧ. 3 δείχνει το μέγεθος του byte εξόδου για κάθε αλγόριθμο που χρησιμοποιείται στο πείραμα. Το αποτέλεσμα του ΣΧ. 1 δείχνει το ίδιο μέγεθος byte εξόδου για κάθε διαφορετικό μέγεθος του αρχείου κειμένου στη περίπτωση και των τριών αλγορίθμων.

DATA	ALGO.	TIME (SEC)	MEMORY (KB)	OUTPUT BYTE
FILE 1 (68KB)	AES	2.2	81,912	131,072
	DES	1.8	85,261	131,072
	RSA	9.4	91,814	65,536
FILE 2 (105)	AES	2.1	62,544	131,072
	DES	1.8	67,531	131,072
	RSA	10.5	77,117	65,536
FILE 3 (124 KB)	AES	2.2	53,902	131,072
	DES	2	55,395	131,072
	RSA	11.4	57,178	65,536
FILE 4 (235KB)	AES	2.4	16,679	131,072
	DES	2.1	21,189	131,072
	RSA	16.2	26,891	65,536
FILE 5 (435KB)	AES	2.6	34,207	131,072
	DES	2.4	42,113	131,072
	RSA	24.4	44,321	65,536

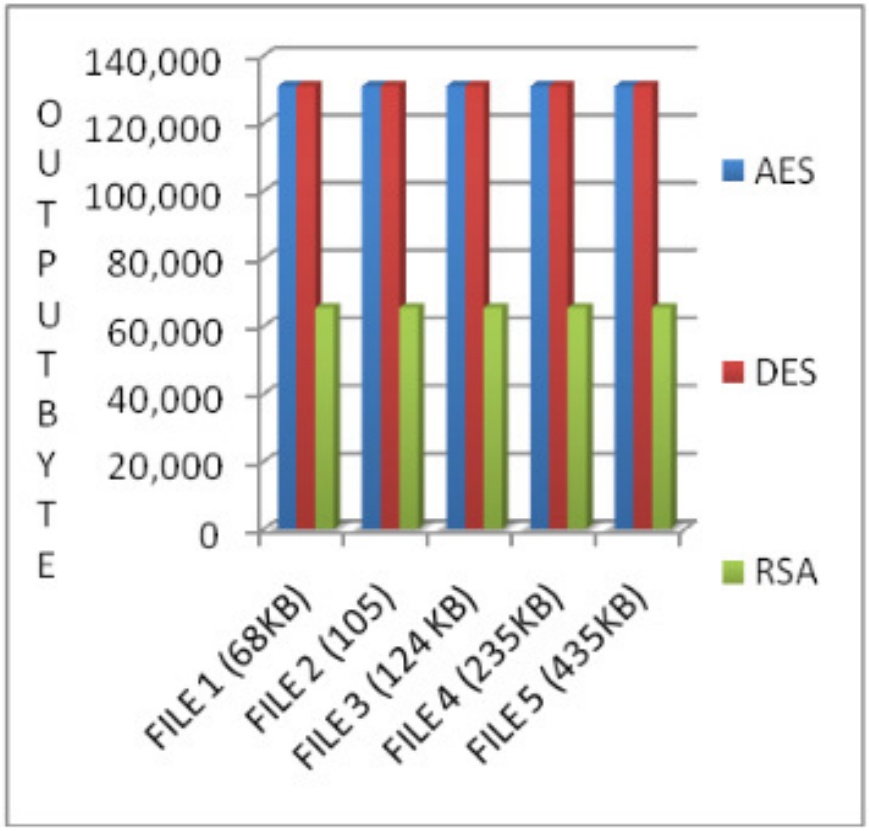
Πίνακας 1: Συγκρίσεις AES, DES και RSA του χρόνου, της μνήμης και εξόδου byte.



ΣΧ. 1: Σύγκριση των Υπολογισμού μεταξύ AES, DES και RSA



ΣΧ. 2: Σύγκριση της χρήσης της μνήμης από AES, DES και RSA



ΣΧ. 3: Σύγκριση των Byte εξόδου που χρησιμοποιούνται από την AES, DES και RSA

5.1.11 Συμπέρασμα

Οι αλγόριθμοι κρυπτογράφησης διαδραματίζουν σημαντικό ρόλο στην ασφάλεια της επικοινωνίας όπου ο χρόνος κρυπτογράφησης, η χρήση μνήμης, τα byte εξόδου και η ισχύς της μπαταρίας είναι το μείζον ζήτημα. Οι επιλεγμένοι αλγόριθμοι κρυπτογράφησης AES, DES και RSA χρησιμοποιούνται για την αξιολόγηση των επιδόσεων τους. Με βάση τα αρχεία κειμένου που χρησιμοποιούνται και το πειραματικό αποτέλεσμα, εξήχθη το συμπέρασμα ότι ο αλγόριθμος DES καταναλώνει λιγότερο χρόνο για κρυπτογράφηση, και ο αλγόριθμος AES έχει τη μικρότερη χρήση μνήμης, ενώ η διαφορά του χρόνου κρυπτογράφησης είναι πολύ μικρή στην περίπτωση του αλγόριθμου AES και του αλγόριθμο DES. Ο RSA καταναλώνει το μεγαλύτερο χρόνο για κρυπτογράφηση και η χρήση της μνήμης είναι επίσης πολύ υψηλή, αλλά τα byte εξόδου μικρότερα στην περίπτωση του αλγορίθμου RSA.

Αναφορές Κεφάλαιο V

[1] T.D.B Weerasinghe , “Secrecy and Performance Analysis of Symmetric Key Encryption Algorithms” , Vol.1, No.2, Sri Lanka June 2012, pp. 77~87
[2]Abdel-Karim Al Tamimi **,”** Performance Analysis of Data Encryption Algorithms**”** http://www.cs.wustl.edu/~jain/cse567-06/ftp/encryption_perf/
[3] Prasanth Ganesan, Ramnath Venugopalan,Pushkin Pedabachagari, Alexander Dean,Framk Mueller,Mihail Sichitiu “Analyzing and Modeling Encryption Overhead for Sensor”
[4] AL.Jeeva, Dr.V.Palanisamy, K.Kanagaram / International Journal of Engineering Research and Applications (IJERA) "Comparative analysis of performance efficiency and security measures of some encryption algorithms" , Vol. 2, Issue 3, May-Jun 2012, pp.3033-3037
[5] Mohit Marwaha, Rajeev Bedi, Amritpal Singh, Tejinder Singh "Comparative analysis of cryptographic algorithms" , Int J Adv Engg Tech/IV/III/July-Sept.,2013/16-18
[6] Diaa Salama , Abd Elminaam, Hatem Mohamed ,Abdual Kader, and Mohiy Mohamed Hadhoud "Evaluating The Performance of Symmetric Encryption Algorithms", International Journal of Network Security, Vol.10, No.3, PP.213-219, May 2010
[7] Mansoor Ebrahim, Shujaat Khan, Umer Bin Khalid "Symmetric Algorithm Survey: A Comparative Analysis" , Volume 61– No.20, January 2013
[8] Er. Rajender Singh, Er.Rahul Misra, Er.Vikas Kumar "Analysis the impact of symmetric cryptographic algorithms on power consumption for various data types",Volume: 1 Issue: 4, MAR 2013
[9] S. Pavithra, Mrs. E. Ramadevi "Study and performance analysis of cryptography algorithms", Volume 1, Issue 5, July 2012
[10] Shashi Mehrotra Seth, Rajan Mishra "Comparative Analysis Of Encryption Algorithms For Data Communication ", Vol. 2, ISSue 2, June 2011